On the identification of architectures for product programs

In a complexity cost perspective

Christian Lindschou Hansen

DTU

Thesis for the degree of Doctor of Philosophy

On the identification of architectures for product programs

In a complexity cost perspective

Christian Lindschou Hansen

December 2013

DTU Orbit ID: 192396514

2nd edition: October 2019

DTU Mechanical Engineering
Technical University of Denmark
Section of Engineering Design and Product Development

Produktionstorvet
Building 426
DK-2800 Kgs. Lyngby
Denmark

info@mek.dtu.dk
phd@mek.dtu.dk
Tel. (+45) 4525 1960
VAT 30060946

Supervisors

Main supervisor

Niels Henrik Mortensen, Professor, PhD
Section of Engineering Design and Product Development
Department of Mechanical Engineering
Technical University of Denmark

Co-supervisor

Lars Hvam, Professor, PhD
Section of Operations Management
Department of Management Engineering
Technical University of Denmark

Opponents

Torgeir Welo, Professor, PhD
Department of Engineering Design and Materials
Norwegian University of Science and Technology
Norway

Hans Johannesson, Professor, PhD
Head of Division Product Development
Chalmers University of Technology
Sweden

Dieter Krause, Professor, Dr. Ing.
Product Development and Mechanical Engineering Design
Technische Universität Hamburg-Harburg
Germany

Abstract

The topic of this thesis is the identification of architectures for product programs with the purpose of improving the competitiveness of industrial companies while reducing complexity. The focus here is reduction of time-to-market for new product development to enable companies to launch products at a faster pace while continuously reducing the associated costs of complexity.

The thesis proposes a set of frameworks and approaches to expand the body of knowledge within the research areas of product architectures and complexity management from the scientific standpoint of engineering design and product development.

The thesis presents frameworks capable of identifying a scalable program architecture for companies in varying situations, namely project-based versus product-based development, architecture for modular versus integrated product structures, and architecture for new versus existing product programs.

For this purpose, the thesis suggests the definition of a program architecture resulting from the purposeful alignment of critical aspects across market, product and production areas including constitutive/structural aspects (what the architecture *is*) as well as behavioral aspects (what the architecture *does*).

The thesis also suggests the definition of life cycle complexity cost factors representing the situations or life phase meetings where complexity costs appear. The thesis presents a 5-step approach for detecting, identifying, quantifying and allocating the complexity costs of a product program by means of the life cycle complexity factors. The approach enables the reduction of complexity reactively by cleaning out unprofitable products, and proactively by guiding product program development towards future reduction of complexity costs.

The presented frameworks and approaches have been tested in a number of case studies across a variety of industries demonstrating significant effects in terms of time-to-market reduction, improvement of R&D efficiency, and complexity cost reduction.

Keywords: Product architecture, program architecture, complexity costs, time-to-market

Resumé

Temaet for denne afhandling er identifikation af arkitekturer for produktprogrammer med det formål at forbedre konkurrencedygtigheden for industrielle virksomheder og samtidig reducere kompleksiteten. Fokus her er at opnå reduktion af gennemløbstid for udvikling af nye produkter og at fremme lanceringshastigheden i takt med at kompleksitetsomkostningerne reduceres kontinuerligt.

Afhandlingen foreslår et sæt af rammeværk og fremgangsmåder, der udvider det eksisterende forskningsfelt indenfor produktarkitekturer og styring af kompleksitet ud fra forskning indenfor mekanisk produktudvikling og innovation.

Rammeværket tillader at identificere en skalerbar programarkitektur for virksomheder i en række forskellige situationer. Disse situationer inkluderer arkitektur for projekt-baseret versus produkt-baseret udvikling, arkitektur for modulære og integrererede produktstrukturer samt arkitektur for nye og eksisterende produktprogrammer.

Afhandlingen foreslår hertil definitionen af en programarkitektur, som resultat af den formålsbestemte afstemning og tilpasning af kritiske aspekter på tværs af marked-, produkt- og produktionsforhold, der inkluderer både konstitutive/strukturelle aspekter (hvad arkitekturen *er*) og adfærdsmæssige/funktionelle aspekter (hvad arkitekturen *gør*).

Afhandlingen foreslår desuden en femtrins fremgangsmøde for at opdage, identificere, kvantificere og allokere kompleksitetsomkostninger for et produktprogram ved hjælp af kompleksitetsomkostningsfaktorerne. Fremgangsmåden gør det muligt at reducere kompleksitet reaktivt ved at rydde ud i urentable produkter, og fremadrettet ved proaktivt at guide udvikling af produktprogrammet imod fremtidig kompleksitetsreduktion.

Afhandlingen foreslår hertil definitionen af kompleksitetsomkostningsfaktorer fra livscyklussen, der repræsenterer de situationer eller livscyklusmøder, hvor kompleksitetsomkostningerne kommer til syne.

Rammeværkerne og fremgangsmåderne er testet i en række casestudier på tværs af forskellige industrier og demonstrerer signifikante effekter i form af reduceret gennemløbstid for introduktion af nye produkter, forbedring af effektiviteten af produktudviklingsaktiviteter samt reduktion af kompleksitet og omkostningerne herved.

Stikord: Produktarkitektur, programarkitektur, kompleksitetsomkostninger, udviklingstid, gennemløbstid

Preface

This PhD thesis documents the outcome of three years of research carried out at the Technical University of Denmark (DTU). The project has been carried out at the Section of Engineering Design and Product Development at the Department of Mechanical Engineering.

The author is a part of the Product Architecture Group led by Professor Niels Henrik Mortensen. The project was initiated in August 2009 and ended in December 2013 with a duration of 36 months.

The research work is intended for both researchers within engineering design science as well as practitioners designing and managing product programs in industrial companies.

In retrospect, the area of architecture-based product development has been an interest of mine since my very first semester at DTU, where my project group and I designed a modular waste handling system for the municipality of Copenhagen. The flexibility of being able to configure the exact group of waste handling units a location needs, while at the same time harvesting economies of scale from a common mechanical building principle (an architecture) was simply too intriguing to discard. Ever since then, my personal interest in the subject has been sustained and developed through a number of industrial projects, and by discussions with other peers and practitioners. In this regard, I would like to thank a number of people to whom I am very grateful.

First of all I would like to thank Professor Niels Henrik Mortensen for bringing me onboard and showing great confidence in my research activities. I would like to thank you for many fruitful discussions, inspirational talks, and your clear-sightedness in helping me bridging the practical problems in industry with the theory-based ones in the academic world.

Second of all I wish to thank Ulf Harlou for valuable guidance on the industrial applicability of methods and approaches. Your team has been a constant source of inspiration.

I also wish to thank the current members of the Product Architecture Group at DTU including Professor Emeritus Mogens Myrup Andreasen and all my fellow PhD students. I have enjoyed every discussion we have had, and I hope you will still share your 'critical enthusiasm' with me going forward.

I would also like to thank my international network of fellow researchers who have challenged me during seminars, courses and conferences across Europe, including Henry Jonas & Co. (Technische Universität Hamburg-Harburg), Marcel Michaelis, Christoffer Levandowski & Co. (Chalmers Tekniska Högskola), Maik Maurer & Co. (Technische Universität München), and Mario Storga (University of Zagreb). You have all made a difference.

At last but not least I would like to thank my family and closest friends for supporting me through stressed times and for showing your interest in my work.

The research is funded by the Danish Agency for Science, Technology and Innovation (36%) and by DTU (64%).

Christian Lindschou Hansen

Copenhagen
December 2013

Table of contents

1 Introduction

The introduction includes a review of the challenges encountered in industry and the needs arising from academia. The review is based on fifteen years of experience within the research group in the area of product architectures and complexity reduction, of which the last six years have included the participation of the author in the form of multiple industrial projects and recurring literature reviews. Lastly, the industrial and theoretical scope of the thesis is addressed.

1.1 Challenges in industry

Many companies developing and producing mechanical products are facing severe challenges in maintaining and improving their competitiveness. There are many reasons for this, but it is clear that the accelerating globalization is playing an enormous role in setting this stage.

A lack of competitiveness is often mentioned as the challenge of

- *Reducing time-to-market* to provide new and improved solutions to the market, faster
- *Offering product customization* and at the same time achieve benefits from economies of scale
- *Achieving attractive cost levels* across a product program while delivering the performance necessary to compete
- *Controlling and reducing the increasing complexity* of products and processes in order to reduce costs and improve agility of business processes

These challenges are not new. However, globalization has resulted in these challenges being more prevalent during the time of writing than ever before.

During the last twenty years, research in product architectures and product platforms has suggested a number of tentative answers to the challenges mentioned above. However, in the meantime, reality has outpaced a number of previous contributions in the sense that globalization has not been standing still. Companies are no longer *just* doing global procurement, but their R&D activities and their production footprint is global too. They also compete in several different market tiers in the global market place, all in all magnifying the challenges mentioned above, while making the need for a solution more and more evident.

Yet, when interviewing decision makers of industrial companies, the awareness of the body of knowledge within research in architectures is more present than it was ten years ago. In most companies, top management acknowledges the relevance of architectures in meeting the challenges mentioned above, and an international survey made among 1,400 senior executives show that 94% of top managers acknowledge the importance of controlling complexity as important to their company's success (KPMG 2011). This is a drastic change from previous years, where the severity of the problem was not widely accepted or even thoroughly understood.

So, despite the increasing awareness of the subject, one can wonder why architecture initiatives are not yet on the agenda in every management board of industrial companies, when the same survey shows that 70% agrees that increasing complexity is one of the biggest challenges their companies face today?

During interviews with industry, the author has recorded a number of apparent barriers limiting the prevalence of architecture initiatives:

- *The cost of complexity* is not known as it is difficult to identify and quantify (Schuh 2001), and most organizations naturally experience difficulties in managing what they cannot measure
- *Lack of down-to-earth methods* for architecture identification and complexity reduction that can be readily applied (Krause et al. 2013)
- *Lack of high quality decision basis* to support the complex decisions that working with architectures entails
- *Extensive project orientation* makes it unattractive to prepare programs for future derived product launches – there is risk involved and no reward
- *Widespread silo thinking* across areas of marketing, R&D and production often making the CEO the only person with complete overview and responsibility for an architecture – even though this is a classic barrier, it still prevails in many organizations

Alongside this development, the German automotive industry reports best-in-class profitability, a projected decrease in time-to-market by at least 30%, increased configuration flexibility and decreased costs exemplified by Volkswagen's introduction of their MQB architecture (*Modularer Querbaukasten* or Modular Transversal Toolkit) (Buiga 2012). The recipe seems simple. Standardize the cost-intensive elements and improve the possibility for product customization to fulfill the requirements for differentiating variety between brands, while improving volume to achieve extensive effects of scale.

Still, while a few companies excel in the exercise, the greater part of companies are lacking behind leaving a huge improvement potential untouched. As caution, the research group of the author has seen several failed initiatives, exemplified by too rigid product platforms being designed a *little bit off* the most attractive market segments, and with cost-points *out of synchronization* with sales volumes and means for achieving scale effects in production – all in all resulting in unattractive cost levels, offerings that are not competitive and little to no preparation towards future launches. However, none of these projects have been carried out by resources not wishing to do their absolute best for their companies. A key takeaway is that identification of a scalable architecture capable of providing significant benefits to an industrial company is not a simple task.

The industrial aim of this thesis is to provide a contribution enabling more companies to identify a scalable architecture capable of providing significant competitive advantages, and to ensure that the reduction of complexity is measurable on the bottom line.

1.2 Needs from academia

Research in architecture-based product development is not a new area of research. However, seen from the perspective of architectures as a means for improving competitiveness and reducing complexity, it lacks maturity in a number of dimensions.

As has been the case for more than a decade, no one denies the validity of the prevailing theories within the area (Baldwin and Clark 2000, Ericsson and Erixon 1999, Meyer and Lehnerd 1997, Ulrich 1995), and a strong majority of researchers acknowledge the existence of a huge potential hidden in the 'optimal' implementation of architectures (Gershenson et al. 2003). The situation is indeed that a lot has been achieved (Jiao et al. 2007). Still, numerous sections describing 'further works' in top tier scientific papers point out that there is a need for frameworks and approaches that:

- Support the identification of flexible architectures supporting future product launches (Johannesson 2013)
- Bridge architecture with economic benefits (Campagnolo and Camuffo 2010)
- Align product architecture with supply chains (Langenberg et al. 2012)
- Take into account the contextually different situations companies operate in (Jiao et al. 2007)
- Provide a hands-on approach to the identification of architectures (Krause et al. 2013)

In the meantime, other contributions are centered on the presentation of huge matrices taking weeks to fill out leaving the practitioners with more questions than answers. Also, two thirds of all contributions assume the existence of *a priori* solutions upon which to apply sophisticated algorithms for optimization (Simpson 2004), where idealistic scenarios are assumed. While this is an intriguing situation for the application of an algorithm, the approach is not supporting the actual needs of industrial companies as recorded during interviews.

To sum up, it appears from literature that there is a need for integrating the support for architecture identification with several business aspects of an industrial company in terms of market fit, complexity reduction in operations, and ability to launch competitive products in a faster pace. This means that there is a need for expanding the current notion of architectures from belonging to the product domain alone; based on the assumption that it is not possible to evaluate a product program solely based on the constitutive or structural aspects of the products themselves.

This expansion is considered a necessary step towards being able to support the development of architectures for product programs providing significant benefits to the companies at question. The expansion also magnifies the need for a comprehensive decision basis to support the complex decision making involved. There is a need for developing support capable of supporting the substantial evaluation of for example:

- Adding or removing a feature
- Decoupling or integrating a module
- Adding or removing a performance step etc.
- Adding or removing a product variant
- Shift the point of product customization or variant creation

A theoretical gap is identified in current contributions' lack of ability to provide support for tracing consequences across market, product and production domains as well as taking into account the costs of complexity for entire product programs.

1.3 Scope of thesis

1.3.1 Theoretical scope

As the research area of architectures and complexity reduction has widespread impact across several theoretical disciplines, it is important to delimit the scope of the research in order to make it *researchable*.

The theoretical scope of the thesis is founded within engineering design science. However, being a composite area of research there are very strong links to the management of industrial operations as well as business marketing and product planning, as the benefits of successful implementation of architectures and complexity reduction initiatives are to be harvested across adjoining parts of the value chain and product program life cycle. Therefore, it is relevant to include literature and concepts from these disciplines as well.

In terms of excluded theory, the areas of organization theory and decision theory are outside the theoretical scope of the thesis. Minor parts of socio-technical theory have been included to explain and elaborate the staging of architecture modeling tools as types of management technology.

Chapter 2.3 will elaborate on the research scope by including the scientific delimitations, and section 2.3.1 will elaborate on the adjacent disciplines that are closely linked to the theoretical focus of this thesis and outline the areas of relevance and contribution.

1.3.2 Industrial scope

The case companies included in the research work are all chosen to cover a wide spectrum of Scandinavian industrial companies. The chosen 12 companies are all developing and producing mainly mechanical products with varying electronics and software contents. The companies operate across 11 different industries:

- Energy
- Construction
- Telecommunications
- Automotive
- Industrial infrastructure
- Electronic solutions for industrial infrastructure
- Mechanical solutions for industrial applications
- Machinery for consumer applications
- Machinery for industrial and commercial applications
- Medical devices
- Business and consumer electronics

The companies vary in business type in the sense that they address both consumer and professional markets. The companies also cover a wide variety of product customization levels as they cover the full range from being 100% product-based with a definite solution space; to being 100% project-based with an open solution space while in several cases also serving OEM customers. Product-based companies develop a range of commercial variants of which they control the specification and design completely. This is in opposition to the project-based companies that develop solutions to match specific requirements from customers resulting in a setup where their customers in many cases own the final specification and design.

These rather diverse ways of being in the market place poses quite different challenges to architecture and complexity reduction initiatives. It has been an important cornerstone for the research presented here to communicate research results that are consciously aware of these contextual differences posed to the 12 different companies of study. There are several reasons for this:

Focus of architecture initiatives

As the focus of an architecture or complexity reduction initiative is both difficult and critical in order to ensure that the desired goals are met, contextual awareness is very important. Goal setting and scoping of architecture and complexity reduction initiatives can vary a lot looking at the diversity represented by the differences among the case companies.

Improvement of comparability and transferability of experiences

It is the experience of the author that is has traditionally been difficult for researchers and practitioners to compare their case studies and to exchange experiences about their research results within this composite area of research. Contextual awareness is considered an important means to ensure initiatives that are tailored to fit the needs of the industrial setting while improving the knowledge transferability among researchers and practitioners. The contextual awareness should include the situational and external factors posed to the industrial setting, where the architecture and complexity reduction initiative should have effect.

Table 1 in section 2.6.1 contains an overview of the case studies conducted during the research.

1.4 Thesis outline

The thesis is divided into six parts:

Figure 1 – Thesis outline

The thesis is structured by parts (i), chapters (i.i), sections (i.i.i), and paragraphs.

2 Research setup

The second part of the thesis will present and elaborate on the setup of the research. This is done by outlining the objectives, research questions, scope, methods and research plan including a strategy for verifying the results. Part two ends by presenting the case studies and how the research stages have been orchestrated to produce the reported results.

2.1 Introduction

The following chapters will elaborate on the research setup as briefly introduced in the previous part of the thesis. The chapters will explain how the challenges mentioned in Part 1 are made researchable by outlining the research objectives, detailing the research scope, and elaborate the research methods applied. Lastly, the research verification method is outlined along with the research plan including an introduction to the case studies.

2.2 Research objectives

2.2.1 Research questions and working hypotheses

The research questions are divided in two separate areas. One is research questions regarding the identification of an architecture for a product program, and the other is regarding complexity reduction based on quantification of complexity costs.

Identification of a program architecture

In order to clarify the title of the thesis, the chapter will start out with a basic definition. A 'product program' is analogous to the definition of a product portfolio in other key contributions:

> "A product portfolio is defined as the complete set of possible product configurations offered by a business unit at a given point in time" (Meyer and Lehnerd 1997)

With this definition in place, let us take a look at the overall research question of the research that is closely associated with the ultimate criterion of the research conducted: How to improve time-to-market and R&D efficiency for the development of product programs?

However, as there are obviously a huge number of different types of answers to this that are simply outside the scope of this research, this question is considered too broad to guide the way for focused research work. Therefore, it is necessary to formulate a research question that captures more detail of this overall challenge and set the stage for being able to answer this on a more tangible level:

Research question A1:

> How to make the most critical aspects of product program design explicit during the early development phases of a new product program?

The term critical refers to the aspects of product program design that are dominating or decisive for achieving the ultimate criterion as described in the overall research question.

The term explicit is used in opposition to implicit, and indicates that the aspect is clearly expressed (e.g. visually) and readily observable by stakeholders across domains, thus implying that there is a need for a modeling language that is cross functionally understood.

The early development phases refers to the time period of a program development project, where it is still unclear which requirements to fulfill, what elements to develop, and how to produce them. Typically, the early phases are characterized by the project being about to enter a Stage-Gate® process or has just done so.

To complement research question A1 which is centered on the identification of a program architecture during the early phases of an already defined development project, a second research question is necessary for the situation where there is no development project defined. In this situation, the challenge is to unveil whether there is a potential to improve time-to-market and R&D efficiency going forward, by identifying and implementing a program architecture:

Research question A2:

> How to identify the most critical program decisions of an existing product program and make the central aspects of these explicit?

In order to guide the research and answering of the research questions, a number of working hypotheses are formulated. The working hypotheses are closely related to research questions A1 and A2, and serve to guide as preliminary and overall answers to these:

Hypothesis A1:

> The modeling of the program critical aspects across market, product and production domains will improve the decision basis for product program design, in order to improve time-to-market and/or R&D efficiency for product launches derived from the program architecture.

Hypothesis A2:

> It is possible to identify a program architecture for a product program including both structural/constitutive and functional/behavioral aspects through the application of explicit modeling techniques and through appropriate staging.

In addition to research questions A1 and A2, a third research question is formulated.

During the research project, the author has experienced that the importance of including situational and contextual awareness is highly critical when considering the success of architecture initiatives.

Research question A3 can be regarded as a supplement research question which is formulated in order to address this situational and contextual awareness:

Research question A3:

> How to take the contextual differences of a company into account when scoping and comparing architecture initiatives?

Architecture initiatives are here defined as 'vehicles' or projects implementing parts of a program architecture. Hypothesis A3 is a tentative overall answer to research question A3:

Hypothesis A3:

> It is possible to classify program architecture initiatives on the basis of differentiating external factors to allow for the inclusion of contextual criteria when defining, scoping and comparing architecture initiatives.

The three working hypotheses are central for guiding the research and come into play while answering research questions A1, A2 and A3.

Complexity reduction based on quantification of complexity costs

The overall research question here is also closely associated with the ultimate criterion of the research conducted: How to reduce the costs of complexity?

As with the previous area, the overall research question is too broad to guide the way for focused research, and it is necessary to formulate questions that allow for more specific answers.

Research question B1:

> How to identify and quantify complexity costs of a product program?

Research question B2:

> How to use quantification of complexity costs to support decision making in product program design?

In order to guide the research and to address each of the research questions, a number of working hypotheses are formulated. Hypotheses B1 is a tentative answer to research question B1:

Hypothesis B1:

> It is possible to identify and quantify the costs of complexity for a product program and allocate the costs directly to the individual product variants.

Hypotheses B2 is a tentative answer to research question B2:

Hypothesis B2:

> It is possible to rationalize a product program based on the calculation of complexity costs.

These will be presented and answered in detail when presenting the papers.

2.2.2 Impact models

In order to stage the research questions in their industrial context, Figure 2 and Figure 3 represent Impact Models showing the desired situation after the application of the suggested frameworks and procedures. The Impact Models are based on the method by Blessing and Chakrabarti (2009). This is to show what the intended impact of the research results is and provide an idea about how the central factors of the research setup are causally linked.

In accordance with the research questions separate impact models have been made for the two areas of research.

Identification of a program architecture

Figure 2 shows the Impact Model of the research regarding the identification of a program architecture.

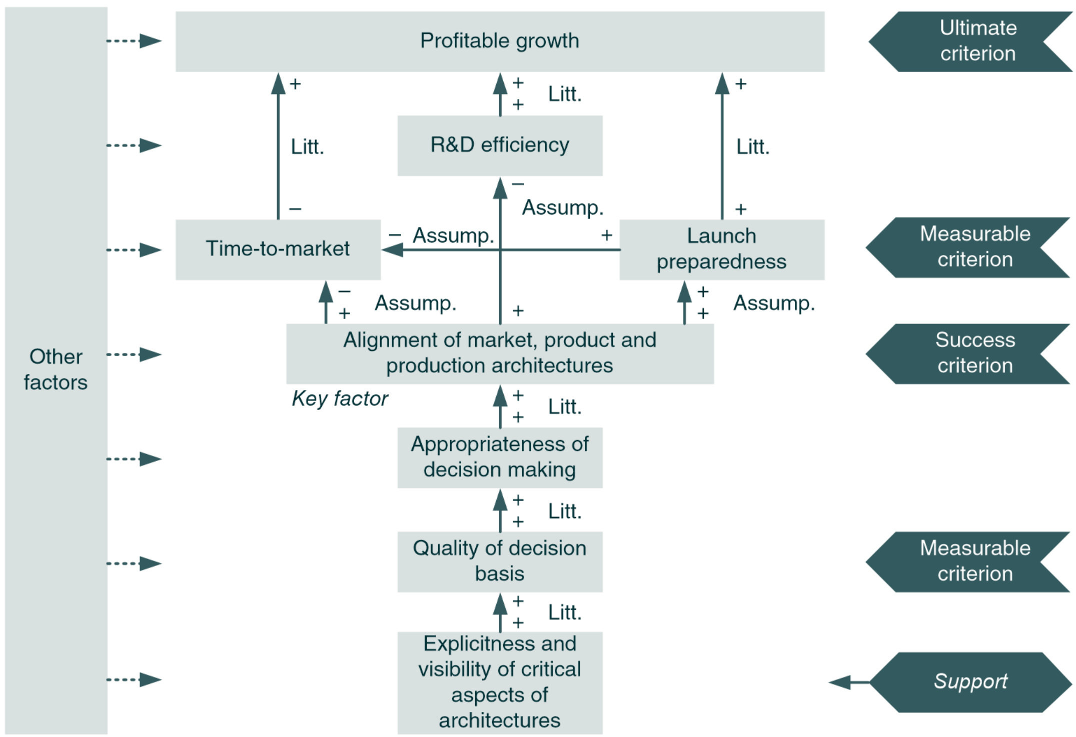


Figure 2 – Impact Model A

Impact Model A takes its starting point in the link between explicitness and visibility of critical aspects of architectures and the quality of the basis for decision making. Hypothesis A1 takes its starting point in this causal link too.

The second link is the relation between the quality of the decision basis and the appropriateness of decision making. Much literature supports this relation being as simple as stating that appropriateness of decision making is highly reliant on the quality of the decision basis. Of course, other factors influence the decision basis as well as the appropriateness of decision making. One example could be competences of decision makers and the underlying incentive structures.

The third link connects the appropriateness of decision making with the resulting alignment of architectures. The concept of alignment will be further elaborated in several sections and appended papers of the thesis. The concept of alignment of architectures is considered the key factor of this research area, as it can be considered the most useful factor to address in order to improve the two important partly measureable criteria; time-to-market and launch preparedness.

Launch preparedness simply refers to the situation where a company is prepared for a product launch in the sense that it is not necessary to start over again when a new product shall be introduced to the market. Launch preparedness could be misconceived for being an abstract phenomenon, but it is an aim of this thesis to clarify what this means in practice.

In line with many other research themes within engineering design research and design science the ultimate criterion is the achievement of profitable growth.

Many links within the Impact Models are based on experiences from literature, which will be covered by the theoretical basis in Part 3, research results in Part 4 and research conclusions in Part 0.

Complexity reduction based on quantification of complexity costs

Figure 3 shows the Impact Model of the research regarding complexity reduction by means of complexity costs quantification.

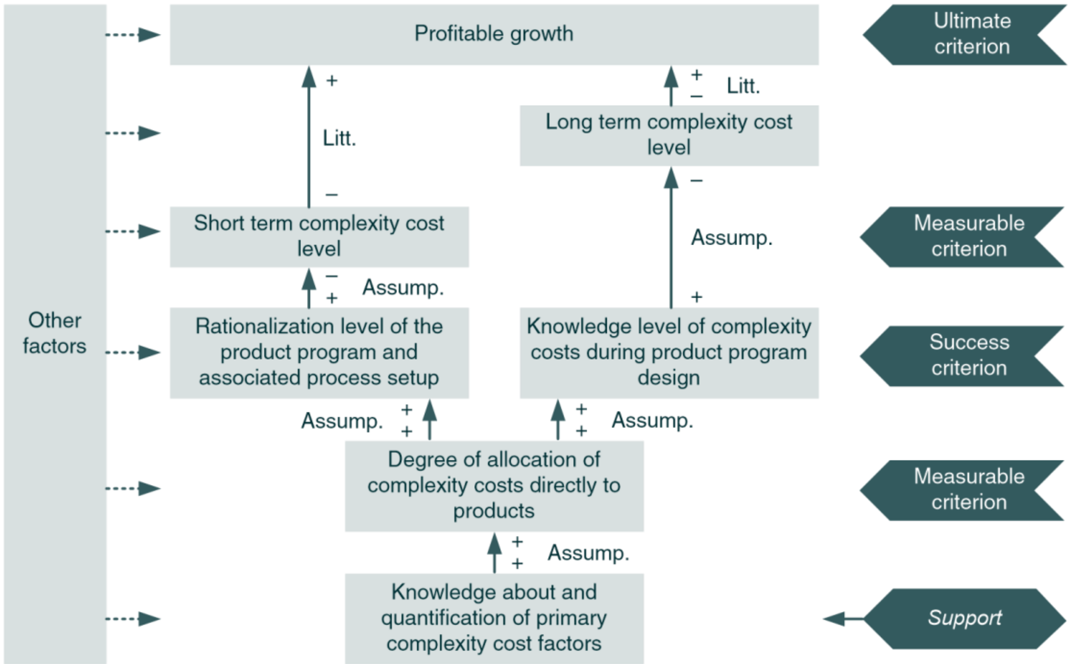


Figure 3 – Impact Model B

Impact Model B takes its starting point in the identification and quantification of complexity cost factors. The primary complexity cost factors are areas of significant costs that are typically asymmetrically or unevenly distributed across comparable product variants. Following this comprehension of complexity costs, the knowledge about and quantification of complexity cost factors is closely linked to the possible degree of allocation of complexity costs to individual product variants. The second factor presupposes the first factor to be fulfilled. In other words, quantification of complexity cost factors makes it possible to allocate a share of complexity costs directly to individual product variants. The allocation factor is linked to the ultimate criterion of profitable growth in two ways:

- The first link is through the rationalization level of the product program and the associated process setup. This is based on the assumption that the degree of allocation of complexity costs makes it possible to rationalize the product program and associated process setup. This can be done e.g. by reducing the number of product variants to increase the program contribution margin, or by process improvements capable of reducing the product program complexity costs in general. This can lead to a situation where the level of complexity costs is reduced in the short term. Therefore, the link between the measure criterion and the success criterion is contingent upon the rationalization initiative following the allocation of complexity costs directly to products.

- The second link is through a proactive avoidance of complexity costs to occur again. This can be done by the incorporation of knowledge of the complexity cost factors and their quantification and allocation into the development phase of product program design. A company's R&D and product management function is the natural client of this knowledge. This is the key to significantly affecting the long term development of the complexity cost level.

2.3 Research scope

The research is founded in engineering design, as described in section 1.3.1., and to a certain extent where the field of engineering design meets primarily operations management – but also including business marketing and systems engineering. The next section will elaborate on the areas of relevance and contribution by applying the modeling technique from Blessing and Chakrabarti (2009).

2.3.1 Areas of relevance and contribution (ARC)

In Figure 4 the ARC-diagram is presented. The center ellipse represents the research subject of the thesis, which is here denominated as 'program architecture and complexity reduction'. The grey circles depict the useful areas of relevance that are all included in the literature studies conducted throughout the research work, and from which many concepts and notions in this research relate to.

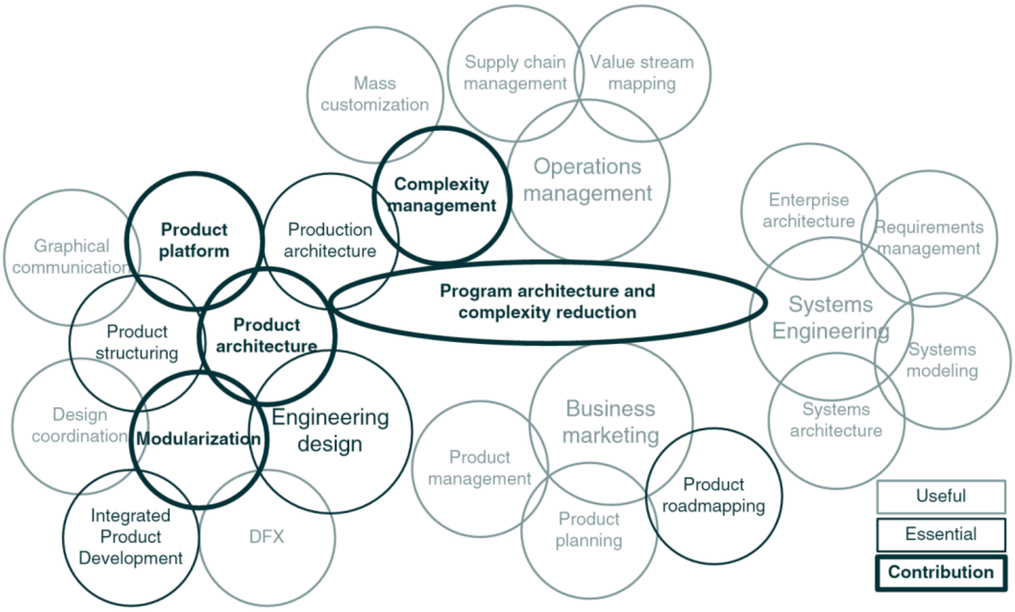


Figure 4 – ARC diagram

The two main areas are 'Product architecture' and 'Complexity management'. The area of Product architecture is closely related to product platform and modularization, and it is linked to 'Operations management' through 'Production architecture' and 'Mass Customization', among others. Closely related areas here are supply chain management and many tools now belonging to the tool-kit of Lean, e.g. value stream mapping.

On the right side of the diagram is 'Systems Engineering', from where many cross disciplinary fields have emerged, which are relevant to this research including 'Enterprise architecture', 'Systems architecture' and the disciplines of 'Requirements management' and 'Systems modeling'. These can be regarded as so-called *supra-domain* fields, as they are not closely founded within a particular engineering discipline, as for instance 'Product architecture' is from the field of engineering design including mechanical engineering and product development. Instead these supra-domain fields focus on the rational structuring of the whole company and its products and operations, which is addressed using a systems theory approach.

On the lower middle section of the diagram, is the 'Business marketing' circle, which more specifically includes the relevant areas of 'Product management', 'Product planning' and 'Product roadmapping', all centered on the commercial planning of products and their fit towards market tiers and the coordination of optimal launch sequences. This area represents the market side of the 'Product architecture' research field and holds great importance concerning the commercial aspects of product program design.

Part 3 will elaborate on the theoretical basis behind the research presented here, while the appended papers presented in Part 4 will include the state-of-the-art relevant to the specific contribution each one of them represents.

2.3.2 Limitations

A number of limitations for the research exist:

- Only very little inclusion of organization theory
- No inclusion of 'activity' or 'knowledge' architectures
- No inclusion of phenomena about architecture ownership
- No inclusion of IT-support needed for identification and maintenance of architectures

The following chapter will elaborate on the research methods applied.

2.4 Research methods

As described in section 1.3.1 the theoretical scope of the research is founded within engineering design science where it belongs within the field of applied research. A mix of research methodologies described by Joergensen (1992) as well as Blessing and Chakrabarti (2009) has been applied.

2.4.1 Approach 1: Problem-based and theory-based research

Being characterized as applied research entails the opportunity of combining problem-based research with theory-based as described in Figure 5 (Joergensen 1992). The problem base reflects the industrial challenges, while the theory base reflects the state-of-the-art that does not provide satisfying explanations to the phenomena at question.

Both the problem-based approach and the theory-based approach have been used during the research. In fact, these have been entangled in the way that the theory based approach has been a key ingredient during the development of new models etc., but very seldom the only ingredient or the governing approach. In order to ensure applicability of the research, the major path of the research has taken its starting point in the problem base, where the theoretical path has been involved in smaller sprints during analysis and synthesis phases.

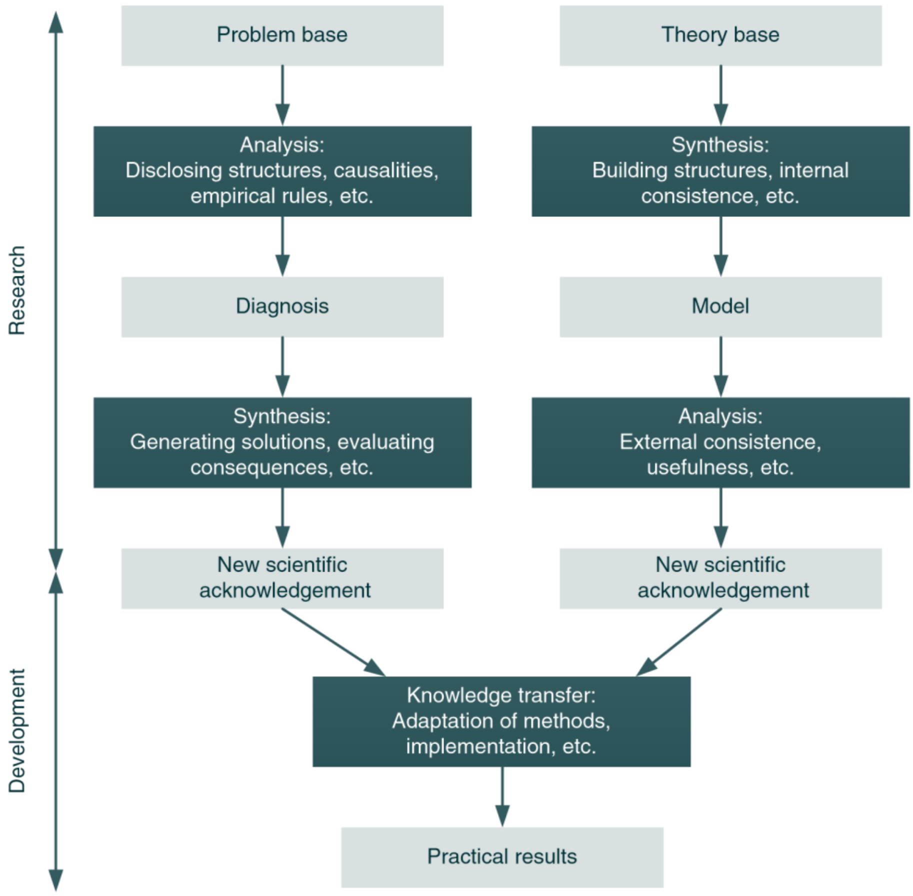


Figure 5 – Problem-based and theory-based engineering design research methods (Joergensen 1992)

2.4.2 Approach 2: Design Research Methodology (DRM)

The DRM framework provides a holistic framework of the engineering design research process. The framework has been applied in the research presented here as a foundation but with modifications.

The DRM framework starts out with a research clarification phase that frames the research goals and describes the desired situation of what impact the research is intended to have. After this, a series of descriptive and prescriptive phases follow that vary between the descriptive analysis of literature and empirical data to build up understanding of the relation between the factors involved; and the prescriptive testing of developed support. The DRM framework has its strength in providing the overview and elaboration of the research phases varying between being descriptive and prescriptive, however, it is the experience of the author that the linearity presupposed in the framework does not always reflect the practical production of research insights that in e.g. case studies. Therefore, two extra DRM stages were introduced in this research, to allow for sufficient iterations in order to qualify research insights that could be regarded as significant (see Figure 8). In other words, having a single prescriptive loop during a PhD study might not always be enough to allow for sufficient immersion into the research field, trial-and-error development of support, and last but not least critical and in-depth reflection on the industrial impact of the support.

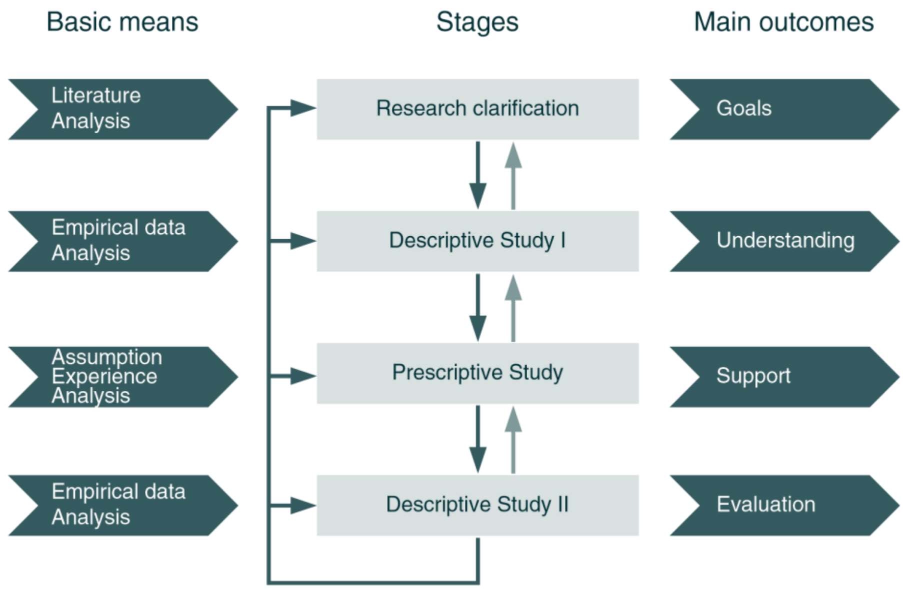


Figure 6 – DRM framework

2.4.3 Method

The research method used here can be described to fit within qualitative research of empirical type, and more specifically switching between the practical problem-based world and the theory-based world (Joergensen 1992). While some ideas for the enhancements of the presented frameworks of the thesis were generated while reviewing literature and discussing with research peers, other ideas were conceived while being embedded in the practical setting of a case study. It is of great importance to the author that the strength of this 'dual path' research setup is conveyed.

The practical problem-based path shares resemblance to the working approach of action research. Many recent definitions of action research exist, while Coughlan and Coghlan (2002) extracted the common denominators as being:

Research in action rather than research about action; research being participative and concurrent with action, and research as a sequence of events and an approach to problem solving.

Action research

The type of research results presented here is difficultly obtained by traditional means including surveys or observational case studies. As the extensive use of visual modeling techniques does not have a long and established tradition within corporate or academic settings, it is difficult to study the phenomena by these traditional means. Therefore, to accommodate this type of research, it is necessary to embrace new ways of theory building. Action research serves as an appropriate means here by closely linking the development and refinement of the modeling techniques while concurrently enabling the researcher to test and improve the modeling techniques to bring about change in the industrial setting. Hereby fulfilling the dual purpose of practical problem solving with incremental theory building, which is the main challenge of action research in practice.

Building knowledge

It is fair to state that action research has been established as a rigorous research method, among the array of alternative ways that research within operations management can take form (Karlsson 2008). In opposition to the positivistic sciences, action research creates knowledge that is situational based and out of practice. The aim is to solve a practical problem and extract emerging theories and results out of contextually embedded data. Hence, the role of the researcher is not to be a detached and neutral observer, but instead being an actor and change agent immersed in the industrial practical setting.

A common critique concerning action research based approaches is the resemblance with 'pure practical problem solving'. The concern can be legitimate if the researchers are not sufficiently aware of the theoretical foundation on which to build upon. If this is not the case, the action researcher cannot take the role of reflective researcher in the practical setting, making the research contribution difficult or impossible to formulate.

Generalizability and transferability

A constant challenge within this research paradigm is the construction of the truth claim. Checkland and Holwell (1998) formulates this in terms of validity requiring a recoverable research process that bases itself upon a prior declaration of the epistemology by means of which the reasoning behind the knowledge acquisition will be made. Hence, the action researcher frames the understanding of what research outcomes that qualify as research contributions, by explicitly stating the knowledge foundation upon which the research builds upon while constraining any formulation of research outcomes, and by adhering to the three basic characteristics of knowledge built from action research: It is situation specific, emergent and incremental (Karlsson 2008). See Part 3 for a review of the theoretical basis.

Being situation specific is of course a constraint towards generalizability and transferability, but carrying out the case studies within several companies across 11 industries, improves the generalizability. Also given the fact that the companies of study are not niche players in their industry, but on the contrary having similar characteristics to many other competitors improves the aspect of knowledge transferability. These issues will be further addressed in Part 0.

2.5 Research verification

This research being qualitative of empirical type stipulates a thorough discussion and consideration about how the results can be verified. As research within engineering design science cannot be isolated from human involvement and is difficult to verify statistically, as in the case of the natural sciences, it is important to verify results by other means.

To ensure *research rigor* a number of validation frameworks are considered to be useful.

2.5.1 Validation frameworks

Quasi-experimentation framework

Four types of validity are considered according to Cook et al. (Cook et al. 1979):

- Statistical Conclusion Validity
- Internal Validity
- Construct Validity of Causes and Effects
- External Validity

As there is only little possibility for making statistical conclusions based on the limited samples from this research project, the statistical conclusion validity is not possible to achieve. However, the residual validation types seem *valid* to apply.

Two-approach framework – derived from mechatronic design research

Two types of validity of design theories are considered according to Buur (1990).

Logical verification

- Consistent argumentation; i.e. no internal conflicts between individual elements of the theory
- Completeness in proposition; i.e. all relevant phenomena observed can be explained or rejected by the theory
- Methodological alignment; i.e. the theory cannot be in conflict with well-established and successful methods
- Practical adherence; i.e. the theory can explain problems encountered in case studies

Verification by acceptance

- Statements of the theory are acceptable to experienced practitioners
- Models and methods derived from the theory are acceptable to experienced practitioners

The framework by Buur (1990) is intriguing in the sense of its pragmatic approach to verification which is closely tied to the theory's practical acceptance and usefulness.

The Validation Square

In order to account for the interpretation of the engineering design research process presented by Joergensen (1992) in Figure 5, a third validation framework is included to match the theory-based and problem-based research paths.

The Validation Square is presented by Pedersen et al. (2000).

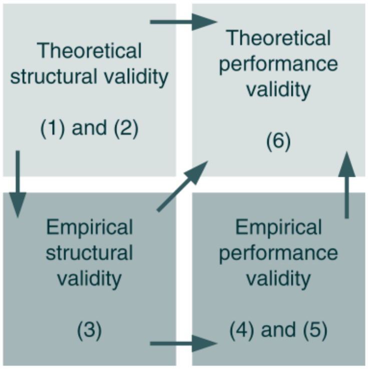


Figure 7 – The Validation Square (Pedersen et al. 2000)

Pedersen et al. (2000) uses the classic partition between structural and behavioral properties of theoretical and empirical engineering design research. Based on this, three theorems are formulated in order to ensure that the research results are effective:

(1) accepting the individual elements constituting the method

(2) accepting the internal consistency if the way the constructs are put together in the method

(3) accepting the appropriateness of the example problems that will be used to verify the performance of the method

Another three theorems are formulated to ensure that the research results are efficient:

(4) accepting that the outcome of the method is useful with respects to the initial purpose for some chosen example problem(s)

(5) accepting that the achieved usefulness is linked to applying the method

(6) accepting that the usefulness of the method is not limited to and extends beyond the case studies

Across scientific disciplines the words verification and validation are sometimes used interchangeably. To avoid confusion, this thesis is based on the interpretation of 'verification' that verified results ensure that the process of obtaining them has been correct, and 'validation' refers to the assessment of the results or outcome being correct.

Section 2.5.2 will elaborate on the verification of research methods used in the case studies, and chapter 5.3 will elaborate on the actual validation of the results.

2.5.2 Case study validation

The results from the case studies are generally verified using six theorems from The Validation Square framework (Pedersen et al. 2000). The validity of the research results themselves are further elaborated in Part 0. Below is an assessment of the validation method used for the case studies as a whole:

(1) *Individual elements:* The research being based on widely accepted system theory, theory of architecture and product families and theory of management technology ensures that the constitutive elements of the tested architecture framework and approaches for complexity reduction are accepted in the field (see Part 3).

(2) *Internal consistency:* The research being based on a widely accepted theoretical basis, the results being published and peer-reviewed, and the completion of 23 case studies for development, testing and refinement of the proposed research results, ensures that the internal elements has been tried out for consistency – simply by proving robustness in the research community and in the field.

(3) *Appropriateness of example problems:* The 23 case studies represent a wide spectrum of industrial settings across 11 industries in Scandinavian companies. However, a common denominator for all the case companies is their growing complexity which is limiting their profitability and slowing down innovation. All case companies developing and producing products of mainly mechanical nature ensures that the example problems dealt with in the case studies serve as an appropriate basis for the research.

(4) *Useful outcome:* As the appended papers bear witness of, the measurable criteria of the impact models (see Figure 2 and Figure 3) have been assessed as successfully met. The technique for achieving this varies from interviews with key stakeholders all the way to quantified measures.

(5) *Link between method application and achieved usefulness:* As the case studies are staged in practical settings with limitless amounts of other impacting factors, this theorem is difficult to fulfill. Moreover, in case studies partially following an action research-based method, it is not possible to subtract the role of the author and evaluate case studies objectively. Therefore, to support the argument of a link between method application and result usefulness, the author can report that no 'competing' change agendas were present during any of the case studies, indicating that the achieved results ought not to be accredited to other resources. In other words, even though many factors are clearly influential, no other factors seem to be attributable to the achieved usefulness.

(6) *Usefulness beyond example problems:* It has been the strategy of the author to make repeatability of the research results probable, by proving their usefulness across several companies and industries. Section 4.2.6 will elaborate on the reflection of usefulness across different industrial settings and the factors differentiating these settings.

2.6 Research plan

This chapter will elaborate on the research plan by outlining the case studies and the validation strategy used, the main research stages, validation strategy for the contributions, as well as the research exchange undertaken during the studies.

2.6.1 Case studies

The research project behind the thesis includes in total 23 case studies conducted in Scandinavian based companies from 2009 to 2013.

Several common denominators exist among the 12 case companies:

- All companies are developing and producing products of mainly mechanical nature with varying degrees of electronic and software content
- All companies are selling products globally
- Most companies have production facilities globally
- Many companies carry out development activities globally
    - All of them have their main R&D site in Scandinavia

The case companies are selected across 11 industries based on the following criteria:

- Fit between company challenges and research objectives
- Existence of a relevant ongoing development activities
- Access to relevant contacts within the organization

The two latter criteria were evaluated during opening talks with the relevant industrial contacts before the case studies were initiated.

From Andreasen (2009), it is discussed whether it is justifiable to claim that an ideal research institution should master 'best practice' (Finger and Dixon 1989). It is the conviction of the author that a deep understanding – and if possible mastering – of the practical work will improve the quality of research contributions. This is part of the reason why the author has undertaken 23 case studies during the research presented in this thesis.

Anonymization of cases

An unfortunate factor of conducting research within the *hearts* of the case companies' development activities is that there is a trade-off between access to data and willingness to make results public. Therefore, the decision was taken quite early during this research project that access to the most delicate and confidential information inside the actual development activities, and access to the most experienced resources, was more important than the ability to make all case details public afterwards.

A clear benefit of this strategy is that the author was permitted to work with genuine cases of central importance to the case companies. This circumstance left no need for fabrication of imitated challenges to test method and tools. Another clear benefit is the access to the best company resources, as these are most often selected for the most challenging and critical projects.

Product icon	#	Product types	Business type	Industry	Project/ product based	Description	Research focus	RQs and hypotheses	Duration
	1	Mechanical	Professional	Energy	OEM/ Project-based	Coordinating the development of product and production architectures in engineer-to-order (ETO) manufacturing companies	• Architecture for OEM • From ETO to configure-to-order (CTO) • Definition of product and production architectures	RQ-A2	8 months
	2	Facilities	Professional	Construction	Project-based	Identification of an architecture of a sub-area in construction of residential buildings	• Architecture modeling • Process architecture	RQ-A2	2 months
	3	Electronic/Mechanical	Consumer	Telecommunications	Product-based	Optimization of product configuration process by making a clear definition of the market architecture	• Product configuration • Identification of market architecture	RQ-A2	13 months
	4	Mechanical/Electronic	Consumer	Automotive	Product-based	Architecture assessment and test of modeling techniques	• Architecture assessment • Identification of market, product and production architectures	RQ-A2 H-A1 H-A2	6 months
	5	Electronic/Mechanical	Professional	Industrial infrastructure	Product-based	Support for selection of architecture	• Architecture assessment • Architecture selection	RQ-A1 H-A1 H-A2	3 months
	6	Mechanical/Electronic	Professional	Industrial applications	Project /product-based	Technology and architecture development in parallel	• Technology architecture	RQ-A1 H-A1 H-A2	7 months
	7	Mechanical/Electronic	Professional	Industrial applications	Project/ product-based	Method consolidation for architecture work in front-loading phase	• Method consolidation • Proactive modeling of architectures	RQ-A1 H-A1 H-A2	7 months
	8	Mechanical	Professional	Energy	OEM/ Project-based	Architecture assessment	• Identification of architecture potential	RQ-A2 H-A1 H-A2	6 months
	9	Mechanical	Professional	Energy	OEM/ Project-based	Support for architecture identification	• Support for identification of architecture • Architecture for OEM	RQ-A1 RQ-A2 H-A1 H-A2	12 months
	10	Mechanical/Electronic	Consumer/Professional	Machinery for consumer applications	Product-based	Product program complexity reduction (I)	• Complexity management	RQ-B1 H-B1 H-B2	5 months
	11	Mechanical/Electronic	Consumer/Professional	Machinery for consumer applications	Product-based	Supply chain complexity reduction (I)	• Complexity cost reduction workshops at key suppliers	RQ-B2 H-B2	4 months
	12	Mechanical/Electronic	Consumer/Professional	Machinery for consumer applications	Product-based	Supply chain complexity reduction (II)	• Supply chain reconfiguration • Postponement of customization point	RQ-B1 H-B2	4 months

	13	Mechani-cal/Electr onic	Con-su-mer/Prof essional	Machinery for consumer applications	Product-based	Product program complex-ity reduction (II)	• Complexity management	RQ-B1 H-B1 H-B2	4 months
	14	Mechani-cal/Electr onic	Profes-sional	Machinery for industrial and commercial applications	Product-based	Product program complex-ity reduction (III)	• Complexity management	RQ-B1 H-B1 H-B2	4 months
	15	Mechani-cal/Electr onic	Profes-sional	Machinery for industrial and commercial applications	Product-based	Product program complex-ity reduction (IV)	• Complexity management	RQ-B1 H-B1 H-B2	4 months
	16	Mechani-cal/Electr onic	Profes-sional	Machinery for industrial, commercial and consumer applications	Product-based	Support for proactive architecture development	• Proactive modeling of market and product archi-tectures	RQ-A1 H-A1 H-A2	4 months
	17	Mechani-cal/Electr onic	Profes-sional	Machinery for industrial, commercial and consumer applications	Product-based	Support for proactive architecture development	• Proactive modeling of market and product archi-tectures	RQ-A1 H-A1 H-A2	4 months
	18	Mechani-cal	Con-sumer	Medical devic-es	Product-based	Support for proactive architecture development (I)	• Proactive modeling of product and production architectures	RQ-A1 H-A1 H-A2	4 months
	19	Mechani-cal	Con-sumer	Medical devic-es	Product-based	Support for proactive architecture development (II)	• Proactive modeling of market, product and pro-duction architectures	RQ-A1 H-A1 H-A2	6 months
	20	Mechani-cal	Con-su-mer/Prof essional	Construction	Product-based	Strategy for complexity management	• Definition of themes for complexity reduction	RQ-B1	3 months
	21	Mechani-cal	Con-sumer	Furni-ture/home accessories	Product-based	Complexity management	• Complexity reduction based on the calculation of complexity costs	RQ-B1 RQ-B2 H-B1 H-B2	5 months
	22	Electron-ic/ Mechani-cal	Con-su-mer/Prof essional	Business and consumer electronics	Product-based	Architecture assessment	• Identification of architec-ture potential	RQ-A2 H-A1 H-A2	7 months
	23	Electron-ic/ Mechani-cal	Con-su-mer/Prof essional	Business and consumer electronics	Product-based	Support for proactive architecture development	• Proactive modeling of market and product archi-tectures	RQ-A1 H-A1 H-A2	4 months

Table 1 – Overview of case studies

2.6.2 Research stages

Problem-based and theory-based stages

Research questions B1 and B2 are primarily theory-based but were not developed before several problem-based research activities were conducted.

Research questions A1, A2 and A3 is primarily problem-based, but has benefitted from loops of theory-based tracks influencing mainly the synthesis phase of the problem-based tracks.

DRM stages

Figure 8 shows the research stages according to the DRM framework.

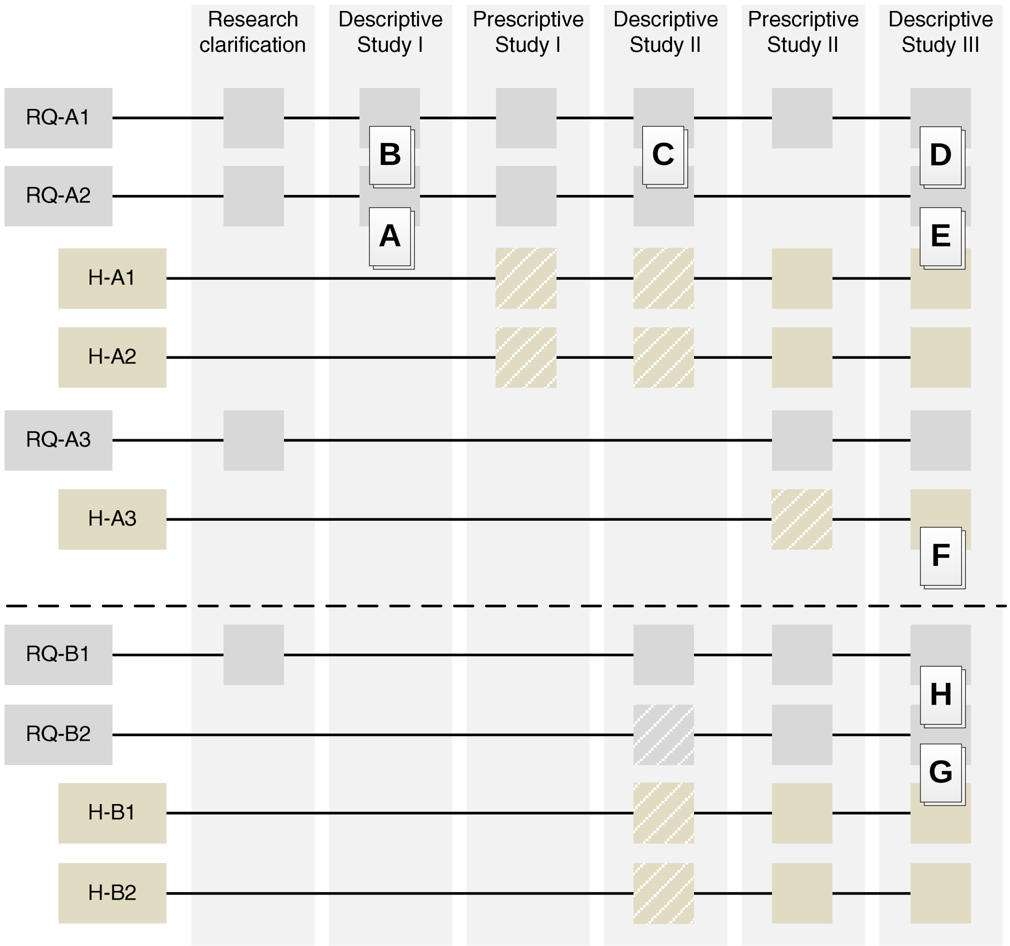


Figure 8 – DRM stages

Figure 8 describes how the DRM stages have been conducted from the research clarification phase to Descriptive Study I, II and III with Prescriptive Study I and II in between. The Descriptive Study I and Prescriptive Study I did not specifically address the research questions B1 and B2. This first loop of descriptive and prescriptive studies was dedicated to the area of program architecture identification, and the result was papers A and B. The Descriptive Study II revealed the necessity of including the research questions B1 and B2 in order to address the area of complexity reduction, and Descriptive Study III resulted in the reporting of the final five papers D, E, F, G and H.

Table 2 provides an overview of the research stages, their level of extent (initial/comprehensive), degree of literature review, case studies, and reporting of results.

DRM stage	Research activities		
	Literature	Case studies	Reporting of results
RC	Review		
DS-I	Comprehensive review	#1 #2	Paper A (journal) Paper B (conference)
PS-I (initial)		#3 #4 #5 #6	
DS-II (partly initial, partly comprehensive)	Review	#7	Paper C (conference)
PS-II (comprehensive)		(see below)	
DS-III (initial)	Review	#8 #9 #10 #11#12 #13 #14 #15 #16 #17 #18 #19 #20 #21 #22 #23	Paper D (journal) Paper E (journal) Paper F (conference) Paper G (conference) Paper H (journal)

Table 2 – Overview of research activities and DRM stages

Research clarification and DS-I

The research clarification phase consisted of a broad literature review within the research field – see Figure 4 – in order to gain a broad understanding of the state-of-the-art within central and closely related fields of research.

The evaluation of existing methods was used in paper A and paper B, where a number of previous case studies from the research group of the author were used as a basis together with two additional case studies (#1 and #2) which were used to gain an understanding of how the current state-of-the-art methods were working in the field.

The outcome of these first phases was a broadened understanding of the current state-of-the-art including field testing of the most recent methods. These activities revealed the methodological gap that was used to formulate and fine-tune the research questions.

PS-I and DS-II

The first prescriptive loop was performed and tested in case studies #3 to #6. Case study #7 included a series of method consolidation workshops within the case company of case study #6, in order to evaluate how to use the proposed methods going forward across development projects of varying size and extent. The status of the research work at this point in time was the formulation the need for modeling the market architecture, product architecture and production architecture in coordination – in order to achieve *alignment*. The result of this was paper C, as the research loop revealed the clear need for the deliberate and formalized inclusion of the market aspect in product program design.

DS-II revealed the need for further method development and more case studies for field tests. This was based on the notion that the separation between constitutive/structural aspects and behavioral/functional aspects of an architecture for a product program was still not clear from the research results (hypothesis A2).

DS-II also revealed the need for the inclusion of the quantification aspects in terms of complexity cost calculations, in order to be able to justify architecture initiatives and put a price on the benefit of cleaning up among unprofitable product variants within a product program. In that way, research questions B1 came into play and research question B2 was formulated to further integrate the two research areas of program architecture identification and complexity reduction based on quantification of complexity costs.

PS-II and DS-III

The second prescriptive loop included extensive research activities in terms of conducting case studies to address the five research questions. Apart from the previous prescriptive research stage, this stage was followed by a descriptive study which can be characterized as partly initial and partly comprehensive. For example, the result of case study #9 was verified using a thorough quantitative evaluation of the benefits achieved through the identification of an architecture for the OEM company. And for the research area of complexity management, the reduction of complexity costs was estimated using detailed evaluation techniques and supplemented by supply chain oriented supplement case studies (#11 and #12) in order to assess the full complexity cost reduction potential of the product program. A supplementary literature review was also conducted in order to relate the results to the state-of-the-art within available literature.

2.6.3 Research exchange

As an important source of research evaluation, a number of research exchange sessions have been carried out during the research period. The research exchange covers lectures, courses and external seminars which have all been used to exchange ideas and thoughts about the research areas in order to include the opinions and viewpoints of other practitioners, advanced students and fellow researchers.

Lectures

The lectures cover both program architecture identification and complexity cost calculations talks comprising methodology, highlights from case work and reflection. The talks have been held at The Technical University of Denmark and Aalborg University.

- The Technical University of Denmark (DTU): Mass customization course
- The Technical University of Denmark (DTU): Technological platforms and architectures
- Aalborg University (AAU): Master in Management of Technology (MMT)
- Aalborg University (AAU): Business and Management Academy

These lectures cover both specialty courses within the area of architectures and complexity reduction, but also mass customization courses focusing on the potential improvement of ordering and specification processes by having a clear program architecture in place.

Courses

Courses include engineering research courses, systems engineering courses and a management technology course, where the architecture modeling techniques were analyzed as being management instruments used to incite the identification of a program architecture.

- The Technical University of Denmark (DTU): Research and PhD studies (2010)
- Universitè du Luxembourg/Technical University of Ilmenau/The Technical University of Denmark: Summer School in Engineering Design Research (2011)
- Technische Universität München (TUM): Spring School on Systems Engineering S3E (2011)
- Copenhagen Business School (CBS): Management technology, inter-organizational relations and performance management (2011)

- The Technical University of Denmark (DTU): Systems Engineering (2012)

External seminars

In the occasion of research exchange, a number of seminars have been conducted with similar research groups from the top research groups of Sweden and Germany.

- Chalmers University of Technology: Research seminar
- Technische Universität Hamburg-Harburg (TUHH): Research seminar

Beside exchange sessions mentioned above, a number of talks have been given in other formal and informal occasions.

Conference attendance

As a strong supplement to the means of research exchange mentioned above, a number of conferences has been attended within the research field. The conferences include:

- Produktudviklingsdagen 2009
- NordDesign2010
- Produktudviklingsdagen 2010
- ICED 2011
- Produktudviklingsdagen 2011
- NordDesign2012
- Radikal Forenkling 2012
- DTU Design & Innovation 10 year anniversary

Figure 9 – DTU Design & Innovation 10 year anniversary

Peer-reviewed papers have been accepted and presented for most of the major conferences as an excellent means of establishing international contacts and most importantly enthusiastic discussion partners (see section 4.1).

All conference presentations have been assisted by large posters (see section 3.2 for elaboration on the role of visualization). Figure 10 show a presentation from NordDesign 2012 of the 5-step approach to complexity cost reduction.

Figure 10 – NordDesign 2012 presentation

3 Theoretical basis

The research is based on fundamentals from system theories relevant for engineering design science, design theories of architectures and product families and theory of management technology. The aim of Part 3 is not to paraphrase the theories on which this research work is founded, but rather explain the guidance and key implication that the theoretical basis represents.

3.1 System theories

3.1.1 Theory of Technical Systems

Origin

The Theory of Technical Systems (TTS) is proposed by Hubka and Eder (1988) interpreting classical systems theory into the field of engineering design. TTS also embraces the fundamentals of cybernetics into the field of engineering design (Ashby 1956). For instance, Ashby's work includes *The Law of Requisite Variety* stating that "only variety can destroy variety". In this context the law is interpreted as it is not possible to eliminate variety as such, but possible to eliminate inappropriate variety by providing the needed variety in a smarter way.

Theory

In short, TTS is a general theory providing a pattern of explanation for the nature of technical systems, their purpose, design, operation and interaction with its surroundings. *Products* can be seen as technical systems, as well as several products can, even though the TTS has a number of limitations regarding the description of programs of products.

The process of creating the desired state of the *operand* is called a *technical process*. The technical process is a subset of a transformation process including namely technical systems. The human system and technical system represent the execution system, whereas the information system and management and goal systems represents a sort of a control system.

The technical processes are divided in a preparation phase, execution phase and a delivery phase named the 'finishing phase'. The operand goes through the technical process and a transformation is carried out taking the operand from the existing state to the desired state. Effects and feedback is exchanged between operand and the active environment as a result of the interaction in the technical process.

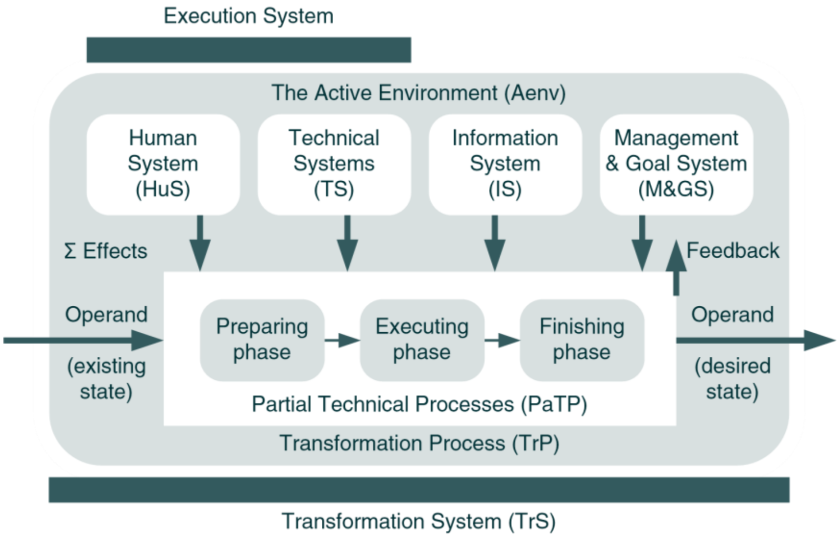


Figure 11 – Theory of Technical Systems - The Transformation System

Classification systematics

TTS also provides a series of systematics to classify technical systems. An interesting classification is by degree of abstraction, which is highly relevant for the modeling techniques used for identifying an architecture.

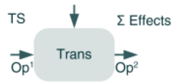
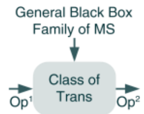
	General Hierarchy		
Level of concretiza-tion	**Designation of Level**	**Defined by (Graphical Model)**	**Established Design Characteristics**
0	Machine System (MS)		MS with mainly mechanical mode of action
0.2	Phylum of MS		- Phylum of Operands - Class of Transformations
0.4	Class of MS	Detailed Black Box Sketch of the Technological Principle Basic Function Structure	- Family of Operand - Technological Principle of Transfor-mation - Necessary Input Effects and thereby the basic Functions
0.6	Family of MS	Detailed Function Structure Rough Component Structure Concept Sketch	- Species of Operand - Function Structure - Inputs to MS - Families of Function-carriers (or-gans) - Combination and Basic Arrangement of Function-carriers
0.8	Genus of MS	Drawing of Component Structure Drawings of Common Compo-nents and Subassemblies	- Complete Parts - Arrangement - Partial Form - Some Dimensions - Types of Materials - Some Tolerances and Surface Property Dimensions
1	Species of Serial Size	Complete Set of Workshop Doc-umentation	Total and Definitive Specifications for Parts and Arrangements. For all parts: - Forms - Dimensions - Materials - Manufacturing Methods - Tolerances - Surface Properties

Table 3 – Levels of Abstraction of Technical Systems and their Representation (Hubka and Eder 1988)

Table 3 shows six different levels of concretization ranging from the highest level of abstraction (Level 0) to the lowest level of detail (Level 1).

Impact for this research

Concerning the research area of program architecture identification, the classification systematics of concretization level provides a framework allowing for an efficient modeling of the critical aspects of architectures. Efficient is mentioned due to the variable concretization level that enables the modeling activities to adapt the level of abstraction to comprise the necessary level of detail – and not more. This is central as a basis for modeling critical aspects of architectures, as the detail level can very easily increase exponentially making it impossible to do efficient identification of an architecture, and also making it impossible to identify critical links between market, product, and production aspects of program architectures.

In general, TTS provides a reference framework for all descriptions of products and the interacting systems, which are central in working with architectures. TTS' clear description of the technical process being an interplay between the technical system and the all the other elements clearly defines the need for differentiating between the structural/constitutive aspects of technical systems and the functional/behavioral aspects. This is further elaborated in section 3.4.3. Whereas in TTS, this notion is highlighted for single technical systems, the author does not see any complications using this concept for compound technical systems, namely entire product programs, as this situation is analog to the single technical process – TTS being recursive in this sense. A program architecture of a product program cannot and should not be evaluated based on the structural/constitutive aspects themselves, but by including which effects it has on the surrounding systems.

3.1.2 Theory of Domains

Origin

The Theory of Domains (ToD) is originally proposed by Andreasen (1980), and has later been refined and developed as a supplement to the emergence of TTS to support an engineering design context. According to the latest revisions of the ToD (Andreasen et al. 2014, Hansen and Andreasen 2002), this theory proposes three different viewpoints (here 'domains') necessary to be able to reason properly about a product.

Theory

The three viewpoints are

- *A transformation domain (later nominated as 'activity' domain)*

  Equivalent to the transformation system of the TTS, the transformation domain represents the viewpoint of which the transformation of operands is considered.

- *An organ domain*

  This domain has equivalence to the function-carriers of the TTS, and the relation between the functions and the organs (function-carriers) are considered here. The link to the transformation domain is that the organs are carrying out the functionality needed to carry out the transformation.

- *A part domain*

  This domain is answering the question of how the product is physically built up. The link to the organ domain is that the decomposition of parts can be done taking the starting point of the organ structures.

As the three domains are to be considered as viewpoints, each view requires the abstraction from the other views in order to allow for structural and behavioral reasoning. This is further elaborated by Mortensen (2000).

Impact for this research

The ToD serves as a powerful foundation for describing which fundamental viewpoints from which reasoning about products can be carried out. Many other viewpoints can be derived as combinations of the basic ones from the theory, making the ToD a powerful basis for deriving useful and efficient ways of analyzing and evaluating products and product programs.

Each of the views being abstractions in themselves puts a challenge to linking of the views. However, it is exactly the linking that makes the ToD applicable for the research conducted here. For instance, by observing how a transformation impacts the organs (function carriers) that again impacts a set of parts enables the evaluation of the appropriateness of which functions are realized. And vice versa, by observing parts and their implementation it is possible to trace the impact on function carriers all the way to the transformation, and ask whether the value of the transformation can justify the presence of the part, and whether this is provided in the best possible way.

The Product Family Master Plan, or simply PFMP, is a further development of the ToD (Harlou 2006), merged with the framework for Integrated Product Development (Andreasen and Hein 1987), and adapted to the industrial context of development of product families. The PFMP demonstrates similar strengths which are realized by identifying the causal relations between transformations and parts.

3.1.3 Theory of Dispositions

Origin

The Theory of Dispositions (TD) is proposed by Andreasen and Olesen (1990) and it is suggests that a number of dispositions are made in engineering design that affect the transformations encountered by the product later on. The TD was further expanded by Olesen (1992) as a very central basis for many newer engineering design theories.

Theory

Figure 12 represents the TD in a score model. It represents two important concepts of the TD that needs further elaboration, namely the dispositional mechanisms and the dispositional areas.

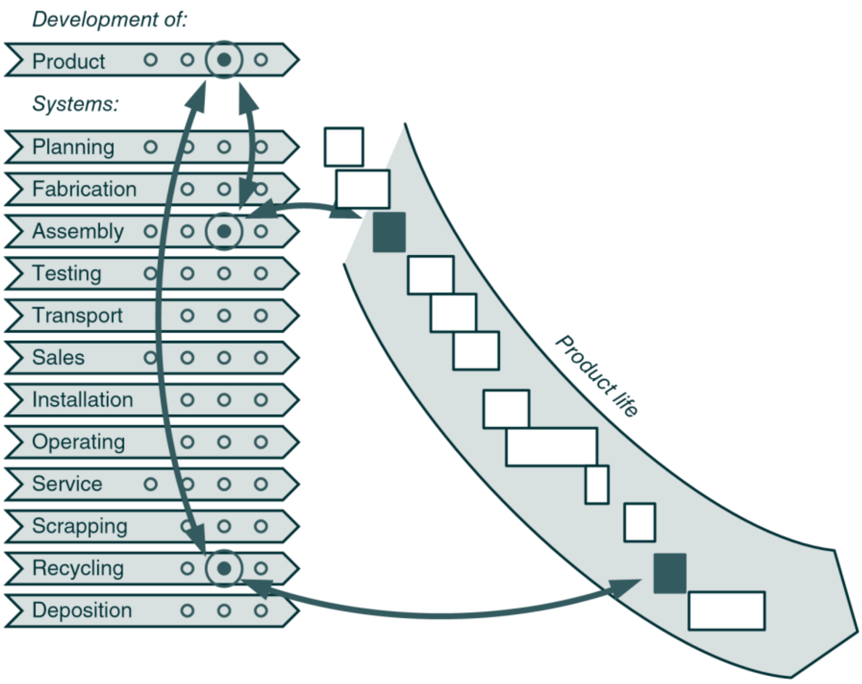


Figure 12 – Theory of Dispositions - a score model (Olesen 1992)

TD suggests a number of dispositional mechanisms that cover different types of dispositional effects:

- *A strategy effect*, capturing the strategic decision-making of e.g. technologies

- *A group effect*, capturing the learning effect arising from implementing e.g. group technology principles
- *A standardization effect*, capturing the repetition effects from re-use and limited variation of modules, components, materials, process instructions, methods etc.
- *An optimization effect,* capturing the effects of an optimal fit between e.g. product and production parameters to enable optimization
- *A resource effect*, capturing the effect of optimal exploitation of staff and machines
- *A correctness effect,* capturing the necessity of having correctness of data as a prerequisite for achieving the previously mentioned effects

A central concept of the TD is that these effects are realized in so-called *meetings* between the product and the life-phase system hosting the transformation process the product is part of.

It is during the meetings that the dispositional effects should be obtained, and TD also suggests a number of dispositional areas serving to classify where these effects typically arise:

- *Cost dispositions* directly concern the cost structure of products, as all other dispositions of course can be measured as having a cost effect (see Figure 13).
- *Process and equipment dispositions* are related to the relations between the product design and the equipment of production processes.
- *Assembly dispositions* are related to the relations on several levels of a product hierarchy of programs, families, structures, and components.
- *Quality dispositions* are related to the effects that product design has on product quality level including robustness towards production tolerances etc.
- *Other dispositions* include production ramp-up/transfer, service, investments and last but not least, sales where properties and features of the products play a central role

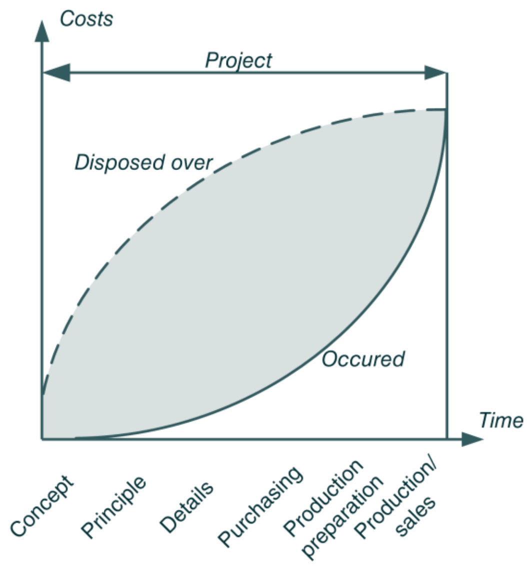


Figure 13 – Example of a disposition area: Cost

Impact for this research

The impact for this research is widespread and fundamental in the sense that a successful architecture of a product program needs substantial and comprehensive dispositions during design phase to ensure an optimal fit towards the life-phase systems for a range of products. Both the dispositional areas and the dispositional effects are considered a vital basis for the research presented here, as the identification of a program architecture can be seen as an activity of identifying, balancing and prioritizing dispositional areas and dispositional effects for a product program.

3.1.4 Contingency theory

Origin

Contingency theory is a behavioral theory that originated from systems theory (Galbraith 1973, Lawrence et al. 1967). Even though contingency theory is today primarily used within organization theory, the fundamentals derived from systems theory are continuously relevant for the research presented here.

Theory

Contingency theory describes in short that there is no universally optimal solution to be found, and that the optimal solution is contingent to the internal and external situation on both sides of the system boundary. Analogous to TTS contingency theory explains that no one single design can be the best in all cases and that the best design depends on the character of the environment (TTS: active environment).

Impact for this research

The impact of contingency theory for this research is that it is not possible to derive universal structuring for architectures, as these would be contingent to the situation or challenge that the architecture initiative should respond to. This does not mean that architectures of product programs should not strive for the fulfillment of the universal virtues of cost, throughput time, quality, efficiency, flexibility, risk and environment (Olesen 1992), and neither does it prohibit the formulation of universal virtues of program architectures, but it underlines that there is no such universally optimal algorithm for contextually independent balancing of these virtues.

3.2 Complexity theories

From Latin *complexus* means 'to embrace'. At the time of writing, the term *complexity* has become complex in itself, *embracing* a huge amount of different multi-faceted definitions within engineering science. Most of them are undoubtedly relevant for various specific purposes. However, the complexity of relevance to this research work is the complexity arising from the meeting between the product and the life phase systems.

Wilson and Perumal (2009) denotes this complexity the 'product-process' complexity, which is in line with the relativity between the product and its surroundings – a central aspect of TTS, ToD and TD. From these theories, it can be derived that describing a product or a process as complex solely by regarding the product or process characteristics themselves has a very limited significance, as the interplay *between* the product and the process is excluded. In other words, what can seem complex from a product perspective is not necessarily complex if a capable process is in place to maintain control. Therefore, the focus here is the complexity arising from the inappropriate product-process interplay. And this is a type of complexity that comes with a cost.

Several theoretical fields are dealing with complexity, but the ones considered here are engineering design science, research within operations management, and research within cost accounting.

3.2.1 From engineering design science

Origin

Several schools from within engineering design science have dealt with complexity. As the area in itself is not particularly mature, there is no wide, accepted and consolidated body of knowledge to provide a rigorous point of departure. Instead, a number of basic contributions have touched upon the subject with different perspectives, underlying theoretical foundations and purposes. It is therefore the aim of this section to point at those contributions which the author considers fundamental for the point of departure of this thesis.

From a simplistic point of view, the complexity term is used in two different ways across the field of engineering design science:

- One track represents the belief that complexity can be identified, described, assessed and even quantified based solely on analyzing the products themselves, where complexity is interpreted merely as a characteristic of a product program, e.g. in terms of multiplicity of parts or product variants.
- Another track represents the belief that complexity is a relative phenomenon arising from the unaligned *meetings* occurring between the products of a product program and the respective life-phases.

Also, the viewpoint of the ascertainment of complexity varies between contributions from engineering design science:

- One group of research is focusing on complexity from the engineering practitioner's point of view encountered while designing. Mainly relevant in term of proactive complexity prevention.
- Another group is focusing on the *resulting* complexity to be managed and reduced from a management point of view. Mainly relevant in terms of reactive complexity reduction.

Theories

Despite the differences in viewpoints and focus of research in complexity within engineering design science, several authors recognize the important distinction between the complexity that can be recorded in real life, and the complexity being a mental perception of a situation.

From Axiomatic Design, Suh (2005) distinguishes between

- Real complexity
- Imaginary complexity

Imaginary complexity is the complexity arising from misconceived uncertainties that could be eliminated by improving the designer's understanding of the design at hand. Real complexity is defined as the measure of uncertainty arising from a design not fulfilling Suh's Independence Axiom, which is a composed measure of the decoupling between functional requirements and design parameters among others, or in the interpretation of the author, popularly speaking equivalent to the reciprocal level of *modularity.*

Andreasen (2009) distinguishes between

- Object complexity
- Mental complexity

Mental complexity being an observation filtered through the mind's understanding including all the personal and conceptual bias included here. Object complexity being the inherent level of composedness and difficulty of objects. Andreasen (2009) sees mass customization and multi-product development as a huge step towards higher complexity due to the task difficulty of managing such concepts, but at the same time as a powerful means of reducing mental complexity by creation of transparency. And also as a powerful means the lower the cost *of* complexity when modularization and modular architectures are properly aligned with the life-phase systems of suppliers, manufacturing processes etc. In other words, there is a distinction between the apparent complexity levels, which can be allowed to increase in the favor of a cost reduction.

The German school of variant management applies a very hands-on approach to complexity. Exemplified by Rathnow (1993), complexity is interpreted as the negative effects of unsuccessful management of product variants, stating that a certain degree of complexity is the optimum in providing the necessary product variance to

the market.
Table 4 shows a list of sources of complexity costs.

Functional area / Product life cycle	R&D	Procurement	Production	Marketing & Sales	Customer service
Development cycle	• Drawings • BOMs • Tests	• Search for and evaluation of additional suppliers	• Additional tooling • Additional work plans	• Additional training • More complex pricing	• Additional documentation • Additional training
Market cycle	• Adapting variants to technical or other changes	• Decreasing order volumes • No volume rebates	• Costlier production control • Longer set-up times • Larger inventory • Costlier quality control	• Larger finished goods inventory to maintain supply capability • More errors in order processing	• Decreasing “fix is right first time” quota
Disposal cycle	• Clearing up of data	• Costlier planning of product approval	• Disposal of tools and other operating resources	• Costlier planning of product withdrawal	• Spare parts inventory 5-10 years after product withdrawal

Table 4 – Potential sources of complexity costs (Rathnow 1993)

Along the line of Rathnow, fellow countryman Schuh (2001) differentiates between external complexity (e.g. variance in customer requirements) and internal complexity (e.g. the response to these requirements in terms of product variants), stating that complexity cost is the result of a too complex internal response to the external complexity.

Lindemann et al. (2009) presents another basic contribution to the management of complexity by distinguishing between

- Market complexity
- Product complexity
- Organizational complexity
- Process complexity

Lindemann et al. (2009) furthermore distinguishes between *complicatedness* and complexity from the basic definition from cybernetics (Wiener 1948), where the system's dynamics decides whether it is complicated (stable over time) or complex (dynamic over time). Complexity is seen as an attribute of systems consisting of numerical, relational, variational, disciplinary, and organizational complexity.

Impact for this research

See section 3.2.4.

3.2.2 From operations management research

Origin

From the field of supply chain management and operations management the approach to complexity is partly gathered within the field of *complexity management*. The basic purpose is here to evaluate and quantify complexity through various sophisticated quantification techniques and to ascertain, manage, and control complexity through various methods and frameworks. The origin of this research is based on two different points of entry, namely either the *product portfolio*, or the supply chain and operations setup seen as a *system*.

Theory

As representatives of the state-of-the-art within this area, a few of the main contributors' definitions are mentioned below with a supply chain and operations focus from the *product portfolio:*

- Closs et al. (2008) defines complexity of a product portfolio as *a state of processing difficulty that results from a multiplicity of, and relatedness among product architectures design elements.*
- Jacobs (2013) and Jacobs and Swink (2011) define product portfolio architectural complexity as *a design state manifested by the multiplicity, diversity, and functional interrelatedness of products within the portfolio.*

Other representatives are regarding the *supplier-customer system* as the starting point setting the system boundary from which the complexity term is taken in use:

- They see complexity more simply *as the variety and uncertainty of a system*, and differentiates between *structural complexity* being the variety embedded in the static system and *operational complexity* associated with the uncertainty of the dynamic *supplier-customer* system (Wu et al. 2007, Sivadasan et al. 2002, Calinescu et al. 2000, Frizelle 1998, Frizelle and Woodcock 1995).

Hence, there is no consensus on the definition of system boundaries for the analysis and approaching of complexity within the field of supply chain and operations research, as there seem to be two different starting points – the product portfolio or the supply chain and operations system, even though a number of authors recognize the link (Langenberg et al. 2012, Perona and Miragliotta 2004).

Only very few contributions from the supply chain and operations field declare themselves in compliance with any technical elaborations or enhancements of systems theory, such as TTS. This could be a possible explanation for their definitions attempting to decouple the product portfolio characterization from the process and operations difficulties arising from the product portfolio complexity. The decoupling is intriguing in itself as it allows the separate description and classification of the problem area of the product and the problem area of supply chain and operations – which improves the applicability of quantitative research, and theoretically allows generalizations within each of the problem areas. However, the decoupling is not in line with the relativity between the product and its surroundings described in TTS, ToD and TD.

Aside from the definitions, almost all contributions within the field share the notion of product program complexity being a limiting factor for the optimization of supply chain and operations.

Impact for this research

See section 3.2.4.

3.2.3 From cost accounting research/quantification of complexity costs

Origin

From developments within cost accounting several sophisticated methodologies have been developed attempting to identify costs of complexity. One such inevitable contribution is activity-based costing (Cooper and Kaplan 1988), which has been heavily debated for the last two to three decades. When introduced, activity-based costing was intended to replace full-cost models that assumed direct cost-proportional allocation of overhead costs.

Theory

Cooper and Kaplan (1988) suggested activity-based costing as a new method to avoid the deficiencies of rather arbitrary allocation of overhead costs. This is done by allowing the allocation of indirect costs first to activities and hereafter to individual orders, customers or even products if sufficient data is available. Following this allocation technique, the method takes its starting point in the resources used and links these to activities and then to cost objects of different type.

An array of further developments of the activity-based costing basis has been developed that are relevant for the research presented her (Lechner et al. 2011, Park and Simpson 2008, Anderson and Kaplan 2007), due to their addressing of two major critiques of the method:

- *Activity-based costing being too time consuming.* The benefits of the method do not justify the resource consumption used for implementation – recognized by the authors themselves (Anderson and Kaplan 2007).
- *Activity-based costing does not capture the cost of complexity.* "Activity-based costing faces a problem in that it assumes a linear relationship between activity costs and the products consuming them. Excess costs caused by product variety are therefore hard to allocate to each product in a family" (Park and Simpson 2008) from Fixson (2004).

Activity-based costing is therefore seen as a central basis for method development within identification of complexity costs. However, the relevant criticism mentioned above serves as input to the research work presented here.

Impact for this research

See section 3.2.4.

3.2.4 Conclusion on complexity theory

Impact for this research – engineering design science

Inspired from engineering design science and Wilson and Perumal (2009), the following interpretation of complexity by the author suits the overall objectives of the research presented here:

> Complexity is a relative phenomenon arising from the unaligned *meetings* occurring between products and processes, when this lack of alignment it attributable to the product-process system having too many parts, solutions, product variants or business processes

'Business processes' include the supply, production, and delivery processes in the manufacturing flow as well as specification and development processes in the order flow.

For this research, there is interest in both methods for *proactive* complexity prevention and *reactive* complexity reduction.

In terms of grasping complexity as either being of mental type or object type, the impact for this research is that both types play a role, as mental complexity is an important parameter for e.g. mobilizing complexity reduction or architecture initiatives, but that it is the object complexity arising from the unaligned meetings between products and processes that play a defining role in respect to reduction of complexity costs. And it is precisely this complexity that is of interest in this thesis – complexity seen from an optimization viewpoint.

In the works of Simon (1962) it is stated that complexity of a structure depends on the description – including purpose and concretization level etc.

"How complex or simple a structure is depends critically upon the way in which we describe it" (Simon 1962)

Even though this might seem a truism, it is relevant for the optimization viewpoint on complexity. Optimization being the justification of the concept of complexity, the author has focused on complexity as the relative characterization of a product-process relation, making it meaningless to characterize a product-process relation as being complex, if there are no possibilities or prospects for improving the product-process relation into a less complex one. This might seem as a perfunctory contemplation, but it is absolutely central to this research's perception of complexity, and its close interrelatedness to the architecture of the product program, as the author regards the very architecture of the product program is the single most powerful key towards sustainable reduction of complexity.

From an aggregated system level, this perception of complexity can be denoted as a complete characterization of a product-process system, where the *level of system complexity* can be seen as *relative* to another configuration or version of the same product-process system. This is based on the notion that it can seem meaningless to denote a system consisting of a product and process setup as being *complex*, without providing a relative measure to compare with. Therefore, one could denote this 'the level of *excess* system complexity', as this is the interesting measure. And this excess system complexity comes with a cost, which is here denoted as the *complexity cost reduction potential:*

The complexity cost reduction potential is directly associated with the level of excess system complexity of the product-process system

Impact for this research – research in operations management

From research in supply chain and operations management, the impact for this research is that the interrelatedness of product and process interplay is not well understood in terms of research in complexity. Yet, this interrelatedness is difficult to study as it is the result of very context specific configurations of product and process setups. Additionally, the concept of coordinated development of product and processes is not a well-established theme within this research area, making it intriguing to pursue the development of various optimization techniques that are focusing mainly *either* on product program complexity *or* supply chain and operations complexity.

An important notion from research within this area is that a very wide body of knowledge acknowledges the negative effects of product program complexity on the optimization of supply chain and operations – regardless of the complexity definition.

Impact for this research – cost accounting research

From cost accounting research it is clear that numerous sophisticated cost allocation techniques exist. A theoretical basis that is a helpful supplement for the detection, identification, quantification and allocation of complexity costs.

3.3 Modeling as a management technology

This chapter includes different theories centered on explaining the nature of modeling of architectures seen as a *management technology*. The concept of management technologies is taken from the cross-field of research between managerial and sociological type, and is exemplified here:

> "In order to make complex and hard things simple and soft, managers need another type of object – management technologies – which are mediators allowing them to operate on the material world from a distance" (Czarniawska and Mouritsen 2009).

This field of research serves to explain important aspects of the nature of modeling of architectures, including purpose, staging, and the interaction between the models and surroundings in companies. Included here is design modeling from an engineering design science point of view, the concept of boundary objects from socio-technical point of view, and visualization seen from socio-technical point of view by examining what constitutes architecture models and which sort of management the models enable in an industrial setting.

This chapter contains quite a wide variety of contributions to explain the nature of modeling of architectures.

3.3.1 Design modeling

Origin

Here, design modeling is seen as a management technology. From the definition of Andreasen (1994) design modeling is seen as the language of the designer, and modeling is seen is an activity aiming for design clarification and verification. A model is here defined as an artifact reproducing properties of an object, and the modeling activity always has an object, property and a purpose.

Theory

When designing programs of products, the design verification process becomes more complicated when a multitude of dimensions is to be verified. Furthermore, design modeling of product programs can reveal evidence that current product programs possess an optimization potential which can be delicate information to an organization.

From Figure 14 it seen that models of different nature can capture reality through the application of theory. These models can be of phenomenological type, information models or computer models.

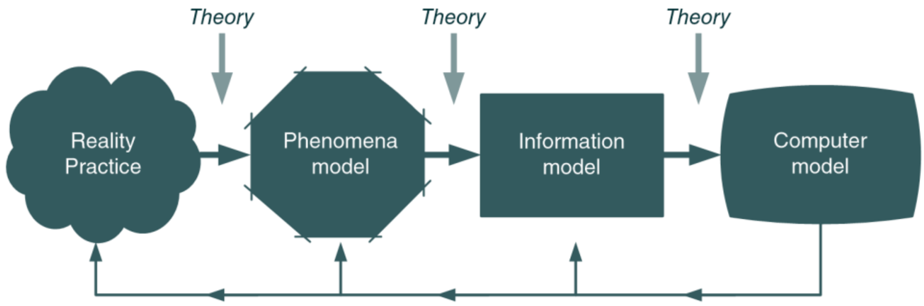


Figure 14 – Deriving models from practice for use in practice (Andreasen 2009, Duffy and Andreasen 1995)

From Andreasen (1994) it is highlighted that it is necessary to create conceptual models of products in order to clarify the fit between the product concept and *functional concept, production concept, assembly concept, operation concept, service concept, environment concept etc.* However, it is also noted that modeling comes with a resource consumption that should be balanced towards the possible gains.

Later contributions from Pedersen (2009) have highlighted the importance of visual modeling in regards to design modeling of product platforms.

Impact for this research

An important notion from the area of design modeling is that the modeling activity always serves a certain purpose. Design modeling of architectures for product programs has a very wide implication in the industrial setting making design modeling a potentially extremely powerful activity and means to pave the way for an improvement. Therefore it is relevant to apply the viewpoint of design modeling as a management technology in the sense that design modeling constitutes a powerful means also for the managerial evaluation of an early stage concept for an architecture of a product program.

Regarding design modeling as a type of management technology underlines the fact that design modeling cannot be seen as a *neutral* activity which is detached from the purpose of the modeling.

Design modeling of an architecture concept reduces the *mental complexity* to enable management to make product program decisions. Therefore it is important to acknowledge the status of design modeling as a type of management technology and regard it as an instrument that can be applied by relevant actors depending on how it is staged in the industrial setting.

3.3.2 Boundary objects

Origin

From socio-technical research, Leigh Star (2010) and Star and Griesemer (1989) invented the concept of *boundary objects* as a means to explain how information can be used and exchanged differently by various communities.

Models of architectures can be seen as a type of boundary object. Therefore it is relevant to take a brief investigation into this concept.

Theory

The authors describe three components qualifying something as a boundary object (Leigh Star 2010)

- *Interpretive flexibility*
  When different practitioners from for example departments of marketing, detailed design and procurement are evaluating an architecture concept, they may have very different interpretations of the concept's ability to fit their individual needs of e.g. commercial attractiveness, design robustness and standardization of purchasing. However, if the architecture model can facilitate an *interpretive flexibility*

of “the same” architecture while suiting the needs of various stakeholders and facilitate a constructive evaluation of the architecture, the architecture model fulfills an important function.

- *Structure of informatics and work process needs and arrangements*
  To qualify as a boundary object, it has to be able to structure information relevant to the interacting stakeholders and fit to their information needs in their organizational arrangement. The driver for setting up the boundary object as a type of infrastructure is to fulfill their information and work requirements as they are perceived locally by each group of stakeholders.
- *Dynamic between ill structured and more tailored uses of the objects*
  An important third cornerstone is the boundary objects ability to facilitate cooperation without the reaching of consensus. This is possible as the boundary object resides *between* the interacting groups in an ill structured way, and when necessary, worked on by local groups making it more specific and more tailored to fit their own needs. The boundary object moves back and forth between these two states.

Boundary objects become part of a ‘life cycle’ characterized by the constant shifting between formal and informal, ill structured and well structured, standardized and unstandardized etc. as depicted in Figure 15.

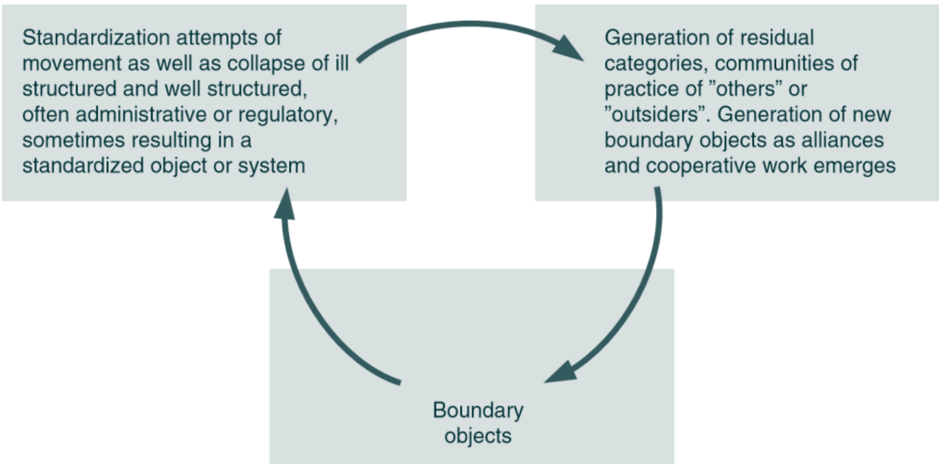


Figure 15 – The ‘life cycle’ of boundary objects (Leigh Star 2010)

Impact for this research

As modeling of architectures can take on the role of a boundary object, it is important to be aware of the implications of having such a role. This can be in terms of allowing a sufficient interpretive flexibility while allowing different groups of practitioners to interact with the models and detail them even further in local groups.

Also recognizing the life cycle of the boundary object, either moving towards or away from formalized structures is an important aspect to include in this research. Examples of this could be the identification of architectures needing new structuring in IT systems, thus requiring the architecture models to move away from existing formalizations and generating new proposals for structures through the conscious use of the architecture models as boundary objects.

3.3.3 The role of visualization (as representation)

From Impact Model A in Figure 2 it is outlined how one criterion of this research is to promote explicitness of the critical aspects of architecture through support. The line of reasoning applied here is that it is very difficult for companies to work with such abstract phenomena as architectures represent, without the support from appropriate models. There are certainly many reasons and factors to consider attempting to explain why the modeling of visually explicit representations of critical aspects of architectures serve as an important means for the identification of architectures, but this section will focus in the fundamental models relevant to the research presented here.

Origin

Two sources of explanation are included here, namely contributions from socio-technical studies of visualization and cognition by Latour (1986) and from organization studies of the role of representation by Cooper (1992).

Theory

Before moving on, from research within *bounded rationality* Simon (1957) has stated:

> "The capacity of the human mind for formulating and solving complex problems is very small compared with the size of the problems whose solutions is required for objectively rational behavior in the real world – or even for a reasonable approximation to such objective rationality" (Simon 1957)

From the concept of bounded rationality it is suggested that the rationality of decision-makers is limited by information level, cognitive ability and time to make the decision. In other words, bounded rationality suggests that there are most often no such rational decisions, as there are always limits to the factors mentioned above.

Cooper (1992) acknowledges the conditions set up by Simon (1957), but explains how *representation* becomes of central importance given these limitations to the rationality of decision-making. This is done by explaining how three concepts of representation, *displacement, abbreviations,* and *remote control,* become mechanisms for managers to control complex and heterogeneous activities.

Firstly, the reduction of a grand empirical space across market, product and production aspects of architectures into a single (or a few) large paper posters which can be brought directly into a meeting arrangement and hung on the wall, can be regarded as an act of displacement. This is the fundamental prerequisite for intervention and malleability. When critical aspects of architectures are represented in a piece of paper, it is possible for practitioners and managers to intervene. Displacing the empirical space enables professionals to draw a line and 'freeze' everything as it is, while the visualization attempts to construct the best possible foundation as a basis for decision making. Cooper (1992) summarizes this as:

> "when information is uncoupled from its action context and represented symbolically, events can be manipulated and combined in new ways, so enabling greater control" (Cooper 1992)

Secondly, illustrations and symbolism can be seen as powerful abbreviations capable of condensing broad concepts into small malleable objects on a poster. Appropriate categorization and classification of objects are central features here, as described in TTS as well. When, for instance, analyzing 500 bills of material, classification holds a key to gather the necessary overview while screening for interesting candidates for product pruning. Several contributions within this field are created to include such specific purpose, e.g. the Product Family Master Plan (Harlou 2006).

In this way, new concepts and manageable objects come into existence, only because they arise out of aggregated abbreviations.

Lastly, the displacement and use of abbreviations serve as means of providing the ability of remote control, which in this aspect is most applicable to the interpretation of 'action at a distance'. Remote control enables actions to be planned independently of physical presence, thus perfectly suited for management meetings etc., with the purpose of managing from a distance. From large-scale development projects (of Scandinavian scale), 50-200 engineers can be involved at the same time. Without the constant provision of remote control, middle management and project leaders would not be able to manage progress, interfaces between professions etc.

Latour (1986) elaborates the concept of representation through the explanation of visualization's effect on cognition. Making modeling visually explicit is a powerful way of breaking down barriers between professionals from different domains. Latour (1986) explains this in terms of the concept *immutable mobiles* that are capable of creating new hybrids, with information coming from different sources – for example from market and production. This is possible due to the optical consistency allowing professionals with different backgrounds, different positions and different interests to lay their daily working habits aside and see the world as the 'same type'. Therefore, the visualization of central aspects of architectures allows the gathering of professionals to treat the task in the 'same way' – due to the model being the same.

One could argue that the visualization acts as a boundary object by facilitating collective alignment of *inscriptions*. Without the visualization, individuals can have difficulties communicating about abstract concepts across professional domains. However, the collective alignment does not happen by merely looking at the visualizations. It takes training of participants' frame of understanding, alignment of terminology and constant use of exemplifications.

A central challenge in obtaining this rather ideal scenario is to provide the visual models with powerful graphics that enable the visual models to achieve this goal.

In order to achieve the unifying and 'enlightening' result of visual architecture modeling, a large amount of information usually has to be abbreviated and condensed into such abstract form that it enables a constructive dialogue without professionals feeling themselves drowned in an information overflow. To bring an example, it is not a challenge to print out 500 bills of material of products from a product program, but it is a challenge to present them in such a visual form that product and production professionals can have a discussion of the rationalization potential. In order to do so, a number of means exist, such as vast usage of intuitive symbolism, color markings ascribed with meanings, simple graphical layouts, appealing 2D/3D product visualizations etc. (Latour 1986) explains this as being able to enhance the contrast and turn incredible statements into credible ones.

As the modeling is often represented in large A0 format posters (or on large screens or projectors) they constitute a flat 2D medium capable of fixating certain phenomena. Latour (1986) describes this as

"nothing you can dominate as easily as a flat surface of a few square meters" (Latour 1986)

In other words, intervention and malleability is made feasible through the visual modeling. An important element of modeling is here, that modeling is always done with a purpose (Andreasen 1994). In other words, the person doing the modeling will always include (deliberate or not) inscriptions having the mere task of carrying a message across. In this research, the purpose of the modeling is to create an optimal fit between the decisions made in the market, product and production domains, in order to increase time-to-market and launch preparedness etc.

Impact for this research

According to the Impact Model A in Figure 2 the explicitness and visibility of critical aspects of architectures is the foundation for an improvement of the decision basis leading to improved decision making. The contributions by Cooper (1992) and Latour (1986) serve as important input to explain the nature of visualization as a form of representation. These are considered important *mechanisms* for identifying architectures.

3.4 Theory of architectures

The purpose of this chapter is not to provide a review of literature, as this is reserved for the appended papers. The purpose here is to provide a quick guidance to some of the fundamental research paths within research in architectures for product programs, in order to clarify for the reader how the author intends to position the research presented in this thesis.

As the ARC-diagram in Figure 4 from section 2.3.1 shows, the area of architecture for product programs and complexity cost reduction is related to a number of adjacent fields of research, many of which also use the term ‘architecture’.

Several areas of research are closely related to the topics of architectures for product programs. From the product side, integrated product development (Andreasen and Hein 1987), DFX methodology, methods for development of modular product architectures (Du et al. 2001, Gonzalez-Zugasti et al. 2000, Ericsson and Erixon 1999), and concurrent engineering (Fine 1998) meet the research areas of operations management, i.e. mass customization, personalization etc. for example Salvador et al. (2009). Unlike the broad definitions from recent systems engineering literature focusing on the architecture of the enterprise as a whole (Rebovich and White 2011), the concepts of program architectures adapted in this research, are closely related to the engineering disciplines of product and production development. Section 3.4.1 and 3.4.2 will elaborate on the elements included in these definitions.

3.4.1 Definitions of product architecture

Many different definitions of a product architecture exist in the literature:

- Ulrich (1995) has the comprehension that a product architecture is the scheme by which the functions of the product is mapped towards the physical parts, thus defining the product architecture as the arrangement of functional elements, the mapping from functional elements to physical parts and the specification of interfaces among these.
- Meyer and Lehnerd (1997) describe the architecture as being the combination of subsystems and interfaces. They argue that every product has an architecture, and that the goal is to make that architecture common across many variants.
- Sanchez et al. (2000) argues that a product architecture is created when a new product design has been broken down into its functional components and interface descriptions have been fully specified. The types of interfaces range from attachment, transfer, control/communication, and spatial, to environmental interfaces.
- Harlou (2006) describes an architecture as a structural description of a product program, product family or a product. It consists of design units, standard designs and interfaces, where design units are characterized by being unique to each product, and standard designs characterized by being reused between one or several product families. In this definition a clear emphasis is put on the decision of reuse, adequate documentation and organizational ownership, but also differentiating from Meyer and Lehnerd’s definition in the differentiation between product, product family, and assortment (program) architectures as hierarchically linked in three levels, where e.g. the product architecture is seen as a subset of the family architecture while requiring the description of standard designs and design units to qualify for the denomination *architecture*.

All of the mentioned definitions reflect important dimensions of the constitutive nature of a product architecture. None of which is directly contradictory, however, even more important is the behavioral aspects of a product architecture – in other words, what the product architecture can do for the company. According to Theory of Dispositions, behavioral aspects are encountered in the *meeting* between the product and the life phases (Andreasen and Olesen 1990).

3.4.2 Definitions of production architecture

This section will elaborate on the contents of a *production architecture*, acknowledging that production is the most significant life phase regarding e.g. costs and other performance parameters, while being directly related to the product architecture.

The comprehension of a production architecture, as described in this research and interpreted by the author, is focusing on the following aspects:

Production task

As Skinner (1974) argued decades ago, a blind-spot for most production managers is the attempt to solve the production tasks of different product families with the same manufacturing setup. This could be a mix of low and high series products, 'low' and high quality products or short or long lead time products, which results in the manufacturing setup having to compete with an impossible mix of demands. Furthermore, the manufacturing task changes over time as markets and product and production technology changes and evolves. When it comes to flexibility, the more flexible, the more expensive, time-consuming and non-standardized processes will be. Therefore, it is often not cost efficient to have the *best of both worlds* in the same manufacturing setup.

Point of variant creation

Different terms exist for denominating the point in time of the production flow where variants are created and/or customer orders are 'pulling' the production. Ramdas (2003) defines the *Point of Variegation* (PoV) as the point where the physical parts become dedicated to a product variant. The placement of this has an effect on all inventory levels and is a key decider for the possibilities of exploiting economies of scale throughout the manufacturing system.

Point of pull-push production

The *customer order decoupling point* (CODP) or *order penetration point* denotes the point in the production flow, where the customer order enters and triggers the production of a specific variant. From the field of mass customization it is often mentioned as a favorable situation, if possible, to postpone the CODP to a late stage (Michelsen and Pagh 2002). This form of control gives the opportunity of managing the production of subcomponents as mass production with high efficiency, leveled output and low waste initiated by a stock level input, followed by an order-initiated production of finished goods optimized for responsiveness and agility. A company exhibiting a postponement strategy might choose to postpone the CODP until the later stages in production, with increased flexibility etc. as benefits (Pagh and Cooper 1998, Feitzinger and Lee 1997). Other companies will have to produce finished variants directly to stock, thus exhibiting a speculation strategy, which limits the responsiveness to changing demands. External factors such as the nature of demand, whether fluctuating or leveled, and the degree of customization needed has a major impact on which strategy is most appropriate.

Manufacturing units and layout

The production layout is an important dimension in the effort of designing a production flow that is compliant with the product architecture and vice versa. A poorly designed production flow might not utilize the benefits offered by product modularity. Traditional process layouts impede a lean flow of products through the factory, and build a large need for intermediate storages. This will often have a drastic increase on lead-time, while efficient machine utilization is hindered. This type of layout also complicates the division of the production task from e.g. different product families with different requirements.

Process technology

As often described in Design for Assembly (DFA) and Design for Manufacturing (DFM) research (e.g. by Miles (1989) and by Boothroyd and Dewhurst (1983)) products can be deliberately designed to accommodate 'smart' manufacturing and assembly processes. However, DFA and DFM methodologies focus mainly on single-product issues, meaning that the evaluation criteria of the assessment of assembly and manufacturing processes of a product family differ from the ones from a single product. In multi-product development, there is a larger focus on process flexibility matching the product variance offered by the product family. For instance, when calculating through scenarios, it requires taking overhead costs systematically into account to evaluate equally with variable costs, as the industrialization of new product families might justify investing in new production equipment.

3.4.3 Architectures for other life-phase systems

According to Andreasen et al. (1996), structures can be defined for every life cycle phase, which is to be taken into account during development. And from Andreasen et al. (2004) a general definition of an architecture is

> "An architecture is a purposefully aligned structure of a system" (Andreasen et al. 2004)

Hence, the deliberate alignment of the structures of the life-phase systems may be denominated as architectures. However, the architectures vary a lot depending on what life-phase system is under consideration. This can range from architectures mainly constituted of structural elements (e.g. production) to architectures mainly constituted of behavioral character (e.g. service).

Constitutive and behavioral aspects

Based on earlier works (Mortensen 2000, Hubka and Eder 1988, Andreasen 1980), Andreasen (2011) proposed a consolidated suggestion for how to classify attributes of systems.

According to this suggestion, Figure 16 shows how attributes can either be constitutive or structural characteristics describing what the system *is*, or behavioral properties describing what the system *does*.

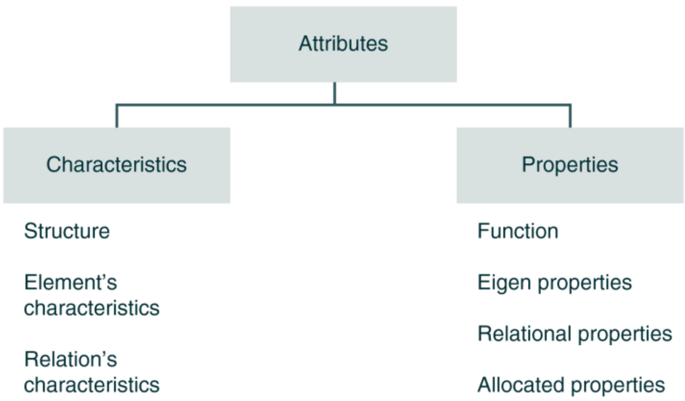


Figure 16 – Classes of structural characteristics and behavioral properties (Andreasen 2011)

Previous proposals for this basic classification of attributes included the differentiation between properties and *qualities*, being the properties that certain users allocated to products – e.g. pride of ownership. The distinction in Figure 16 includes these to be of different classes but all of the same type of attribute – namely a property.

Relational properties are related to the meeting between the products and the life-phase systems – an example is cost or time. One can argue that characteristics and properties are used interchangeably throughout many areas of engineering literature – but the terminology in itself is insignificant as such. The conscious distinguishing between behavioral and constitutive aspects is not.

Likewise, according to the quote from the introduction of section 3.4.3, it can be stated that

> An architecture of a life-phase system carries both constitutive or structural characteristics as well as behavioral properties

This statement has central importance to this research, as it underlines the necessity of including the behavioral properties when identifying, describing, and evaluating architectures for product programs.

4 Results

The purpose of Part 4 is to present the research results. According to the research questions, the results of the research work are divided into two main groups:

- *Identification of a program architecture*
- *Complexity reduction based on the quantification of complexity costs*

4.1 Publications within this research

In order to maximize the knowledge exchange during the research period, a number of publications have been created. The publication strategy included papers for conferences within the engineering design society and publications in relevant journals, all indexed by Thomson Reuters' Web of Knowledge – formerly known as the ISI index.

The publications appended this thesis include:

A) *Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P. &* ***Hansen, C.L.*** *2010, "Making Product Customization Profitable", International Journal of Industrial Engineering (Online), vol. 17, no. 1, pp. 25-35.*

B) *Mortensen, N.H.,* ***Hansen, C.L.,*** *Hvam, L., Andreasen, M.M., (2011). Proactive modeling of product and production architectures. Proceedings of the 18th International Conference on Engineering Design: Impacting Society through Engineering Design, 133-144*

C) ***Hansen, C.L.,*** *Mortensen, N.H., Hvam, L., (2012). On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market. 12th International Design Conference - Design 2012*

D) ***Hansen, C.L.*** *& Mortensen, N.H. (2014), Proactive identification of scalable program architectures: How to achieve a quantum-leap in time-to-market. International Journal of Industrial Engineering. 2014, Vol. 21(2)*

E) ***Hansen, C.L.,*** *Bruun, H.P.L., Mortensen, N.H. & Hvam, L. (2013), "Identification of a scalable architecture for customization of complex parts"*

F) ***Hansen, C.L.,*** *Mortensen, N.H., Hvam, L., (2012). Towards a Classification of Architecture Initiatives: Outlining the External Factors. Proceedings of NordDesign Conference 2012*

G) ***Hansen, C.L.,*** *Mortensen, N.H., Hvam, L., (2012). Calculation of Complexity Costs: An Approach for Rationalizing a Product Program. Proceedings of NordDesign Conference 2012*

H) *Hvam, L.,* ***Hansen, C.L.,*** *Mortensen, N.H. & Forza, C. (2019), The reduction of product and process complexity based on the quantification of product complexity costs. International Journal of Production Research. Published March 4, 2019*

I) *Mortensen, N.H., **Hansen, C.L.**, Løkkegaard, M., Hvam, L. (2016), Assessing the cost savings potential of shared product architectures. Concurrent Engineering: Research and Applications 2016 Vol. 24(2)*

Please see Figure 8 and Table 2 for an overview of connection between the publications and the research stages and activities.

Other publications within this research include:

J) ***Hansen, C.L.** 2012, Virksomheder forsømmer produkterne! (in Danish), Web article edn, Produktion 360 - tendenser, meninger, metoder, Copenhagen.*

K) *Mortensen, N.H., Gamillscheg, B., Bruun, H.P.L., **Hansen, C.L.,** Cleemann, K.K. & Junkov, K.H. 2012, Radikal Forenkling via Design (in Danish), DTU Mechanical Engineering.*

The publications are created in cooperation with co-authors of the research group in order to ensure sufficient professional discussion and challenging viewpoints.

4.2 Identification of a program architecture

The first research area is centered on the identification of a program architecture – an architecture for an entire product program.

The relevant research questions here are A1, A2, and A3 supplemented by hypotheses A1, A2, and A3, as presented in section 2.2.1.

4.2.1 Paper A

Title:	*Making Product Customization Profitable*
Journal:	International Journal of Industrial Engineering (2010)
Contribution:	Paper is written by supervisor based on case work and figures created by the author and colleague.
Case studies:	#1

Research question

The research question specific for paper A is A2: How to identify the most critical program decisions of an existing product program and make the central aspects of these explicit?

Research contribution

Paper A was the result of the first descriptive stage (DS-I), based on case study #1. Paper A contributes partially to the answering of research question A2, and in the sense that the proposed framework was not tested across many different companies and industries with different contextual settings. However, the results obtained in case study #1 were significant, making a thorough foundation for the further development of the framework.

Research method and result

The framework was derived through literature studies during the initial research clarification phase tested in a case study in a company producing mechanical systems for the energy industry as an OEM supplier.

The framework is shown in Figure 17 representing the three major areas; market, product and production/supply, which is supported by organization and work process aspects. One of the unique features of the framework is that the aspects are linked – here shown with red arrows connecting critical aspects across market, product and production/supply.

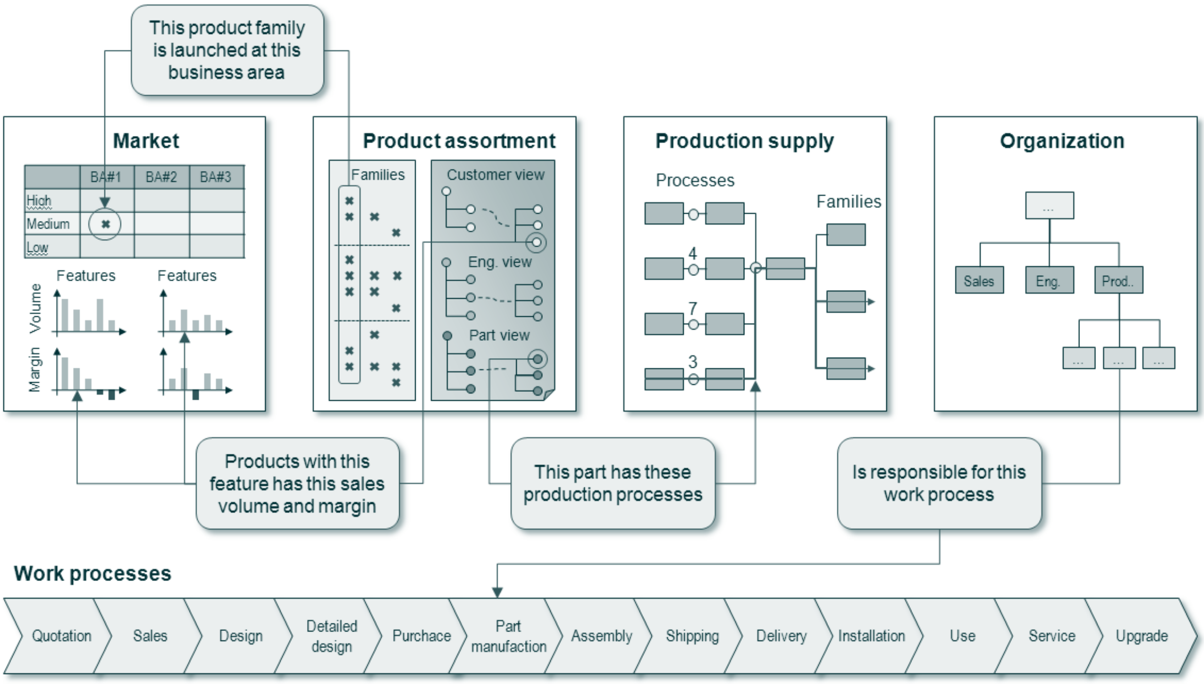


Figure 17 – Framework for Product Family Master Plan (Mortensen et al. 2010)

The framework proved to be a means of identifying an architecture for the OEM company in order to transform the engineer-to-order (ETO) fulfillment flow into a configure-to-order (CTO) fulfillment of customer orders. The aligned product and production architectures were the key to bring about this fundamental transformation.

Reflection on results

Subsequently, the achieved results has subject to the following reflection:

- The framework seems especially applicable for more traditional mechanical products where over-engineering of performance by +3-5% can be tolerated if significant benefits can be harvested in the transformation from an ETO to a CTO order fulfillment.
- The framework follows a top-down approach to scaling of performance and achievement of balanced performance points across the relevant performance range, as well as a top-down approach to the definition of mechanical and fluidal interfaces.
- The framework was tested in a project-based case company thus being applied for the definition of an open *but limited* solution space – in contrast to a definite solution space of a product-based company. Applying the framework to other situations than the project-based one demonstrated here cannot be done without sufficient caution and adaptation.

The framework is linked to paper D that proposes a framework for the proactive identification of an architecture for product-based companies.

4.2.2 Paper B

Title:	*Proactive Modeling of Product and Production Architectures*
Conference:	ICED'11 – 18th International Conference on Engineering Design (2011)
Contribution:	Paper is written in cooperation with supervisor with the main chapter (5) written by the author including all figures.
Case studies:	Based on case studies by previous members of the research group.

Research question

The research question specific for paper B is A1: How to make the most critical aspects of product program design explicit during the early development phases of a new product program?

Research contribution

Paper B was the result of the research clarification phase and first descriptive stage (DS-I) evaluating how the current state-of-the-art of the research group can support the early development phases of a product program. The proposal is an operational model as a collection of modeling techniques focusing on the constitutive aspects of architectures with the inclusion of the roadmap dimension to capture parts of the behavioral aspects of this – namely which derivate launches to be derived from the architecture.

Research method and result

The focus was on the coordinated development of product and production architectures, even though parts of the market aspect were included too. The operational model was developed through literature reviews and through the research group's experience within this theoretical field, and was tested in a case study in a global company producing professional equipment for industrial applications.

The operational model proposed proved to enable

- Earlier and more explicit definition of high-end/mid/low-end products including the clarification of features and options layout.
- Conscious decisions on interfaces to enable truly interchangeable modules that are prepared for the next three product launches.
- Improved synchronization between product program development and production development including specification of completely new production lines with three different level of automation: Fully automatic, semi-automatic and manual production and assembly.
- On time launch of the first generation of the product program (the later generations was introduced after the case study ended).

Reflection on results

- The paper was published rather early during the research period in order to receive feedback from the scientific community in due time to prepare the following prescriptive studies (PS-I and PS-II). Therefore, the contribution is focusing on the collection and conceptualization of complementary modeling techniques into an operational model that can be applied by companies from the early phases of product-based development of product programs (in opposition to project-based development).
- The paper uses the concept 'market architecture' even though this was not an established or defined concept at the time of writing. Paper C was prepared in order to address this issue being the market aspect of product program design.
- The paper should be seen as preliminary work leading up to the framework presented in paper D.

4.2.3 Paper C

Title: *On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market*

Conference: DESIGN 2012 – 12th International Design Conference (2012)

Contribution: First author

Case studies: #3 #4 #5 #6 #7

Research question

The research question specific for paper C is both A1 and A2 in the sense that a conceptualization of an architecture for the market aspects of product program design becomes an important part of the answering of A1 and A2. Thus paper C addresses the market side of A1 and A2.

Research contribution

The contribution from paper C is the conceptualization of the concept of an architecture for the market. The 'market' is to be interpreted as the market side of product program design. Alternative titles for the concept could be 'sales architecture' or 'offerings architecture'.

Research method and result

Figure 18 is an expansion of the model of the three coherent aspects of the market, product and production/supply (Mortensen et al. 2008), following the classic partitioning from Integrated Product Development (Andreasen and Hein 1987).

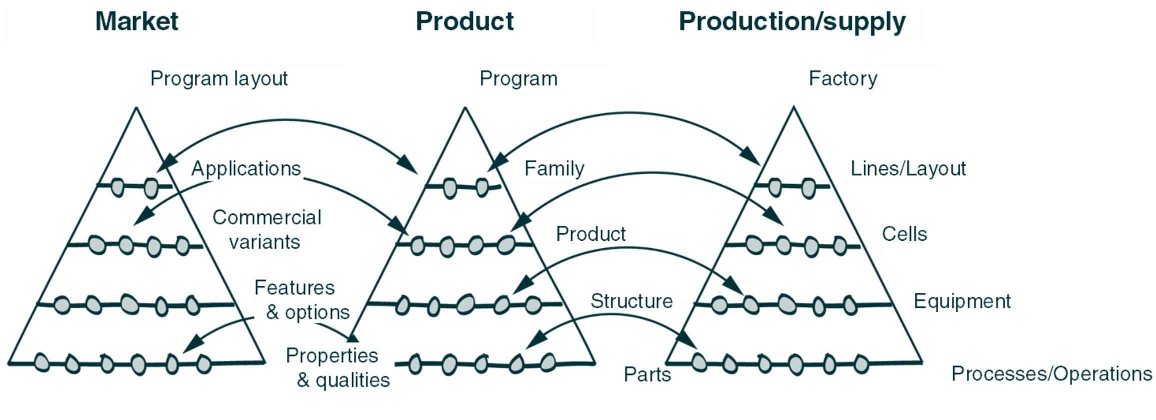


Figure 18 – Three architectures: Market, product and production/supply (Hansen et al. 2012)

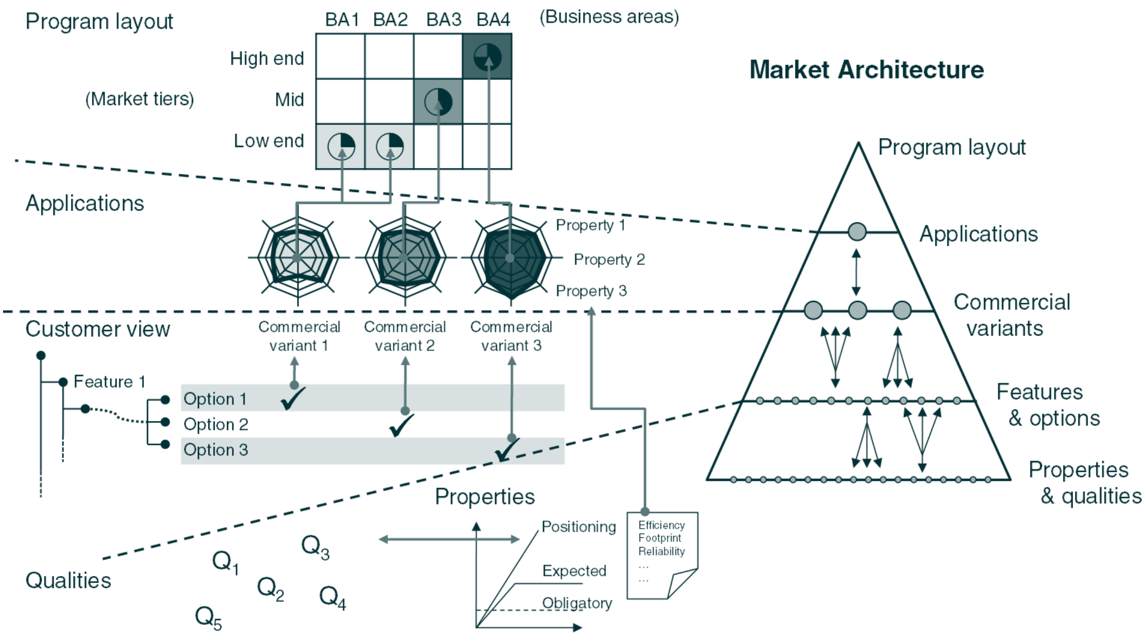


Figure 19 – The market aspect of product program design (Hansen et al. 2012)

The paper presents seven classic pitfalls of product program design, which the definition of an architecture for the market can support the avoidance of. Furthermore, five groups of requirements for the market architecture are formulated and Figure 19 presents a conceptual model of the result.

Reflection on results

Feedback from the scientific community was centered on two aspects:

How is it possible to design an architecture of the market, when the market is not a physical object of design?

- As the 'market' term represents the sales phase of the product life cycle (which partly can be said to represent the use phase), it is possible to characterize the life-phase as a system (see section 3.4.3). This *system* is hierarchical and has close ties with the product architecture in the sense that it is supposed to guide and control the development of the product program. The constitutive aspects of the market architectures has to do with the selection of features (included/excluded) answering the question of what is the architecture covering or not covering and by which features. The behavioral aspects of the architectures have to do with the performance and competitiveness of the product program in the target market segments – namely answering the question of how well the architecture covers the target segments.

The naming 'market architecture' can be mistaken for a characterization of the market place as a separate system. This is not the intention. Therefore, the author suggests the following synonyms used for explanatory purposes only:

- *Sales* architecture
- *Application/feature* architecture
- *Offerings* architecture

4.2.4 Paper D

Title: *Proactive Identification of Scalable Program Architectures: How to Achieve a Quantum-leap in Time-To-Market*

Journal: International Journal of Industrial Engineering. 2014, Vol. 21(2)

Contribution: First author

Case studies: #7, # 16, #17, #18, #19, #22, #23

Research question

The research question specific for paper D is A1: How to make the most critical aspects of product program design explicit during the early development phases of a new product program?

Research contribution

Paper D presents the Architecture Framework for Product Family Master Plan, which is a further development from the model presented in paper B. The most significant additions are the more distinct differentiation between constitutive and behavioral aspects while introducing a number of modeling techniques including ones specifically suited for evaluation purposes. This is to answer the question of how well the program architecture is fulfilling its purpose.

Research method and result

The architecture framework consists of a set of modeling techniques and is specifically tailored and tested in product-based companies in order to proactively identify a scalable program architecture for the entire product program.

The term 'program architecture' is reflecting the result of aligning the market, product and production architectures and coordinating these with the multi-level roadmap in order to ensure an improved preparedness towards future launches. Thus, the term differentiates itself from a product architecture in the sense that a program architecture requires the alignment with market and production architectures to achieve on one side competitiveness and the other side attractive cost levels – from first launch and over time.

The term 'scalable' refers to the scalability of solutions by development of modules that are designed to be scalable in the desired range and in relation to key design characteristics. This enables the achievement of attractive cost of system performance optimized for price, value or performance.

The modeling techniques for evaluation support the identification of program architecture, where:

- Module consequences of feature inclusion are known – e.g. by evaluating coupled vs. decoupled implementation
- Cost of preparation towards future launches is carefully balanced against the probability of future use in order to evaluate the optimal level of design preparation
- Production consequences of future module variants are known and balanced towards the effort of preparing the production setup for the new modules

The framework has proved to enable a significant reduction in time-to-market for products derived from the program architecture during case studies in four different companies. The framework is presented in its entirety in a conceptual representation in Figure 20. A larger version is included in chapter 6.4.

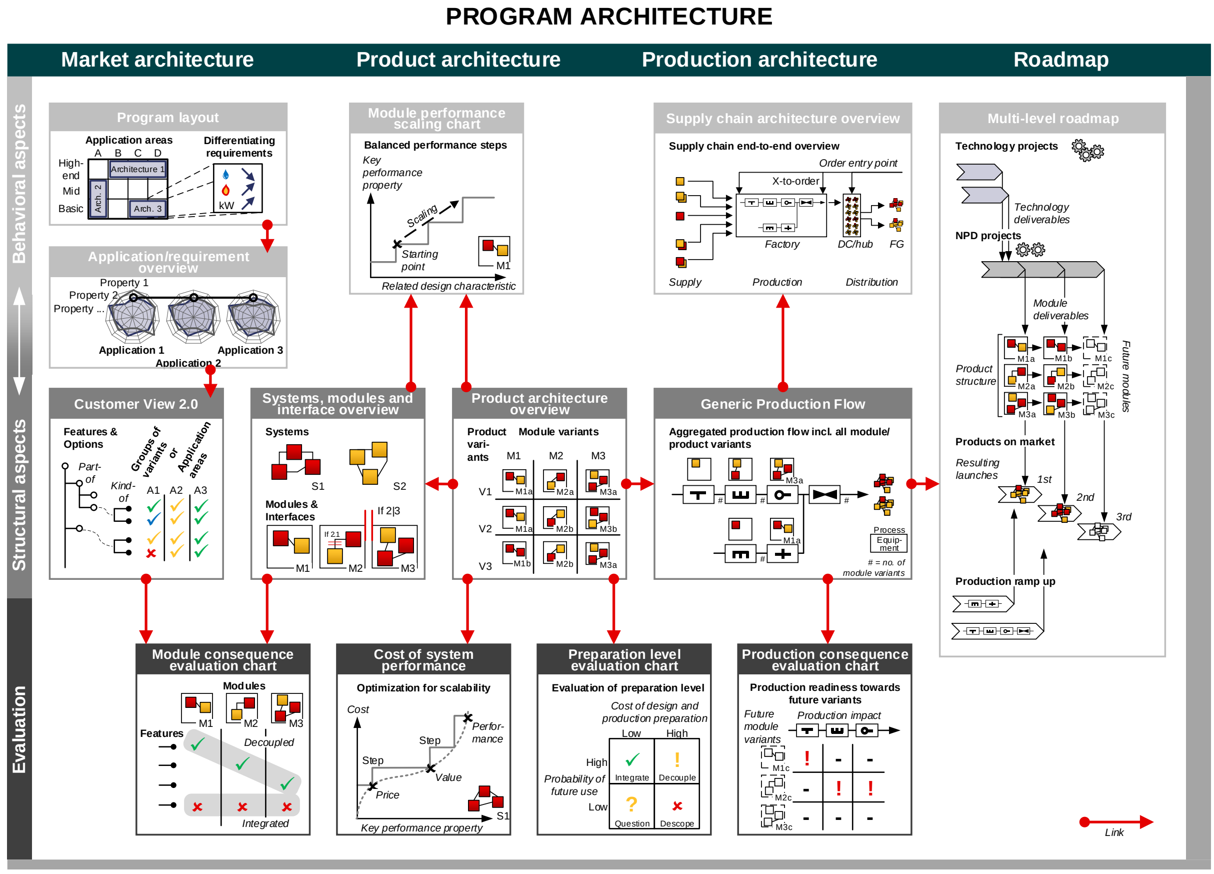


Figure 20 – Architecture Framework for Product Family Master Plan used for the identification of a program architecture (Hansen and Mortensen 2013)

Reflection on results

- The term system performance relates to TTS in the way that 'systems' here are analogous to the 'function-carriers' or ToD's physical implementations of 'organs'. Therefore, performance should be derived from the performance of systems through performance related properties, and the link to the design characteristics is established through the modules. Where a 1:1 relationship exists between the performance property, the module, and the design characteristics delivering variation within each module, the design can be nominated as 'decoupled', as the organ is fully encapsulated.

- Where product architectures cannot be truly modular for various practical reasons, the distinction between systems and modules is a reasonable way to describe the function structure on one hand and the physical design or production structure on the other hand. The governing structuring principle for modules is often related directly to design encapsulation or production requirements of producibility.

- As the four case companies experienced different challenges during the development phases, not all 13 modeling techniques from the architecture framework was applied in each case. As it is detailed in paper D, a selection was tailored to meet the specific needs of the cases.

- It is difficult to create a truly comparable measure of the results, given the amount of varying external conditions and different levels of interaction with the development teams in the companies. Results from the application of the framework varies between being able to develop 50% more product variants using comparable resources (fourth company), and reducing development lead time of the early phases by –50% (first, third and fourth company). In the second case company, the effect on production preparation level was the optimization for quick responsiveness (for inclusion of new modules) bringing the loss of productivity to a minimum, compared with the previous situation, where rebuilding and expansion of production equipment used to lead to an unacceptable downtime of production. This was the key enabler for reducing time-to-market for derivative projects included in the program architecture. Thus, different means have played a role in achieving the time-to-market reduction.

4.2.5 Paper E

Title: *Identification of a Scalable Architecture for Customization of Complex Parts*

Contribution: First author

Case studies: #8, #9

Research question

The research question specific for paper E is A2: How to identify the most critical program decisions of an existing product program and make the central aspects of these explicit?

Research contribution

The contribution presented in paper E is a framework including a step-wise approach for the identification of an architecture for the product and production setup. The framework is tailored to support project-based companies doing customization by engineering of products with highly integrated product structures of complex parts. These are often serving their customers in an OEM setup.

The contribution differentiates itself in its ability to identify and define an architecture for the product and production setup for project-based companies that serve their customers by developing and producing products on a project-basis. The framework focuses on those companies that cannot apply traditional modularization, as the functionality provided through their highly integrated product designs cannot become encapsulated in traditional physical modules separated by interfaces. Instead, paper E suggests a framework and a step-wise approach to identify a scalable architecture for customization of complex parts in order to reap the benefits of modular architectures, without 'assistance' from traditional modularization.

In addition to paper A, paper E partly answers the research question A2 in the sense that it enables the identification of critical program decisions of an existing product program during its step-wise approach in nine steps, using lead variant designs as the *focal point* for focusing the architecture definition.

The framework can be used on existing product programs in the sense that it does not require new development of sub-solutions etc., but a new scaling principle to achieve adequate variation within market parameters closely coordinated with the scaling principle used for the production equipment.

Research method and result

The framework has been developed based on literature and experience. As Table 1 (page 35) and Table 5 (page 75) show a number of case studies have been carried out in project-based companies, ensuring that the framework has been developed on a thorough basis of experience too. Previously, a number of earlier case studies have been carried out within the research group in previous research projects, providing a solid foundation for suggesting the framework presented in Figure 21.

Following the classic partitioning from Integrated Product Development (Andreasen and Hein 1987), the framework is also in alignment with the hierarchical model presented in Figure 18.

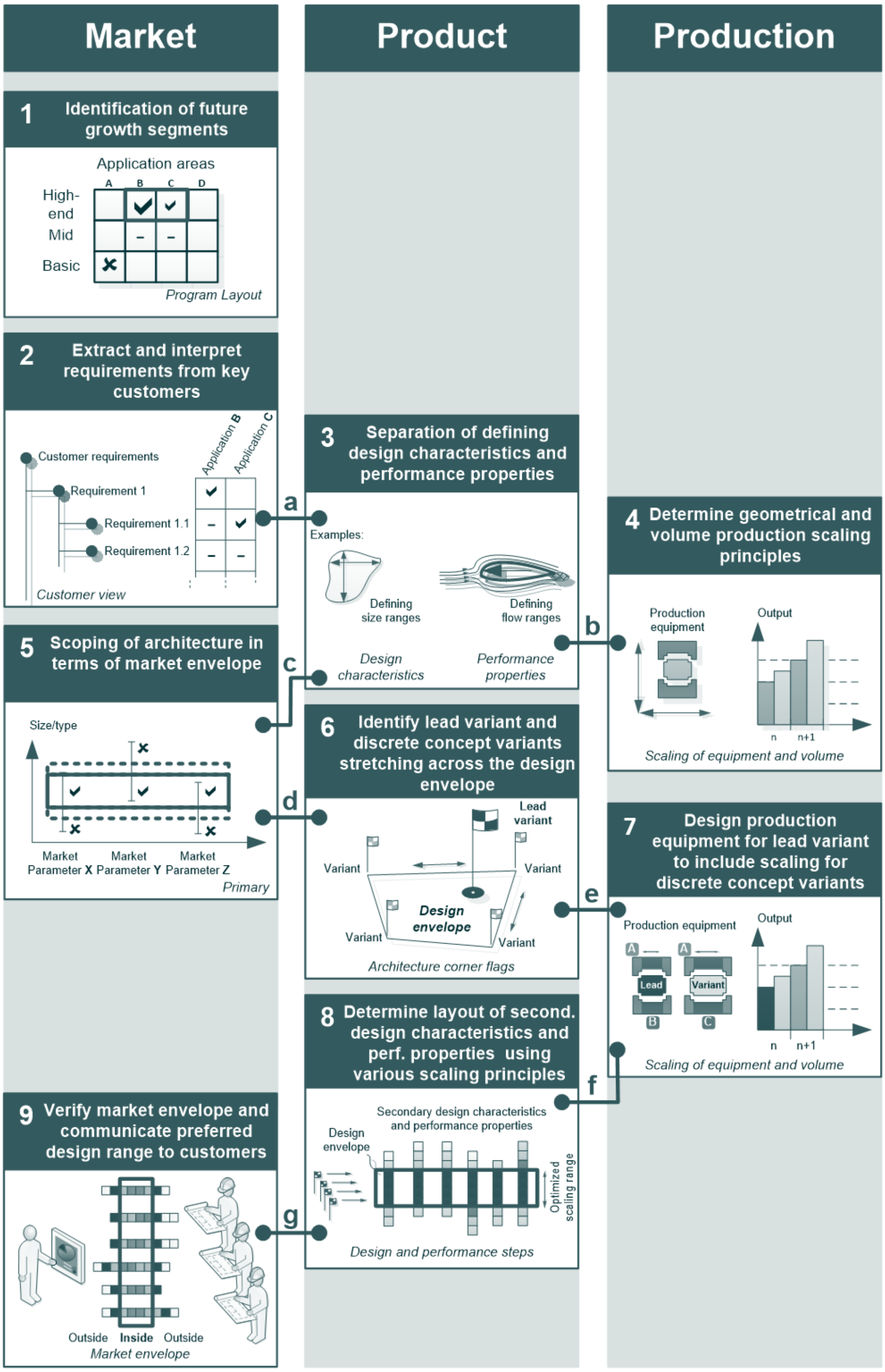


Figure 21 – Framework including step-wise approach

Each step contains a modeling formalism as well as a link to other steps – indicated by red lines. The linking is an important feature of the framework, as it is impossible to achieve attractive cost- and price points without the coordinated development of product and production scaling principles in alignment with the market envelope and requirements from key customers.

Please see the appended paper E in Part 6 for a full-scale version of Figure 21.

Reflection on results

- In opposition to the framework presented in paper A, this framework is especially suitable for high performance products where no +3-5% over-engineering can be tolerated.
- Whereas the framework in paper A suggests a top-down approach, the framework presented in paper E could be characterized as a bottom-up approach in related to the domain of product design, as no compromises on performance is allowed for the targeted high performance products.
- One of the framework's strengths is the focus on fulfilling the primary performance properties by explicitly linking them to the defining design characteristics, from which the scalability of performance must be controlled to minimize impact on production equipment and quality level.

4.2.6 Paper F

Title:	*Towards a Classification of Architecture Initiatives: Outlining the External Factors*
Conference	International Conference of NordDesign 2012
Contribution:	First author
Case studies:	-

Research question

The research question specific to paper F is A3: How to take the contextual differences of a company into account when scoping and comparing architecture initiatives?

Research contribution

The contribution presented in paper F is the introduction of a set of external factors capturing the contextual differences that set the stage for architecture initiatives. The external factors are divided into classes of market factors and product/production factors.

During the progression of the research project, the author has experienced that the importance of including situational and contextual awareness is highly critical when considering the success of architecture initiatives. Research question A3, which can be considered supplemental to A1 and A2, is formulated in order to address this situational and contextual awareness.

The contribution has relevance in the sense that it addresses the improvement of:

- Scoping and goal setting of architecture initiatives
- Comparability of initiatives
- Transferability of experiences

These arguments of motivation are based on the notion that architecture initiatives are experienced as being difficult to scope, difficult to compare, and in many cases difficult to communicate between academic societies and industrial practitioners. From the perspective of the author, the lack of understanding of the contextual differences that set the stage for the architecture initiatives provides an element of explanation for these difficulties.

Research method and result

Even though the result presented in paper F does not represent a completed research result, this intermediate result does match the hypothesis A3, in the sense that it is possible to classify program architecture initiatives on the basis of differentiating external factors to allow for the inclusion of contextual criteria when defining, scoping and comparing architecture initiatives.

Table 5 presents a mapping of architecture case studies versus the classification criteria presented in paper F.

Classification criteria	*Case studies*	#1	#2	#3	#4	#5	#6	#7	#8	#9	#16	#17	#18	#19	#22	#23
Market factors								-								
Market launch clock speed	*Stabile architecture*	X	X			X	X		X	X		X	X	X		
	Evolving architecture			X	X						X				X	X
Supply chain position	*Direct to end-customer*				X	X										
	Sales subs.			X							X	X			X	X
	Contract./prof./OEM	X	X				X		X	X						
	Public proc.												X	X		
Market positioning	*Customer have bargaining power*	X		X	X				X	X						
	Company has bargaining power		X			X	X				X	X	X	X	X	X
Product/production factors								-								
Primary driver for product positioning	*Sheer performance*					X	X		X	X						
	Feature multiplicity	X	X	X	X						X	X	X	X	X	X
Product customization	*Definite solution space (e.g. product-based)*			X	X	X	X				X	X	X	X	X	X
	Open solution space (e.g. project-based)	X	X						X	X						
Product and production technology clock speed	*Slow*	X	X				X		X	X	X	X	X	X		
	Fast/Rapid			X	X	X									X	X
Volume per variant (no. of nominally identical items)	*One-off (e.g. ETO)*															
	Low volume (e.g. CTO)	X	X			X	X		X	X						
	High volume (mass customization)			X	X						X	X	X	X	X	X
Macro-economic environment (and/or)	*Currency and material prices*	X				X			X	X					X	X
	Production/sourcing location	X	X			X	X		X	X						

Table 5 – Case studies mapped towards the classification criteria

Reflection on results

- Valuable input was collected while presenting the external factors at the conference. One of them was the proposition for dividing parameters into the ones than can be partly affected, such as supply chain position, and the ones that cannot be affected at all. This is a natural step for further works.
- The inclusion of additional classification factors from TTS – although overlapping in many cases – would be able to provide perspective and expand the external factors to a more complete level. Examples from TTS could be the classification of technical systems by type of production, design originality/degree of novelty, difficulty of designing, production location and degree of standardization (Hubka and Eder 1988).
- Instead of proposing the division between market and product/production factors, synonymous terms could be commercial and technical factors instead.
- The inclusion of the OEM customer in the contractors/professional customer category might seem a 'stretch' in the sense that these two groups can represent quite different situations of sales and specification, as OEM supplier-customer relationships are characterized by the customer having the major part of the buying power.

4.3 Complexity reduction based on quantification of complexity costs

The second research area is centered on complexity reduction based on the quantification of complexity costs.

The relevant research questions here are B1 and B2 supplemented by hypotheses B1 and B2 as presented in section 2.2.1.

4.3.1 Paper G

Title: *Calculation of Complexity Costs – An Approach for Rationalizing a Product Program*

Conference: International Conference of NordDesign (2012)

Contribution: First author

Case studies: #10, #11, #12

Research questions

The research questions specific for paper G is B1 and B2:

- B1: How to identify and quantify complexity costs of a product program?
- B2: How to use quantification of complexity costs to support decision making in product program design?

Paper G has a focus on B2.

Research contribution

The contribution from paper G is a five step approach for rationalizing a product program.

The approach includes the analysis, quantification and allocation of product program complexity costs by means of identifying a number of Life Cycle Complexity Factors capturing the product-process complexity of interest. The complexity factors are defined as being relative from the optimization viewpoint presented in section 3.2.4:

> The life cycle complexity factors represent the life-phase meetings where the costs of having *too many* parts, solutions, product variants or business processes appear

'Business processes' include the supply, production, and delivery processes in the manufacturing flow as well as specification and development processes in the order flow. The definition is an elaboration from Wilson and Perumal (2009). To the above definition it is added:

> The life cycle complexity factors are characterized by either having an uneven cost distribution across product variants (relative comparison) or by being unproportionally costlier than other comparable processes (absolute comparison)

Research method and result

The approach is developed through literature studies and experience, to contribute in filling out an important gap between research in architecture-based product development and research in complexity costs from operations and supply chain research. Even though significant contributions exist within both areas, very little research has been centered on the actual quantification of benefits to be achieved from the architecture-based approach to product program development. This contribution is an attempt to diminish this gap by proposing an approach that allows for a rationalization of a product program by quantification of complexity costs. The author sees this step as an important starting point for proving the benefit of rationalizing a product program (a precondition for B2), which is a common objective shared by this approach and the contributions presented in papers A to F.

Consequently, the approach presented here carries two important objectives

- *Reactive* rationalization of a product program – 'the cleaning out' approach
- *Proactive* avoidance of complexity costs by improved scoping of architecture initiatives (e.g. by Design-for-X focus and location of variant creation points) and improved program management of products (knowledge about an estimated true profitability)

The approach includes five steps:

Step 1:	*Scoping of analysis:* Selection of confined product program
Step 2:	*ABC analysis of product profitability:* Evaluation of gross profitability
Step 3:	*Life Cycle Complexity Factors:* Identification, analysis, quantification and allocation of complexity factors to achieve and complexity adjusted profitability of product variants
Step 4:	*Short-term fixing:* Calculation of scenarios for fixing the product program and initiation of short-term initiatives
Step 5:	*Complexity reduction program:* Definition of initiatives reducing complexity costs and identification, minimization and decoupling of Life Cycle Complexity Drivers

The identification of complexity cost factors can be made using a top-down approach analyzing the cost structure of the product program or by using a bottom-up approach formulating hypotheses of cost asymmetry of individual complexity factors. Cost asymmetry is characterized by products variants contributing unevenly to the indirect costs that are not already directly accounted for in calculated product costs.

The objective of proactive avoidance of complexity costs is closely related to addressing research question B2 by linking the complexity drivers to the recorded complexity factors. In this way, the fifth step of the approach allows for the specific addressing of the complexity drivers in order to either eliminate them or delimit their negative effects on the complexity factors. This relationship can be formulated as:

An action chain of causal relationships between causes and consequences
can be established between life cycle complexity drivers (causes)
and life cycle complexity factors (consequences)

Reflection on results

- As paper G was published in a limited conference format, the paper is only partly addressing the research question B1. Paper H brings a more thorough presentation of the motivation for quantification of complexity costs, the approach development, and experiences from field testing the approach in case studies.

- Paper G does not address the definition of complexity costs, as it is necessary to introduce a wide body of knowledge, which the conference format did not allow to. As from the discussion in section 3.2.4, the complexity costs of real interest here, is the excess system complexity that can be removed by rationalizing the product program based on initiatives formulated in the fourth and fifth step of the approach. However, while detecting, identifying and quantifying the complexity costs during the third step, it is not yet known which share of the complexity costs can actually be removed as excess complexity costs. In other words, the complexity cost reduction potential is contingent upon the defined initiatives aiming to reduce the complexity costs. This is an important concept for understanding the use of the term 'complexity cost' as is applied here.
- The approach was tested in case study #10 followed by supply chain related complexity analyses in case study #11 and #12. Case studies #13-15 were set up to further test the approach and gain further experience with the method.

4.3.2 Paper H

Title: *The reduction of product and process complexity based on the quantification of product complexity costs*

Journal: International Journal of Production Research (published March 4, 2019)

Contribution: Paper H is a further development of paper G (first author), and as the second author the contribution has been large parts of the method, literature review and figures.

Case studies: #13, #14, #15

Research questions

The research questions specific for paper H is B1 and B2:

- B1: How to identify and quantify complexity costs of a product program?
- B2: How to use quantification of complexity costs to support decision making in product program design?

Paper H has a focus on answering B1 more thoroughly than the short conference format of paper H allowed to.

Research contribution

The contribution presented in paper H is focusing on research question B1, as paper H is an extension of paper G. Thus paper H adds to paper G in terms of:

- Clarification of development and testing of the procedure
- Elaborate detailing of a case study where the procedure is field tested
- Elaborate outlining of possible initiatives for complexity reduction addressing both complexity factors directly (for example inventory of components and finished goods), but also the complexity drivers (for example lack of freight optimization in R&D, inappropriate solutions for country specific customization etc.)
- Outlining of insights and experience gained from testing the procedure
- Discussion of major difficulties encountered and recommended solutions
- Discussion on procedure's relevance given that it enables comprehensive insights into the complexity costs of a product-process system with limited and affordable resource consumption

Research method and result

As in paper G, the procedure presented in paper H is the result of literature studies and experience collected within the research group and the primary authors.

The approach includes five steps that are slightly altered from the ones presented in paper G:

Step 4: *Identification and quantification of possible initiatives for the reduction of complexity costs*

Step 5: *Evaluation and prioritization of initiatives and establishment of the complexity cost reduction program*

In this way, Step 4 focuses on the broad definition of initiatives including specific but preliminary quantification of benefit potential associated with the initiatives, and Step 5 includes the evaluation and prioritization of initiatives.

The procedure contributes to the literature on strategies for complexity reduction, and to the literature on process complexity by establishing a clear link through the complexity factors. The procedure contributes to the literature on product complexity in the sense that it suggests the A-B-C grouping of product categories and shows how the product complexity can be associated with the complexity factors, and that B- and C-products often have higher costs than A-products. This is one clear indication of complexity cost reduction potential.

Reflection of results

An analysis dimension which was de-scoped from papers G and H is the *complexity cost sensitivity*. The complexity cost sensitivity is defined as:

> The complexity cost sensitivity is the sensitivity of a complexity cost factor towards an increase or decrease of a complexity cost driver

This allows for the calculation of complexity cost reduction scenarios by defining scenarios for decreasing (or increasing) the complexity cost drivers. The concept is elaborated in Figure 22.

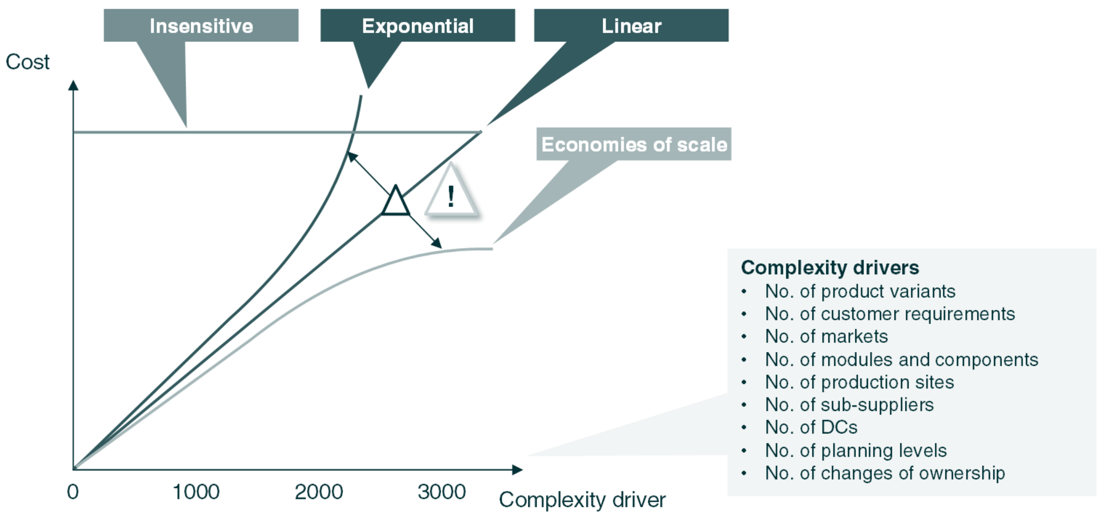


Figure 22 – Four types of basic complexity cost sensitivities

From Figure 22 four different types of basic sensitivities are defined:

- *Insensitive:* The cost is insensitive to the complexity driver.
  Example: The salary of the CEO does not depend on the number of product variants.
- *Exponential*: The cost is increasing exponentially when the complexity driver is increasing.
  Example: The cost of inventory handling increases when the utilization of a warehouse reaches the upper limit for the capacity – for example as a result of an increasing number of product variants
- *Linear:* The cost is increasing in a linear relationship to the complexity driver.
  Example: The sales cost increases linearly to the number of markets.
- *Economies of scale:* The cost increase is decreasing as the complexity driver is increasing.
  Example: The cost of changeover when adding more modules that fit the same production equipment.

Other types of cost sensitivities obviously exist, and most often the relation will be *step-wise* made up by a mix of these four basic sensitivities.

The cost sensitivity can become a major issue if a cost which is regarded as benefiting from economies of scale indicates to be exponentially sensitive. This can play a role in explaining why merging companies seldom reap the benefits of improved economies of scale from day one.

During several case studies, the complexity cost sensitivity in relation to the product variants has been examined. If the cost sensitivity is 20%, it means that by reducing the number of variants to a theoretical minimum of 0 (or 1), the cost index would only decrease by 20% to index 80. Similarly the cost index would decrease to index with a cost sensitivity of 40%.

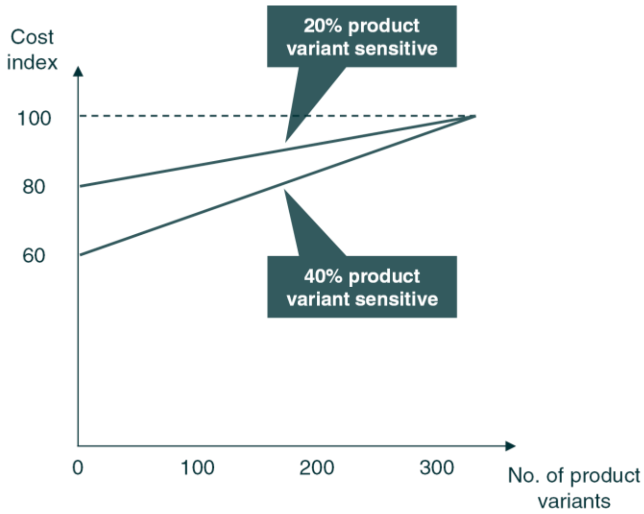


Figure 23 – Cost sensitivity towards no. of product variants

This concept makes it possible to calculate the delta between the current cost and a single variant scenario. The single variant scenario approach is also adopted by Lechner et al. (2011).

4.3.3 Paper I

Title: *Assessing the cost saving potential of shared product architectures*

Journal: Concurrent Engineering: Research and Applications (published March 4, 2016)

Contribution: As second author the primary contribution has been the case study work itself along with synthesizing the framework.

Case studies: Based on case studies beyond the original scope of the thesis.

Research questions

The research questions specific for paper I are:

- B1: How to identify and quantify complexity costs of a product program?
- B2: How to use quantification of complexity costs to support decision making in product program design?

Research contribution

The contribution presented in paper I is:

- An operational method for architecture mapping and evaluation

Research method and result

- Experiences from case studies show that a reduction of –60% of the number of architectures is often possible, leading to a cost reduction between 0,5%-2% of the turnover.

5 Conclusion

The purpose of Part 0 of the thesis is to conclude on the presented research. This will be done by summarizing the addressing of the research questions that serve as general research questions across the presented papers from Part 4.

5.1 Research findings – answering the research questions

As described in section 2.2.1 five research questions were formulated providing the objectives of the investigation this research represents. The following sections will elaborate on the specific answering of these.

5.1.1 Identification of a program architecture

For the identification of a program architecture, research questions A1 and A2 were central:

Research Question A1:

How to make the most critical aspects of product program design explicit during the early development phases of a new product program?

Research Question A2:

How to identify the most critical program decisions of an existing product program and make the central aspects of these explicit?

As presented in section 2.2.1, a number of working hypotheses were formulated to guide the answering of the research questions. Hypothesis A1 was the first of these.

Hypothesis A1:

The modeling of the program critical aspects across market, product and production domains will improve the decision basis for product program design, in order to improve time-to-market and/or R&D efficiency for product launches derived from the program architecture.

Hypothesis A1 is guiding the answering of both research questions A1 and A2, in the sense that it outlines the three critical domains of market, product and production as the key to achieve improvement in time-to-market and/or R&D efficiency. The contributions presented in papers A – B – C – D – E are all compliant with the hypothesis A1 in the sense that they – in their entirety – prove that the identification of a market, product and production architecture can be considered contributing elements for achieving the mentioned improvements. The papers show how the critical aspects across the three domains are captured in the architecture models, which leads to hypothesis A2:

Hypothesis A2:

It is possible to identify a program architecture for a product program including both structural/constitutive and functional/behavioral aspects through the application of explicit modeling techniques and through appropriate staging.

Hypothesis A2 is guiding the answering of both research questions A1 and A2 too, in the sense that it outlines how the description of an architecture for a product program – called a program architecture – should include both structural/constitutive and functional/behavioral aspects, in order to provide an improved quality of the decision basis to promote the alignment between the architectures of the market, product and production (see Impact Model in Figure 2). To sum up:

A program architecture is a model of an architecture for a product program resulting from the purposeful alignment of market, product and production architectures including both constitutive and behavioral aspects

The *purposeful alignment* includes the evaluation and optimization towards the program architecture that satisfies the critical goals of the company in the best possible way with the given resource constraints.

Two of the most important behavioral aspects are the preparation towards future launches and the relation between the performance of key properties and the cost of achieving these.

Appropriate staging

Hypothesis A2 further guides the working approach by describing how the application of explicit modeling techniques must be staged appropriately. The question of appropriate staging has been touched briefly in each paper where it is outlined how the working approach of the modeling techniques includes the use of large A0 format posters where the modeling techniques have been applied.

The term appropriate staging also includes the orchestration of intra-domain reviews of the architecture models as well as cross-domain reviews. The intra-domain reviews serve the purpose of detailing models within a specific domain with experts, whereas cross-domain reviews serve the purpose of aligning decisions across domains of market – product – production.

Intra-domain vs. cross-domain models

As described in chapter 3.3 the basis behind utilizing modeling as a management technology is outlining how powerful architecture models can be, if they are staged appropriately. Intra-domain models do not have to become boundary objects, as they primarily serve the purpose of clarification and conceptualization within a specific domain (ref. section 3.3.2). However, the cross-domain models benefit largely from becoming boundary objects, if this process can be controlled by the participants. During the case studies, the author has been the primary source of control of the architecture models, as these cannot be handed to an organization from one day to the other. In most cases, the models have been handed over to company professionals after the research case studies have ended, with dedicated resources being responsible.

> The application of explicit modeling techniques promotes the identification of a program architecture if models are properly prepared for constructive intervention and malleability through structured reviews by competent and influential domain representatives:
>
> - The models can be *intra-domain specific* for clarification of details within architectures.
> - The models can be *cross-domain hybrids* to promote alignment across architectures, if the resulting models are staged as a boundary object between representatives from market, product and production domains.

As a researcher with a strong focus on industrial practice, the question of whether a company already has an architecture in place or not, has often been asked. There is of course no simple way of answering this, and it does require analyses to provide a comprehensive answer. However, it can be argued that the probability of having an improvement potential on the program level is most often great and proportional to the lack of cross-functional alignment. Cross-functional alignment here is not considered only an activity on the strategic level, but also on the tactical and operational. Furthermore, as there is most often only very few formal descriptions of the current elements of architectures in place, the probability of an identifying an improvement potential, when analyzing a company from market, product and production sides, is considered extremely high.

As a supplement research question to A1 and A2, A3 was formulated:

Research question A3:

> How to take the contextual differences of a company into account when scoping and comparing architecture initiatives?

The basic idea starting out the answering of research question A3 was formulated in hypothesis A3:

Hypothesis A3:

> It is possible to classify program architecture initiatives on the basis of differentiating external factors to allow for the inclusion of contextual criteria when defining, scoping and comparing architecture initiatives.

Hypothesis A3 was formulated from the basic notion that a fundamental and natural step towards a maturing of the research into architecture must be to classify architecture initiatives. A first step on this challenge was the ascertainment of the external factors that pose a set of differentiating influence on the architecture initiative. Therefore hypothesis A3 does not give a detailed answer to research question A3, but an overall one in the way paper F explains how to be aware of the varying external factors, map and characterize them, and prioritize how to take them into account when scoping an architecture initiative.

5.1.2 Complexity reduction based on the calculation of complexity costs

For complexity reduction based on the calculation of complexity costs, research questions B1 and B2 were central:

Research question B1:

How to identify and quantify complexity costs of a product program?

Research question B2:

How to use quantification of complexity costs to support decision making in product program design?

In order to guide the research and answering of the research question, two working hypotheses were formulated. Hypothesis B1 is a tentative answer to research question B1:

Hypothesis B1:

It is possible to identify and quantify the costs of complexity for a product program and allocate the costs directly to the individual product variants.

Both papers G and H provide answers to research questions B1, and both in compliance with the preliminary answer of hypothesis B1, as the allocation of costs directly to individual product variants an integrated part of the presented 5-step approach.

As mentioned in the third part of the thesis, it is not the absolute level of complexity costs that is of interest to the author. On the contrary, it is the complexity cost reduction potential, which is a fraction of the total level and directly associated to the excess system complexity of the given product-process system (see chapter 3.2).

Hypotheses B2 provides a tentative answer to research question B2:

Hypothesis B2:

It is possible to rationalize a product program based on the calculation of complexity costs.

Especially paper G provides an overall answering to research question B2 by confirming hypothesis B2. Paper G shows how the calculation of complexity costs lead to opportunity of rationalizing the product program *reactively* and *proactively* providing input for avoiding complexity costs to increase again.

5.2 Core contributions

The frameworks and approaches are the primary contributions of the research. Along with literature studies, the research work has resulted in theoretical contributions too.

5.2.1 Frameworks

The frameworks span a wide area of potential uses and are applicable for use in a number of different situations. Papers A to F present architecture frameworks capable of supporting the identification of scalable program architecture for companies in a number of different situations:

Product vs. project-based development

- Product-based companies developing a definite solution space of product variants
- Project-based companies developing architecture for product customization

Scalable architectures for modular and integrated product structures

- Identifying scalable and modular architectures with modular product structures
- Identifying scalable architectures of complex parts with integrated product structures (incl. approach)

Architecture identification for new and existing product programs

- Proactive identification of architecture to cover future launches
- Identification of an architecture for an existing product program

5.2.2 Approaches

Paper G and H present the approach of five steps for reducing complexity based on the quantification of complexity costs.

Complexity reduction based on the quantification of complexity costs

Reactive rationalization of the product program to decrease complexity costs:

- Cleaning out unprofitable product variants based on an estimation of the true profitability
- Substituting less profitable product variants with more profitable ones

Proactive avoidance of complexity costs increase:

- Scoping of Design-for-X focus: Which life-phase has complexity cost reduction potential?
- Order fulfillment: Where to create product variants going forward (point of variant creation and/or customer order decoupling point)?

5.2.3 Theoretical contributions

- The definition of a program architecture as the result of a purposeful alignment of architectures across market, product and production domains.
- The expansion of the architecture concept to include the behavioral aspects in order to ensure a clear link to the critical goals of many companies being time-to-market and the promotion of an efficient 'R&D factory'.

- The formulation of an architecture of the sales phase – named a market architecture, promoting the commercial fit of the product program towards target market segments to increase competitiveness of product programs based on the program architecture.
- The ascertainment of the external factors as a basis for an actual classification of architecture initiatives to improve scoping of architecture initiatives, comparability between initiatives and transferability of experiences from specific initiatives.
- The definition of life cycle complexity cost factors as representing the life-phase meeting where the complexity cost appears.
- The definition of the complexity cost sensitivity of the life cycle complexity factors towards the causally related life cycle complexity drivers.

5.3 Evaluation of the research

This chapter includes the evaluation of the research presented in papers A – H including a validation of the contributions. The validation of papers and contributions is supplemented by the case study evaluation presented in section 2.5.2.

The evaluation will be carried out using the measurable criteria mentioned in the impact models (see Figure 2 and Figure 3), and the validation is performed using a mix of the proposed validation frameworks presented in section 2.5.1 with a focus on The Validation Square by Pedersen et al. (2000).

5.3.1 Evaluation of papers A and E

As the research results presented in paper A and E are both concerning the identification of an architecture for project-based engineering companies carrying out product customization, the evaluation of these are joint.

(1) *Individual elements:* As the frameworks are made on the basis of the widely accepted PFMP methodology (Harlou 2006), merged with the function-oriented focus on performance properties to identify coherent features, performance and production scalability, the individual elements making up the framework and widely accepted both in the academic world and in practice.

(2) *Internal consistency*: Because the frameworks represent combinations of the PFMP-based approach by a functionally oriented focus on performance properties in line with later contributions on classification of attributes (Andreasen 2011), the internal consistency is regarded rigorous.

(3) *Appropriateness of example problems:* Since the frameworks have been tested in case studies #1, #8, and #9 with global companies experiencing major challenges in identifying an architecture to control their product customization to improve profitability and time-to-market, the example problems are considered highly relevant for field testing of the frameworks.

(4) *Useful outcome:* From papers A and E it is reported how the benefits of application of the frameworks have contributed to a projected EBIT increase by almost 10%, and a time-to-market reduction of almost –50%, the usefulness of the method outcome is considered substantial.

(5) *Link between method application and achieved usefulness:* As it is theoretically impossible to separate the role of the author from the research outcome, this is a difficult theorem to fulfill. Therefore, to support the argument of a link between method application and result usefulness, the author can report that no ‘competing’ change agendas were present during any of the reported case studies, improving the probability of rightfully dedicating the achieved usefulness to the framework application.

(6) *Usefulness beyond example problems:* By making the actual case studies anonymous and by highlighting the general (but limited) applicability of the frameworks in other companies with similar external conditions and challenges, the usefulness beyond the case studies are considered positive. As with any framework, there is a need for adapting it to the specific purpose, and the frameworks are particularly tailored for companies in the specific situation mentioned in papers A and E.

In addition to the above accounting of elements leading to efficient and effective research according to Pedersen et al. (2000), verification has also been fulfilled by acceptance from the participants of the case studies (Buur 1990). The acceptance has been collected as a standard element in the evaluation meetings of each of the case studies.

5.3.2 Evaluation of papers B, C and D

As the research results presented in papers B and C are related in the way they propose modeling basics for the architecture framework presented in paper D, they jointly address the proactive identification of a program architecture tailored product-based companies.

(1) *Individual elements:* Since the architecture framework builds upon the PFMP methodology by adding a very basic system theoretical aspect of behavioral modeling of market, product and production architectures, the individual elements are considered scientifically rigorous. While the input for all of the three aspects are referenced in the papers, the additions of modeling of performance properties versus design characteristics and cost, and the addition of evaluation-focused modeling techniques are all in line with basic contributions from engineering design research.

(2) *Internal consistency:* The architecture framework is consisting of a number of architecture models that are mutually linked. As from Figure 18 and Figure 20, the links are a central part of the framework enabling the traceability of consequences and opportunities across the market, product and production side of program development. However, the links are not embedded in a fixed way among the architecture models, as every architecture initiative can pose the need for linking the views on various levels. Thus, the internal consistency between the models is considered fulfilled, despite being contingent upon the competence and discipline of practitioners during application.

(3) *Appropriateness of example problems:* The architecture framework has been tested in 11 case studies with active participation of the author, while a subset of the framework has been tested in the *Radikal Forenkling* (in Danish) project by a number of teams of master students specializing within platforms and architectures (see publication K). There are several common denominators among the case companies and their situations being under pressure for improving time-to-market while being challenged to manage a product program with growing complexity.

(4) *Useful outcome:* As reported the outcome has been varying from strong indications of time-to-market reductions to recorded and confirmed results.

(5) *Link between method application and achieved usefulness:* As stated in section 5.3.1, there is no evidence pointing at other likely explanations for the achievement of the reported results. This is not evidence in itself, but as the author has been centrally situated in the development projects in question, there has not been recorded any disturbance of the method application that could indicate that the link between the method application and achieved usefulness is not strong.

(6) *Usefulness beyond example problems:* The strength of paper D is the broad experience base from which the case studies have contributed to valuable experience. The broad base of experience is considered an indication of usefulness beyond example problems themselves, and the contextual awareness of external factors affecting the case studies (see Table 5) contributes to this indication. However, the need for adapting the architecture models to the specific situation does not reduce by this.

As with the previous section, the verification by acceptance has also been a central ingredient here. Being engaged in industrial case studies and working closely with practitioners requires a great deal of attention from the company experts as well as access to data from the case company. This is very difficult to obtain without the acceptance from practitioners, as well as it is difficult to arrange and engage in follow-up case studies without approval and acclaim from work done in the previous case study.

5.3.3 Evaluation of paper F

As paper F does not represent a closed research work, but merely an ascertainment of external factors affecting a future classification of architecture initiatives, the evaluation of paper F is carried out including this concern.

(1) *Individual elements:* The external factors selected as differentiating criteria originate in a broad body of knowledge across research in engineering design and operations management. Thus, they are not new, but merely selected to represent the variety of external factors affecting the industrial setting that an architectures initiative should have effect in.

(2) *Internal consistency:* The consistency between the elements is considered covered by (1) and can be further elaborated by the paper's reflection including the classification systematics from TTS, of which the external factors share similarities.

(3) *Appropriateness of example problems:* The paper does not report on details from specific case studies, but merely on the general experiences in taking the ascertained external factors into account. However, the paper does propose to use the external factors in relevant situations of scoping and comparing architecture initiatives.

(4) *Useful outcome:* The outcome is so far embedded in the results reported in papers A to E, as paper F does not represent a specific field test.

(5) *Link between method application and achieved usefulness:* See (4)

(6) *Usefulness beyond example problems:* The usefulness has proven itself in the case studies reported in papers A to E, for example by highlighting the need for different solutions whether the external product of 'primary driver for product positioning' is focusing on sheer performance (framework presented in paper E) or feature multiplicity (framework presented in paper A). This sets key boundary conditions for the architecture initiatives that results in top-down oriented approaches (paper A) and bottom-up oriented approaches (paper E).

5.3.4 Evaluation of papers G and H

Papers G and H present the 5-step approach to complexity reduction by quantification of complexity costs. Although papers G and H are presented with slightly different purposes – referring to research questions B2 and B1 respectively – they are evaluated in their entirety.

(1) *Individual elements:* The individual analysis elements of the approach are created using a variety of literature of complexity costs from operations management and cost accounting research as well as experience, hence providing a thorough basis for deriving individually rigorous elements.

(2) *Internal consistency:* The primary elements of novelty of the approach is on one hand the identification and quantification of the complexity cost factors leading to an estimation of the true profitability of product variants, and on the other hand the approach's straightforward use and clarity of execution. The internal consistency is ensured by calculating the contribution ratios consistently, before and after adjusting for complexity cost factors, as well as quantifying the complexity factors using the same conversion into net revenue related costs, enabling the overall adjustment of the contribution ratio and margins for individual products.

(3) *Appropriateness of example problems:* As reported in
Table 1, the case studies #10 to #15 have tested the approach representing similar situations of

companies experiencing a growing product program with decreasing average contribution ratios due to an increasing product-process complexity. The case studies are therefore considered appropriate for testing the approach.

(4) *Useful outcome:* From papers G and H the outcome regarding short term complexity cost reductions were scenarios indicating several percentage points of EBIT improvement. This outcome is augmented by long term complexity cost reductions due to an improved decision base for product managers, due to the estimated true profitability being known.

(5) *Link between method application and achieved usefulness:* As the applied complexity analyses has improved the current decision basis for product management functions by a significant level, there has been no evidence proving that the cause of the results was not derived from the method application. Furthermore, the link has been expressed evidently by central decision makers with close involvement in the case studies.

(6) *Usefulness beyond example problems:* By the presentation of the 5-step approach, it has been outlined how the approach fits particularly well to companies being the result of mergers and acquisitions, lack of common IT-support for reporting of key figures, and with consumer products in multiple regions making it difficult to apply traditional tail-cutting exercises without losing revenue irrevocably. However, in addition to the already reported field tests the 5-step approach has also been tested in a manufacturing company in case study #21 with remarkable results. The detailed reporting of case study #21 is out of scope of the thesis, yet, the case study reveals that the usefulness beyond the reported case studies seems very promising, as the company experienced significant productivity gains from clearing out intermediate goods classified as C-products with regards to complexity costs.

5.4 Boundary conditions and limitations of results

5.4.1 Research method

Problem-based and theory-based

The contributions presented in this thesis are the results of research work conducted from both a problem-based starting point as well as a theory-based (see section 2.4.1). Both have been equally important as fruitful research paths to enable the development of the presented results, and both of them are dependent on the insights from each other.

Action-research inspiration

The practical problem-based research path shares resemblance to the work approach of action research, thus making it impossible to extract the author's role of being a researcher from the equation. Therefore, an important task in action research is to declare the theoretical basis of the research as presented in Part 3 and in several sections of the appended papers (Karlsson 2008, Checkland and Holwell 1998). This is to declare in advance the starting point from which the research in action is building upon. The action research-based approach is primarily relevant for the contributions concerning identification of program architectures, whereas the contributions concerning complexity reduction based on the quantification of complexity costs mainly originate from theory-based research work. However, as stated earlier, the problem-based and theory-based research paths are inseparable in the way they complete each other's need for theoretical foundation and practical application, respectively.

Recoverability, generalizability and transferability

As a basis, action research should be recoverable. Recoverability has been ensured by extensive journal keeping in case studies, making it possible to 'roll back' the line of events and review meetings to document research progression.

The research has been generalized by putting the case studies into context. The reflection is provided by the presentation of each paper in Part 4, including paper F that outlines the varying external factors of the research in architectures. This reflection has the aim of accounting for generalizability as well as the preparation for transferability. As the research work includes 23 case studies, the transferability has initially been proven already, whereas verified evidence will await the further application of the research across various industrial settings and challenges.

5.4.2 Identification of program architectures

Regarding the research contributions centered on the identification of program architectures, from a practical perspective, a number of prerequisites exist for succeeding in practice. This section includes the ones recorded during field studies.

Appropriate staging

As elaborated in section 5.1.1 the architecture models presented require appropriate staging in order to be able to capture the critical aspects of market, product and production areas.

Practice in modeling techniques

The application of sophisticated, but yet simplistic, modeling techniques require experience and practice. A Master's background within mechanical engineering or product development is a given, as a vast amount of data often need to be interpreted and scanned for product program development relevance in order to be prepared for architecture modeling. This is a task that requires training.

Orchestrating reviews

In order for the architecture models to have relevance, it is necessary that they are updated on a frequent basis. It is also important that they do not lag behind the momentum of the development project for which they should support decision making. This pitfall has the consequence of making the models obsolete by reflecting historical versions of the program design.

To keep the architecture models pertinent, it is of great practical importance to maintain a steady frequency of intra-domain reviews and cross-domain reviews (see section 5.1.1), leaving time for clarification runs in between. Typically, changing between intra-domain and cross-domain reviews provides an efficient working model allowing sufficient preparation time for participants and other meetings.

Inclusion of key competences

The difficulty in designing product programs is significant compared to the design of single products or product families. Due to the size, product programs entail larger risks but also a substantially greater potential for achieving benefits. To support the identification of a program architecture, it is therefore of great importance to include competences from senior resources across market, product and production domains. It requires a broad experience across an industrial company to make cross-functional decisions that are truly forward looking and have the potential for reducing complexity and increasing competitiveness. It is the experience of the author that it is the fruitful interaction between the architecture models presented in this thesis and these types of competences that can make this happen. The inclusion of key competences can also be the key to the necessary ownership of decisions following the identification of a program architecture. As described in section 2.3.2 it is out of scope of this thesis to research in the maintenance and 'sustainability' of program architectures.

5.4.3 Complexity reduction based on the quantification of complexity costs

Also regarding the research contributions centered on complexity reduction based on the quantification of complexity costs, from a practical perspective, a number of prerequisites exist for succeeding in practice. This section includes the ones recorded during field studies.

Access to data

As the 5-step approach describes, it is of fundamental importance to gain sufficient access to data on a number of key figures from sales, production and distribution. Such access unfortunately requires the research results to be anonymized, as this is highly confidential data to most industrial companies.

Identification and quantification of complexity factors

Apart from the described top-down cost structure view or bottom-up hypothesis of cost asymmetry (as described in paper G), the identification of primary complexity factors can be difficult without the inclusion of key resources – for example in a workshop format. In continuation of this, the quantification part can be practically difficult for several reasons. The provision of quantification objects to make a sufficiently reliable quantification may require additional workarounds, which can reach a level of sophistication that is difficult to verify and confirm from other sources.

Access to key competences

In accordance with the need for access to data, the need for cost accounting competences is also of vital importance for the application of the 5-step approach. It is crucial to use data of a high quality with a correspondingly high certainty of exactly which products and processes that are actually recorded in the data. Another source for data clarification is key resources from the relevant life-phases in question which can support the correct interpretation of the data and verify that assumptions are justifiable and in accordance with the products and processes behind.

5.5 Evaluation of the research impact

5.5.1 Academic impact

From Figure 4 the diagram showing the areas of relevance and contribution show how this research is considered a contribution to the areas of

- Product architecture
  - Product platform
  - Modularization
- Complexity management

As from the definition of Harlou (2006), a platform can be seen as an instance of an architecture, making it equally relevant for the research presented here.

Program architecture

From chapter 5.2 the list of core contributions report how the concept of a *program architecture* expands the current notion of architecture for product programs by including a number of additional constitutive and behavioral aspects.

The author considers this an important academic contribution, as it underlines the role of modularization as being a means to achieve effects – not a goal itself. This is further elaborated by the contributions from paper E, where a scalable architecture for complex parts is identified beyond the traditional physical interfaces.

The contribution also opens for a more pragmatic take on the existence of architectures. Many companies might be struggling with architecture initiatives over a period of time, and previous contributions within the theoretical area were divided between either stating that

- *Any* product program can be considered as having architecture as the architecture was considered the aggregate of all the product structures included (mainly from American contributions)
- *Only* product programs with a situation specific optimized alignment between market, product and production aspects can 'plead' the architecture nomination.

The contributions of the author slightly alter this debate in the direction of

> Any product program can have elements of program architecture thinking in place, but the architecture *improvement potential* is related to
>
> - the lack of alignment of constitutive aspects
> - the lack of behavioral aspects' ability to meet the critical goals of the company at question (examples include competitiveness, time-to-market, R&D efficiency etc.)

Linking architecture and complexity reduction

The contributions presented here represent a step towards the further integration of the areas of architectures for product programs and the quantification of complexity costs. Still, many areas remain uncovered within this cross-field to be conceptualized by new contributions, but the formulation of the life cycle complexity cost factors and approach for detecting, identifying, quantifying and allocating complexity costs is seen as an important step. There is no future limit to how sophisticated allocation algorithms can be, or how advanced the proactive use of the complexity cost information can be, but the contributions presented here are considered as providing a basis for these further endeavors.

5.5.2 Industrial impact

Given that the empirical scope of the thesis is companies operating out of Scandinavia, the evaluation of the industrial impact is based on the experience derived from these.

Preparation and execution

Many Scandinavian contributions from recent years are highlighting the ability of architecture-based product development to split the development task in a preparation and execution phase as a virtue of the implementation of architecture-based development (Nielsen 2010, Kvist 2009, Pedersen 2009, Harlou 2006).

The experience from the case studies conducted by the author is that even though this might seem an intriguing goal; it is often not the best 'vehicle' for introducing truly architecture-based product development. This does not mean that the basic principle of preparation and execution is suggested to be abandoned, but rather that the transformation of a company into an architecture-based one entails such a huge 'complexity' in itself, that it can be very difficult to identify and design a program architecture separated from the actual development activities.

Online and offline projects

In order to account for the variation between activities of architecture identification being carried out as a part of a development project or not, there is a need to distinguish between *online* and *offline* projects.

Architecture projects denoted as online are architecture identification activities carried out to support an ongoing development project. The larger the project is, the higher the relevance of identifying an architecture is, due to the project's possible influence across market, product and production domains being larger here leading to an increased improvement potential (ref. definition from section 5.5.1).

Architecture projects denoted as offline are architecture identification activities carried out decoupled from ongoing development projects in the sense that no development activities are occurring alongside the offline architecture identification project.

Pilot and assessment projects

The experience collected during this research project is that both online and offline projects can be feasible vehicles of identifying program architectures, yet, for these projects to succeed there is a need to carefully orchestrate each one of them.

For companies with little or limited experience in architecture-based product development, *online pilot projects* are considered a such feasible vehicle. It is the experience of the author that the careful selection of a representative product program is vital here, as this will enhance the transferability of experiences from the pilot project towards subsequent implementation activities. The online pilot project does not initially differentiate between preparation and execution activities, as this distinguishing becomes obsolete considered the development activities leading to the first product launch. However, regarding the whole online pilot project, the differentiation between preparation and execution activities becomes viable in the sense that the online pilot project should take on the challenge of preparing the product program for the launches subsequent to the first launch.

For companies in need of assessing the architecture improvement potential before embarking in online development activities, the initiation of an *offline assessment project* is considered a feasible vehicle leading to the identification of a program architecture. The offline assessment project can be used for screening market, product and production domains to uncover the hidden potential of identifying a program architecture. The offline assessment project can screen for the lack of alignment between the constitutive aspects (what the architecture *is*), of the current product program as well as the gap between the behavioral aspects (what the architecture *does*) and the critical goals of the company in terms of competitiveness, productivity, time-to-market etc. It is the experience of the author that an offline assessment project leads to an improved scoping and prioritization of which architecture initiatives to pursue and in which sequence. Especially in global companies where regional differences pose a major challenge to the definition of program architecture focus.

Key resources

As described in section 5.4.2 the inclusion of key resources in the architecture work is a pre-requisite of succeeding in the identification of a program architecture, due to the level of criticality and difficulty in achieving the cross-domain alignment central to the architecture work.

Architecture virtues

Having being engaged in several industrial case studies, a number of common denominators seem to repeat from project to project and from company to company. They are common in the sense that they capture central elements of *program architecture thinking*, thus serving as guidance for architecture identification. These are denoted *the architecture virtues*.

From (Olesen 1992) the seven universal virtues are formulated: *Costs, throughput time, quality, efficiency, flexibility, risk and environment*. The universal virtues are based on the experiences of the author and earlier contributions from (Büchler 1990, Rode and Sant 1983, Skinner 1974), and they address the ultimate goals of most companies situated in the top of a hierarchical goal structure. The architecture virtues in Figure 24 do not represent such ultimate goals, and they do not serve the purpose of replacing the universal virtues, but merely guiding how architecture initiatives can maximize their adherence to these.

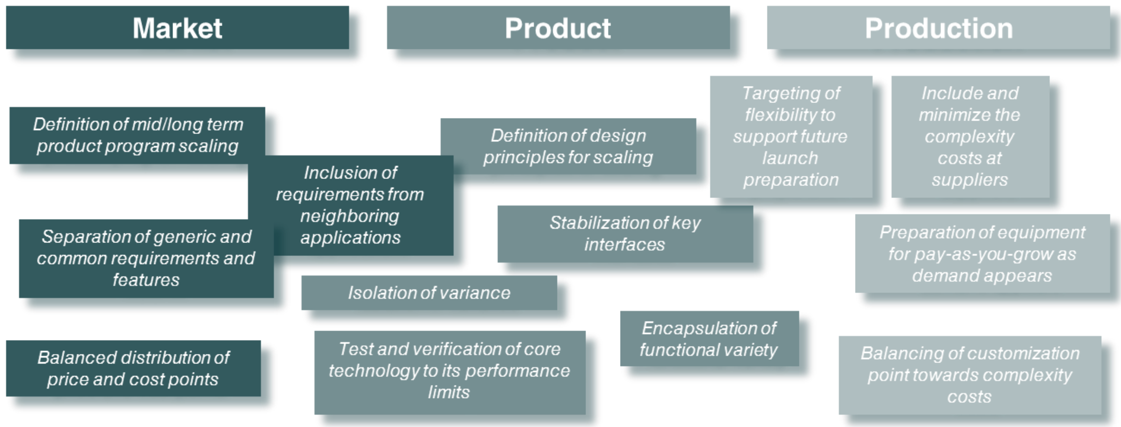


Figure 24 – The Architecture Virtues

The architecture virtues should not be interpreted as dogma setting strict rules for decision making in architecture projects, but rather as a high-level checklist serving the purpose of supporting a high-level evaluation of the scope of an architecture initiative.

Enhancing the preparation level

In architecture projects the concept of *preparation level* often appears as a central means for achieving downstream reductions in time-to-market. Therefore, a popular interpretation of the core of architecture-based product development is that the preparation level is a key enabler for achieving many of the desired effects.

The research work has identified a number of techniques from various industrial projects for enhancing the likelihood of an improved preparation level. As the concepts are used differently across different companies, the list below includes the interpretation of the author:

Volume of activity (VOA) denotes the volume surrounding a module that should be reserved for various activities encountered during interaction with the life-phase systems. Examples include VOA for assembly, VOA for service etc. The definition of VOA can enable an improved level of module reuse and eliminate non value-adding design activities used for adapting subsequent designs to accommodate the module.

Volume of control (VOC) denotes the volume surrounding a module that should be reserved for future upgrades of the module. The VOC can be defined by superimposing future upgrades balanced towards the likelihood and cost of implementation, in order to identify the *maximum configuration* of a group of products or a production line setup. The definition of VOC can enable an improved interchangeability of modules, as the surroundings of the module can be designed to accommodate the future variance encapsulated in the module. It is noted that the definition of VOC can be time-consuming if the verification level is desired to be close to 100%, however it is the experience of the author that a significantly lower verification level represents a much more feasible ratio between cost of preparation and future benefit.

Ghost designs denote the level of design completion where a feature or option of a product or production setup is fully designed, prepared and verified, but not yet implemented. Ghost designs are relevant in terms of critical features which cannot be decided upon during the development phases, and therefore requires implementation flexibility beyond the first product launch. Thus, ghost designs become configurable elements with an extremely short implementation lead time when the demand appears.

5.6 Suggestions for further research

The frameworks presented here open up for an array of contributions refining and detailing various modeling techniques of market, product and production architectures in order to promote the alignment into program architectures providing competitiveness for the benefit of companies and society:

Inclusion of more quantitative aspects to develop program architectures that are even more justifiable and supportive of the critical goals of the company

A higher level of formalization of modeling techniques to support the teaching and practicing of novices in the techniques of architecture modeling.

A more rigorous articulation of the role of interfaces in architecture identification. Stabilization of interfaces is often the key to identifying 'sustainable' program architectures, and there is a need to define and classify the nature of interfaces to a higher level of comprehension.

An exploration into architecture work and organizational setups describing how different architecture organization can be matched against the task posed by contextual factors.

A complete classification of architecture initiatives building on the definition of the external factors presented in this thesis. The classification should aim for devising a recommended match between the contextual factors and the type of architecture initiative.

An exploration into a concept of a company encompassing architecture perhaps bridging the concept of a program architecture presented in this thesis with the work by Miller (2001) on activity and knowledge architectures.

5.7 Concluding remarks

It is has been stated by many previous PhD students that the research work represents a journey for the researcher. For me, the journey has consisted of several tracks.

The theoretical journey into the world of research has been exciting in the sense that its epistemological side has helped to understand how notions and ideas that may seem different from appearance, all are connected in a complex web consisting of different views and languages.

The practical journey into the world of industry has been equally exciting in the way that I consider industry the *laboratory* of research in engineering design research. Therefore, I have been eager to undertake case studies when the opportunity has emerged providing a fruitful shifts of scenery between the theoretical world with peers at conferences and at the university and with practitioners in industry.

And last but not least, the mental journey from being an apprentice researcher into an author of a thesis.

I hope it has been possible to pass on a share of the enthusiasm and dedication from which the presented contributions have been derived.

6 Appended papers

6.1 Paper A

MAKING PRODUCT CUSTOMIZATION PROFITABLE

Niels Henrik Mortensen[1], Lars Hvam[1], Anders Haug[2], Per Boelskifte[1], Christian Lindschou Hansen[1] & Simon Frobenius[1]

Department of Management Engineering[1]
Product Architecture Group
Technical University of Denmark
Building 426
DK-2800 Kgs. Lyngby
Email: nhmo@man.dtu.dk, lhv@man.dtu.dk , pbo@man.dtu.dk , chrlh@man.dtu.dk, sfr@man.dtu.dk

Department of Entrepreneurship and Relationship Management[2]
University of Southern Denmark
Engstien 1
DK-6000 Kolding
Email: adg@sam.sdu.dk

The main result presented in this paper is the Framework for Product Family Master Plan. This framework supports the identification of a product architecture for companies that customize products and services. The framework has five coherent aspects, the market, product assortment, supply-production, organization and work processes. One of the unique results is that these aspects are linked, which make it possible to make explicit recommendations for an architecture (the way a product family should be structured with clear interfaces), architecture elements and consequences. By means of a case study it is shown that the potential EBIT (Earning Before Interests and Taxes) improvement of the case company is 10%.

Significance: Many companies make customization, but have severe difficulties becoming profitable. This paper suggests a framework for identifying an architecture that can provide a basis for increasing profitability.

Keyword: Product Family Design, Architecture, platform, product development.

(Received: Accepted:)

1. INTRODUCTION

For many companies that deliver customized products and solutions it is often very difficult to make satisfactory EBIT (Earning Before Interests and Taxes). Below is an example from a company making customization of building equipment. The Gross Margin (GM) distribution across projects has a variation as shown in figure 1.

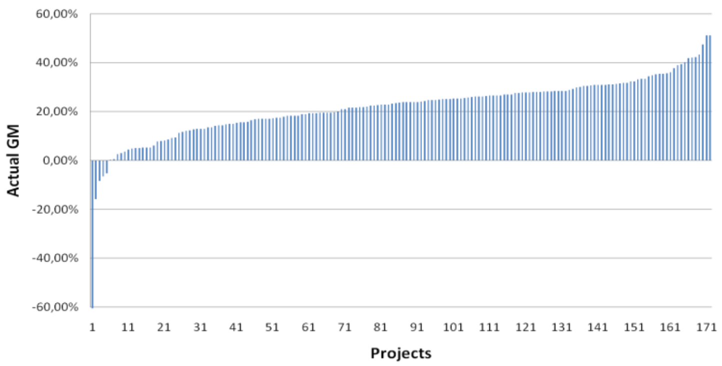


Figure 1: Gross margin for projects

The x-axis shows individual projects in the company and the y axis shows the actual gross margin in each individual project. In the company above all quotations are calculated based on a gross margin of 20%. As the diagram shows, quite a significant amount of projects have a margin far below 20%. What can also be seen is that some projects have a higher margin than 20%. It is clear that there is significant margin deviation in the majority of projects. As a result this, the company has an overall EBIT of a few percent some years and a negative EBIT in other years.

The above situation is no single incident but is more or less the general picture for the 40 companies that we have been working with over the last years, Hvam, Mortensen & Riis (2007). There are exceptions but the majority of the companies making complex engineering customized products have severe problems making a satisfactory EBIT. In periods where it is relatively easy to get new projects these companies could make more money if they were more professional in selecting the profitable customers and projects. The situation today with financial crisis many companies try to shoot at everything, leading to a situation where many projects become even less profitable.

There are certainly many reasons for the lack of ability to make a satisfactory EBIT. This paper is based on the assumption that EBIT can be improved significantly if a more clear architecture for the product assortment is designed and implemented. A clear architecture means that the building blocks in the product assortment and related interfaces are clearly defined. This further means that the organization, processes and systems are designed to handle this architecture. In other words the architecture describes where the company can make profitable projects. A further implication is that a company will then be able to recognize a profitable customer or project upfront. The reason is that a relative stable architecture means that robust and optimized processes can be developed. If the architecture, including core interfaces is changed in each project, it is nearly impossible to develop robust processes across the whole company.

This paper will propose an operational framework, Framework for Product Family Master Plan, which can serve as a basis for identification an explicit architecture. The research is based on PhD, Master and consultancy projects in more than 40 companies mainly within Scandinavia. Before going into the framework, we will first examine some of the barriers for implementing architectures. Then the framework is presented and is then related to other research work and finally experience from application of the framework in an industrial company is presented.

2. BARRIERS FOR MAKING A CLEAR ARCHITECTURE

Most Board of Management teams recognize the need for having a more clear architecture. The phenomena have many different phrasings such as standardization, preferred solutions, platforms, fast track, mass customization, lean processes etc. They all somehow express the intention that there should be two execution processes, a fast track and a standard track. This is shown in the figure below.

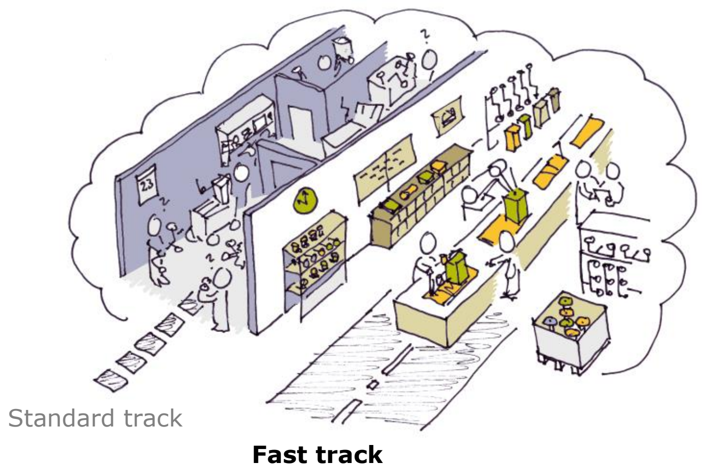


Figure 2: Clear distinction between fast track and standard track. Projects within the architecture shall be executed as fast track and projects outside the architecture shall be executed as standard track or simply rejected.

The standard track is based on the architecture and the standard track is covering the remaining projects. Often there is no clear separation between the two tracks leading to a lot of complexity and inefficiency in project execution.

Despite that the vision for many companies is expressed more or less explicit, very few companies seem to be able to implement it. This section will summarize the most common barriers that companies have expressed during architecture projects.

"We deliver value to our customers by delivering exactly what is requested" This will sometimes be true but in many cases not. There are many examples of variety in a product assortment that does not provide value to customers but only add complexity cost in companies. A few examples of this phenomenon are. One company is delivering products with actuators that are bolted, welded and glued. This means that three types of production processes have to mastered, leading to increased cost. Seen from a customer point of view this variety does not add value. Another company is

having pressure tanks certified for 4,1 bar, 4,4 bar and 5 bar. In this case certificates and approvals have to developed and maintained without any extra value to the customers.

"We will loose our customers" This might be true, but there is a clear tendency that the companies trying to shoot at everything make significant less money than the one with a clear and focused market strategy. Without an architecture it is difficult to recognize a profitable customer. An architecture makes it more clear where a company can develop and deliver customized products with satisfactory EBIT. Thereby a significant better decision basis is available for accepting or rejecting a project or a customer. Some of the companies that we have studied have reduced turnover due to fewer customers, but the customers they have contribute to an increased margin.

"Product assortment ownership is unclear" Companies develop products within projects with a clear responsibility as long as it is a project. But after the project and on portfolio level the responsibility is often unclear. There are procedures for how to introduce new products and components, but often there are no procedures for removing them. The criteria for accepting or rejecting a project are often very fuzzy.

"We do not know where to start" In many companies the product assortment is so complex that it is difficult to get an overview. Over the years product programs often become extremely complex, due to customization, acquisition of new companies and reinventing the wheel in projects.

"Architectures is a looser project – do not touch it" To succeed with an architecture project commitment from sales, engineering and production is necessary. In sales a company might have so say no to customers, in engineering a more clear focus on sharing and reuse is needed and in production capabilities have to be adjusted according to the architecture and vice versa. Who is responsible for the link between sales, engineering and production? Ultimately it is the CEO. Often we see architecture initiatives isolated in the sales, engineering and production but they are not coordinated and do therefore not create significant benefits.

"Benefits are difficult to quantify" It is difficult to prove that companies with a more clear architecture make more money than companies without, but among the companies that we have studied there is a clear tendency. Among the most important reasons why benefits are difficult to quantify is that savings often are related to decreased overhead costs. The increased ability to develop new products, shorter lead time is not easy to quantify. Many sophisticated techniques are available in literature, such as activity based costing and total cost management. In the companies we work with more simple techniques have been utilized. The main technique has just been to evaluate what task are added or removed in each functional area in a company.

There is no simple answer to address the above barriers, but one aspect that can support decision making is a systematic approach to identify what could be concrete architectures, architecture elements and benefits. One such approach is the Product Family Master Plan Framework.

3. FRAMEWORK FOR PRODUCT FAMILY MASTER PLAN

This section will briefly describe the framework that has been utilized for analyzing a company and the product assortment with the purpose of identifying an architecture. The intention has been to describe a framework which can support answering the following questions:

1) What are the existing variety of the product assortment, seen from customer, functionality and part point of view?
2) Is the variety creating value for the customers?
3) What sort of complexity is created in production, due to product variety
4) What are existing variety in production and production processes?
5) Which production processes are creating value?
6) What variety exists in the work processes when customer specific solutions are designed?
7) What are the dispositional relation between the product assortment variety and the order and development process?

The product modeling basis in the Product Family Master Plan Framework is the Theory of Technical Systems (TTS), Hubka (1988) and Theory of Domains (ToD), Andreasen (1980). According to TTS and ToD a product can be modeled from four points of views: process, function organ and part. A process describes the transformation that a product is able to do. A coffee machine is able to carry out a process where water and coffee beans are united in to coffee. The functions are defined as the purposeful effects necessary to carry out the process mentioned above. In the coffee machine example the effects necessary are creation of heat, filter the coffee beans and the water. The organs are the elements which are able to realize the functions. Examples of organs in a coffee machine are the heating element, the filter, the can and the chassis. The parts are the physical elements which in an interplay is able to realize the organs. Example of parts in e.g. the heating element might be the wires, the screws, and plates. In the framework processes and function view are combined in to what is called a customer view. The customer view is then the subset of the processes and functions that are relevant seen from a customer point of view.

Object oriented models consist of objects and relations. An object is characterized by its identity, structure and behavior. According to Coad & Yourdon (1991), relations can be whole-part, generalization - specialization, message connection and instance connections. Whole-part (part of) is relations between entirety and elements. Generalization - specialization (kind of) are relations between super and sub classes. Message connections are data flow between objects, e.g. that calculation in one object requires data from another object. Instance connections are relations between classes and instances. TTS and ToD is describing single products whereas object oriented modeling adds variety in such a way that product families can be modeled.

The Product Family Master Plan (PFMP) is originally proposed by Harlou (2006) and is based on TTS, ToD and Object oriented modeling. In this paper the PFMP is expanded by means of a market, supply, organization and work process dimension. The totality is named Framework for Product Family Master Plan, se figure 3 below.

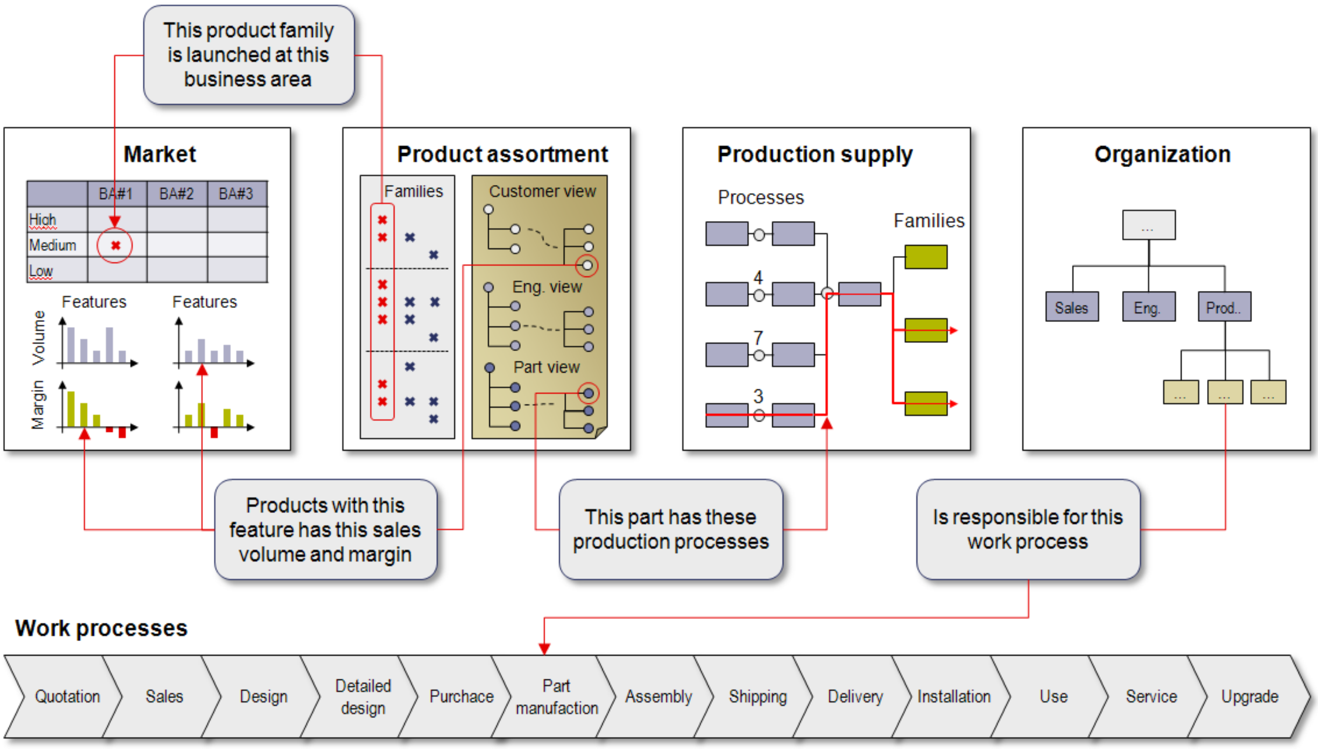


Figure 3: Framework for Product Family Master Plan

The Product Family Master Plan (PFMP) Framework consists of five aspects. The market, product assortment, production supply, organization and work process aspect. In this section the framework and the contents is explained further.

Product Assortment aspect

Starting with the Product aspect, it consists of a customer view, engineering view and part view. Each view consist of two sub structures a part_of structure and a kind_of structure. In the customer view all features that are of importance to the customers are described. The engineering view contains the functional units (organs in the TTS) and variants within the product assortment. The part view describes the physical elements and variants of the products.

What is of special importance is the links between the views. Each of the views is causally linked meaning that certain types of traceability can be described. The relation between customer view and engineering view describes how certain customer features are realized by means of certain functional units. The relation between engineering view and part view explains how functionality is realized by means of physical parts and sub-assemblies. Reading the from the part view to the engineering view explains how a certain part contributes to delivering functionality of the products. From the engineering view to customer view the relation describes how functional units deliver customer features and, hopefully, value to the customer.

From these relations a number of important conclusions can be derived. Some examples are:

- The more relations that exist from a certain feature in the customer view to functional units in the engineering view, the more complexity are the product assortment inherent. This means that if a feature in the customer view is changed; all the related functional units and subsequent parts have to be changed or updated.

- The relations between the engineering view and the customer view often reveal that certain functional units in the engineering view have no explicit relations to the customer view, meaning that a none-value adding variety has been added to the product assortment.

Further more the commercial variants of the product assortment are mapped. This means that it can be clearly seen what in terms of customer features, functional units (organs) and parts that are shared across the Product Family of the products. In figure 3 three product families, A, B and C are described.

Market aspect
In this area two sub aspects are described. The Power Tower matrix (shown in the upper part of the market aspect in figure 3), Meyer & Lehnerd (1997) sales volume and turnover for each customer feature is mapped. The Power Tower has two dimensions. The first one is business areas (BA in figure 3) and the second one is the high, medium and low end products. The power tower provides a good overview on how the product families are covering the whole market. When deciding on an architecture, it is of high importance to determine which areas in the Power Tower matrix it shall cover. Mapping the customer features in terms of sales volume and contribution margin provides a good overview on the consequences of adding or removing a feature to the product assortment.

Production – supply aspect
This is in principle the same as the product assortment aspect – just that the production processes are the modeling object. The principle here is a generic process diagram, meaning that all process is mapped on a generic level. The production flow of each family is then mapped in the process diagram. For each process, the numbers of part or assembly variants are described (described in the circle after each process). A shadow behind the boxes indicates that variants of the process exist, e.g. two different welding processes are carried out. The production flow for each family is described in the production – supply aspect (this is shown by means of the lines connecting the boxes and end up in individual families, A, B, and C). Mapping the production flow of individual families, gives an overview of how much of the production processes and equipment that are shared across the product families.

Organization aspect
In this aspect the persons and departments being active in the sales, delivery and production of a product variants is mapped. The purpose is to see how many times change of ownership is carried out in the chain of work processes. Mapping the different IT systems utilized also provides valuable information for determining the complexity that have to be handled.

Work processes
In this aspect the generic work process in the company is mapped, i.e. quotation, sales, design, purchase, quality etc. When deciding on an architecture it is easy to optimize against a few work processes, but it has to be optimized with respect to whole work process chain in order to provide significant benefits, e.g. lead time reduction. The next section will briefly explain the relations between the different aspects. It is due to the explicit relations that the most important conclusions can be derived.

Product assortment – Market aspects: There are two different relations shown in figure 3. The first one is relating product families and the power tower. This relation shows in which business areas the product families are marketed. The second relation shows how classes of features relates to product sales and volumes.

Product assortment – Production supply aspects: This relation shows how individual parts are manufactured in a sequence of production processes. By means of this relation commonality between parts and production processes can be visualized.

Organization – work process aspects: The relation shows which part of the organization that is responsible for the individual work processes.

In section 5 the case study will explain how the above framework has been utilized as basis for making decisions on a product family architecture.

4. EXISTING APPROACHES IN LITERATURE

The following review is based on the results from the phd dissertation of Morten Kvist, Kvist (2009).

Modular Function Deployment: The modular function deployment (MFD), Ericsson & Erixon, (1999) builds largely on the methodology of the QFD and on the formulation of eight so-called module drivers. The purpose of MFD is to enable cross functional teams (including mainly marketing, development and production personnel) to create a mapping from the physical structure of the products within a family to the functional structure of those products and to ensure that the functional structure corresponds to the demands of the customers. Modular Function Deployment method consists of five consecutive steps. Customer requirements are mapped to functional criteria and subsystem design characteristics and subsequently forming a physical design in which a modular architecture supports a carefully selected set of modularisation incentives called module drivers.

Design Structure Matrix: This approach takes a starting point in the decomposition of a product into components/systems and an identification of interfaces/relations among these, Pimmler & Eppinger, (1994), Höltta-Otto & De Weck, (2007). By the use of algorithms, it is possible to encapsulate components into modules or chunks that are closely related to each other from an interaction point of view [Steward, 1981]. This process is referred to as clustering. The outcome of a DSM is a proposal for a future modular product architecture.

Generic Bill of Materials: The generic BOM originate from the assemble-to-order environment, van Veen & Wortmann, (1987) The end-products typically have a number of features for which a number of options are available to choose from. Not many options are required in order to make the number of combinations (i.e. end-products) enormous. The number of end-products can easily become too large to able to define specific BOM's for every single combination. Furthermore, forecasting, BOM-storage and maintenance become unmanageable. The generic BOM is a concept that is introduced to enable creation of a specific manufacturing BOM when the customer places an order, by replacing. The generic BOM is used to describe related products in one all-embracing model by using generic and specific items.

Decision tree: The decision tree, Rea, (1965)is used by Tiihonen & Soininen (1997)as a product configuration model, which basically represents all the valid combinations of the components that can be used to obtain the desired functions for the customer. The product configuration model, Mesihovic & Malmqvist (2004). The decision tree presents the multitude of component variety within a product family and by the use of positive combinatory relationships (e.g. if "engine size"=D13 then "engine power" must be 360 or 420 hp) and/or incompatibility relations (e.g. if "engine size"=D13 then "engine power" cannot be 220 or 700 hp) it defines the possible product configurations.

Value analysis: Value Analysis is a discipline founded at General Electric in the late 1940's , Fowler (1990). In short, value analysis is a methodology that has as its purpose to relate cost with functions in a product. It is a stepwise methodology in which a product is partitioned into smaller constituents for further analysis – that may be analysis of cost or value. Value is not the same as the Japanese idea of customer value we may see within the lean paradigm. Value is specifically defined as the "worth" relative to cost, i.e. value = worth/cost. Worth in this sense actually resembles the idea of customer value in lean very well. It is a denominator of those aspects, functions and features a customer wants to pay extra for. The customer is regarded as the downstream stakeholders in the supply chain. Worth is – in other words – a function of the totality of needs and demands of the customers, the customers' customers, the distribution channel etc. Some practitioners try to quantify worth and relate it directly to cost. Obviously cost is rather quantitative and measurable in hard currency, while "worth" is a more soft and qualitative size. Whether qualitative or quantitative, value has a focus on identifying value elements from a customer perspective and relate it directly to the functions of the product and thereby indirectly to the way the products are built.

Function structures: The function-based design methods are characterized by the establishing either a function model Pahl & Beitz, (1996), Otto & Wood (2001) or the schematics of the product Ulrich & Eppinger (2000). The function structure describes the flow of material, data, and energy through sub-functions of the product using a set of rules (e.g. the rules that are referred to as the functional basis which basically is a common language to describe functional elements. The schematic of the product is somewhat similar to the function model. But where the function model describes the product using functional elements the schematics on the other hand can describe both functional and physical elements, whichever being the most meaningful. Having established an understanding of the functional structure of the product some methods base identification of modules on experience and some simple guidelines, i.e. a rather qualitative approach Pahl & Beitz (1996), Otto & Wood (2001), Ulrich & Eppinger (2000), [Pimmler & Eppinger, 1994]. Basically, these methods identify potential modules in a way similar to the way the MFD method makes use of the so-called module drivers.

Multi criteria assessment: Otto & Hölttä-Otto (2007) presents a technique based on multi-criteria assessment where platform concepts are given a score based on a set of different weighted criteria. Although, the method is designed to be used for screening of preliminary platform concepts, and not - as it is the focus of this research - analysis and re-design of product families, the method include analysis aspects that should be considered. The method is based on relatively quantitative metric adapted from the field of modularity, platform design, and product development in general (e.g. functional structure, DSM, commonality indices, etc.).

Value stream mapping: Most value stream mapping tools has a focus on information and physical goods passing through the supply chain. The value stream is consequently often perceived as the flow of materials through the value adding processes. There are several value stream mapping tools Womack & Jones (2003), Rother & Shook (1998). This section describes the “traditional” value stream mapping tool. Other tools or methods re describe in the subsequent sections. A less graphical depiction of the value stream is a process activity map. It is a schematic representation of the critical path of a production. It is basically a matrix containing a mapping between process steps and machines, time consumption and distance along with other factors of choice. This tool may be used in conjunction with the traditional value stream map or as a preparation of that.

Conclusion: It is clear that all the above approach can play a role in identifying an architecture for a product family or product assortment. The contribution of the Framework for Product Family Master Plan is mainly the relations between the different aspects. Most properties of a company and product assortment are so-called relational properties, Andreasen (1994). This means that e.g. production cost is the result of a meeting or relation between a product and a production system. The existing approaches main focuses on single aspects and not the interplay. Value stream mapping is widely used in lean projects. Often the product dimension is not taken into consideration. This is particularly relevant for companies manufacturing customized products and solutions. It is difficult to achieve a lean process on a complex product assortment with high number of part variants and unclear interfaces. Most of the data necessary for filling out the framework is often available in companies but is distributed across IT system, departments and persons. Making the relations visual is often very beneficial and makes it possible to make conclusions that are otherwise not possible.

5. INDUSTRIAL EXAMPLE: COOLING SYSTEMS

The case company is selling, designing, manufacturing and delivering customer specific cooling solutions for large OEM (Original Equipment Manufacturing) customers. The company has been grooving significantly over the last 10 years. One of the reasons is that their customers have grown significantly. While the customers have had a profitable growth, then the case company has experienced a declining EBIT over the last years.

The business situation for the case company can be characterized as follows:

- Increased pressure on prices –because customers becomes bigger and is thereby obtaining higher barging power
- Customers becomes global and is therefore expecting global delivery to an increased number of design and manufacturing sites
- The expectations on shorter lead time for quotations is expressed clearly by the customers
- Competition from EU and China is expected to be further intensified
- Shorter time to market for new products is necessary to maintain the current market share.

To address the above challenges many initiatives have been started – one of them is investigation of the potential benefits of a having a more clear architecture for the products and solutions.

Over a couple of month’s data for were collected and structured as described in the Framework for Product Family Master Plan. Due to confidentiality and practicality (10 A0 posters for describing the contents of the whole framework) the actual posters and contents are not shown in this paper. Among the important conclusions from utilizing the framework were:

- The dimensioning tool utilized in the sales phase is utilized to calculate the critical parts in the solutions. The consequence is that each solution will be unique. This will again lead to variety that is not value creating to the customers.

- Many classes of parts exist, e.g. tubing equipment for 4,1, 4,2 and 4,3 bars – this is also an example of non value creating variety.

- The product structure shows a complex mapping between the customer, engineering and part view. This means that each time a customer feature shall be different from the previous; it is a complex engineering task. In production it will lead to a new variant that have to be managed.

- A visit to a key customer revealed that a lot of the variety in terms of connections, fasteners, approvals etc was not very critical to the design of OEM equipment.

- There was significant variety in the manufacturing processes which is mainly adding complexity and will increase lead time.

- In many of the cooling systems the customer order decoupling point is placed very early in the production process chain which again will lead to long lead time and production complexity in general.

- There were in period's significant quality problems and related costs. One of the reasoning for this is that the amount of manufacturing processes made it difficult really to master them on world class level.

- Some of the projects is characterized by reinventing the wheel, meaning that the solutions already exists, but was designed again due to lack of overview concerning existing solutions.

The systematic mapping according to the framework was considered as being very beneficial both to the management team and senior personnel in sales, engineering and manufacturing.

The next step was to propose an architecture for the product assortment. The result of this design work is shown below, in figure 4. By critically looking at the framework for product family master plan an architecture with standardized interfaces were proposed at shown below.

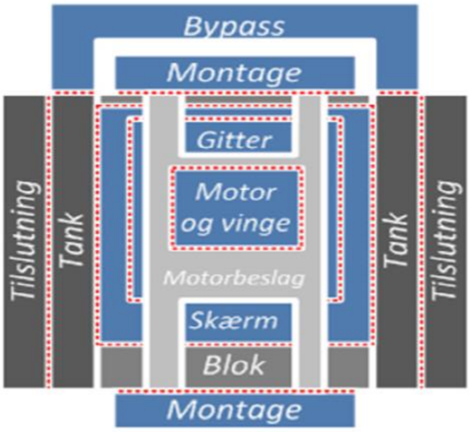


Figure 4: Proposed architecture for cooling solutions, in Danish

Among the main benefits estimated was:

- Reduction of number of parts from 600 to approximately 100
- Reduction of direct cost with 10%
- Reduction of complexity cost in the whole organization by 10%
- Significant reduction of lead time for new prototypes
- Significant reduction of lead time for new products

It was estimated that the new architecture will be able to cover 65% of the total sales, which means that that EBIT will be increased by 10% over a period of two years. At the moment a detailed design of critical components is carried out in order to verify the above benefits.

6. CONCLUSIONS

The main result from this work is the Framework for Product Family Master Plan. This framework can be utilized as a basis for systematic analysis of companies making customer or market variants of products. The framework enables a company to point out which of the activities and product elements that provide value to the customers. This serves as a valuable basis for identifying an architecture of a product assortment or product family. A good architecture enables a company to recognize profitable customers and projects. The case study indicates that a significant improvement of EBIT is possible with a fully implemented architecture. As the case study shows development of an architecture is not necessary a complex development project, but in many cases more a question of making decisions on preferred sub solutions.

A real implementation of the architecture is a very challenging task, because it has to work across sales, engineering and production. Many preconditions for a successful implementation exist, e.g. a focused market strategy, a modern IT infrastructure and a proper working product management. Due to the highly cross functional nature and market strategic impact the responsibility ultimately have to be anchored by the CEO and board of management.

7. REFERENCES

1. Andreasen, M. M., “Syntesemetoder på systemgrundlag – Bidrag til en konstruktionsteori” (in Danish), Doctoral thesis, Department of Machine Design, Lund Institute of Technology, Lund, Sweden, 1980.
2. Andreasen, M.M: Modelling - The Language of the Designer, Journal of Engineering Design, Vol. 5, No. 2, 1994.
3. Coad, P. & Yourdon, E., “Object-oriented design”, Yourdon Press Computing Series, Prentice Hall, Upper Saddle River, NJ, USA, 1998.
4. Ericsson, A. & Erixon, G., “Controlling design variants – Modular product platforms”, ASME Press, New York, NY, USA, 1999.
5. Fowler, T. C., “Value Analysis in Design”, Van Nostrand Reinhold Company, New York, NY, USA, 1990.
6. Harlou, U., “Developing product families based on product architectures – Contribution to a theory of product families”, Doctoral thesis, Department of Mechanical Engineering, DTU – Technical University of Denmark, Copenhagen, Denmark, 2006.
7. Hvam, L., Mortensen, N. H. & Riis, J., ”Product Customization”, Springer Verlag, Berlin, Germany, 2007.
8. Hubka, V. & Eder, W. E., “Theory of technical systems“, Springer Verlag, Berlin, Germany, 1988.
9. Hölltä–Otto, K. & De Weck, O., “Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints”, Concurrent Engineering: Research and Applications, Vol.15 Issue 2, pp. 113–126, Sage Publications, Thousand Oaks, CA, USA, 2007.
10. Kvist, M.: Product Family Assessment, Doctoral thesis (submitted to Technical University of Denmark), Department of Management Engineering, Technical University of Denmark, Copenhagen, Denmark, 2009
11. Mesihovic, S. & Malmqvist, J., ”A process-oriented approach for management of product configuration models”, Proceedings of the ASME 2004 International Design Engineering Technical Conferences, Salt Lake City, UT, USA, September 28 – October 2, 2004.
12. Meyer, M. H. & Lehnerd, A. P., “The power of product platforms – Building value and cost leadership”, The Free Press, New York, NY, USA, 1997.
13. Pahl, G. & Beitz, W., “Engineering Design – A systematic approach”, Springer Verlag, Berlin, Germany, 1996.
14. Pimmler, T. U. & Eppinger, S. D., “Integration analysis of product decompositions“, Proceedings of the ASME Design Theory and Methodology Conference, Minneapolis, MN, USA, September, 1994.
15. Rea, R. C., “Helping a client make up his mind - The use of a "decision tree" helps a client in planning his estate”, The Journal of Accountancy, pp. 39-42, May, 1965.
16. Rother, M. & Shook, J., “Learning to see: Value stream mapping to add value and eliminate MUDA”, Version 1.3, The Lean Enterprise Institute Inc., Cambridge, MA, USA, 1998.

17. Tiihonen, J. & Soininen, T., “Product configurators – Information system support for the configurable product”, Product Data Management Group, Helsinki University of Technology, Helsinki, Finland, 1997.

18. Ulrich, K. T. & Eppinger, S. D., “Product design and development”, 2nd edition, McGraw-Hill, Boston, MA, USA, 2000.

19. van Veen, E. A. & Wortmann, J. C., “Generic bills of material in assemble-to-order manufacturing“, International Journal of Production Research, 1st International Production Management Conference, Vol.25 Issue 11, pp. 1645-58, 1987.

20. Womack, J. P. & Jones, D. T., “Lean thinking – Banish waste and create wealth in your corporation”, Simon & Schuster UK Ltd., London, UK, 2003.

6.2 Paper B

PROACTIVE MODELING OF PRODUCT AND PRODUCTION ARCHITECTURES

Niels Henrik Mortensen, Christian Lindschou Hansen, Lars Hvam og Mogens Myrup Andreasen

ABSTRACT

This paper presents an operational model that allows description of market, products and production architectures. The main feature of this model is the ability to describe both structural and functional aspect of architectures. The structural aspect is an answer to the question: What constitutes the architecture, e.g. standard designs, design units and interfaces? The functional aspect is an answer to the question: What is the behaviour or the architecture, what is it able to do, i.e. which products at which performance levels can be derived from the architecture? Among the most important benefits of this model is the explicit ability to describe what the architecture is prepared for, and what it is not prepared for - concerning development of future derivative products. The model has been applied in large scale global product development projects. Among the most important benefits is contribution to:

- Improved preparedness for future launches, e.g. US versions of the products.
- Improved synchronization between product- and production development
- Achievement of attractive cost- and technical performance level on all products in the product family
- On time launch of the generation of the product program

Keywords: product architecture, modeling product architecture, multi product development, production architecture.

1 INTRODUCTION

Many industrial companies are facing serious challenges in maintaining competitive advantages. Among the most often mentioned challenges are:

- There is a need to reduce time to market (and more importantly time to money) for new products and solutions. Some of the companies that have participated in this research have lost 25% of market share in certain business areas during the last year. The reason for this is that they do not have the right products available on the market.
- There is a need to achieve right cost level for global products– Immelt et al. [1] mention that for GE to be cost competitive, the company needs products that are 80% cheaper in China compared to US products.
- The need for localization and customization of products are increasing [2].

There are certainly many approaches to handle the above challenges, which are of organizational-, process-, tool-, and competence nature. The focus in this paper is architectures, i.e. design of product families or product programs based on stable interfaces and standard designs (modules). Implementing an architecture have relations to all of the above aspects, but the overall hypothesis of the research presented in this paper is that in order to improve the design of product families, architectures have to be modeled explicitly and visually.

Many kinds of research projects have been carried out in order to improve the understanding of architecture work. Among the most important contributions are [3], [4] and [5]. So why is there a need for further investigations? One answer is that nearly all definitions of architectures are of structural nature, i.e. what the architecture *is*. This is for obvious reasons very relevant, but equally important are the functional aspects of architecture, i.e. what the architecture able *to do*. For instance the ability to answer the question: Which products can be derived from the architecture? This phenomenon is not very widely understood and described. Furthermore, the links between market, product and production/supply architecture are relevant. This is also not in itself a new recognition, but when it comes to e.g. evaluating the consequence of adding or removing a feature in a product, it is very difficult to model the consequences market- and production wise. It is the ambition to make a model

that allows operational linking between the three architectures. Almost none of the research literature explains explicit relations between market, product and production architectures.
The reason proactive is mentioned in the paper title, is to address that there is a big business potential and necessity for companies to think ahead in product program design, meaning that the next 2, 3 or 4 launches of derivative products have to be taken into consideration explicitly. Architecture wise this means that an architecture shall be able to show the *preparedness* for the launching of future product generations.
The results presented in this paper is based on research in 3 PhD projects, Kvist [6], Harlou [7], and Pedersen [8] within modeling of architectures. The structure of the paper is as follows. Section two will report on some of the findings from observation of architecture work in main Scandinavian companies. Section 3 will identify the relevant modeling aspects to be included for modeling architectures. Section 4 will present state of the art concerning modeling of architectures. After that section 5 will present a proposal for how to model market, product, and production architectures.

2 WHY IS THERE A NEED TO IMPROVE ARCHITECTURE MODELING?

If a product assortment in a company is described by means of a traditional market matrix, it can be shown as below in Figure 1

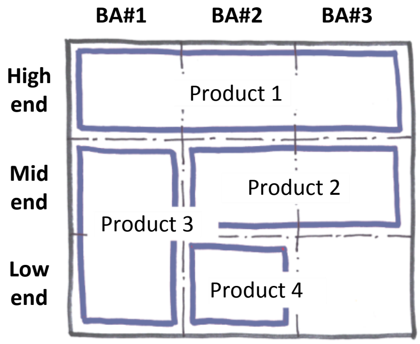


Figure 1. Product mix of a company:
The horizontal axis shows the Business Areas (BA) and the vertical axis describes the performance level of products ranging from low to high performance products

Because many products are designed without conscious decisions concerning the market coverage, poor product family design is carried out. Some of the bad decisions that we have observed in this research are:
One size fits all: In many companies the product architecture is shared from high end to low end products. One consequence of this is that low end products have too high costs and high end products are not sufficiently prepared for future launches. In some companies there is a conception that “stripping” the high end products is a way of developing low end products. There are perhaps examples where this can be done, but is in many cases it is not possible. In other words, “stripping a Rolls Royce does not bring a Volkswagen into existence”.
Dedicated products – future generation products are not addressed: Product or product programs are designed without sufficiently addressing facelifts and next generations. Some examples of this are e.g. US variants developed on European development sites. The consequence is that US product variants are significantly delayed. Another company is developing a dedicated product for hospitals. This product shall at a later stage also be used in large industrial laboratories. The consequence of developing a dedicated hospital product is that the industrial product program is delayed at least 3 years.
Spaghetti products: Some product families consist of subsystems with very complex interfaces and interactions. The consequence is that development of even small updates becomes very complicated and resource intensive.
Non value adding variety: There are many examples of variety in a product assortment that does not provide value to customers but only adds complexity cost in companies. A few examples of this phenomenon are: One company is delivering products with actuators that are bolted, welded and glued. This means, that three types of production processes have to mastered, leading to increased cost. Seen from a customer point of view, this variety does not add value. Another company is having

pressure tanks certified for 4.1, 4.4 and 5.0 bar. In this case certificates and approvals have to be developed and maintained without adding any extra value to the customers.
The consequences of the above issues are higher costs and reduced ability to launch new products. One of the means to handle the above issues is to develop product families based on explicit architectures. The next section will take a closer look on which phenomena to include in the modeling of architectures.

3 WHICH ELEMENTS SHOULD BE INCLUDED IN MODELING OF ARCHITECTURES

The paper is based on the so-called Product Family Master Plan Framework [7], [9], Theory of Technical Systems [10] and Theory of Domains [11]. Consequently three types of architectures are necessary, i.e. market, product and production/supply chain.

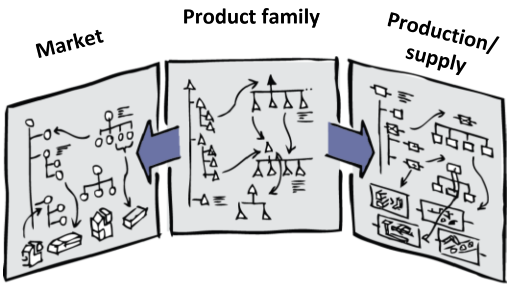


Figure 2. The Product Family Master Plan Framework (by Kvist (2009) based on Harlou (2006))

There exist many definitions of architectures in literature. Some of the most often quoted are:
The combination of subsystems and interfaces defines the architecture of any single product. Every product has an architecture; the goal is to make that architecture common across many products. Any single product's architecture therefore has the potential to become a product platform architecture if it is designed and then used as the basis for creating several more derivative products", Meyer & Lehnard [5].
An architecture is a structural description of a product assortment, a product family or a product. The architecture is constituted by standard designs and/or design units. The architecture includes interfaces among units and interfaces to the surroundings", Harlou [7].
"In essence, a PFA (Product Family Architecture) means the underlying architecture of a firm's product platform, within which various product variants can be derived from basic product designs to satisfy a spectrum of customer needs related to various market niches", Jiao & Tseng [12].
All of the above definitions are underlying the importance of interfaces and description of how product families can be described. This is certainly very important, but the above definition is missing the clear distinction between structural and functional aspects of an architecture. Furthermore it does not explain the type of elements that are relevant in the structural and functional definitions. In accordance with Theory of Technical Systems [10] this research will reserve the word structure to how individual products are built up and architecture will be reserved for describing how a product family is built up including the future derivative products.
The next sections will explain some of the necessary architecture modeling requirements in market, product and production architectures that this research have identified.

3.1 Market architecture requirements

The overall purpose of the market architecture is to model what the product family shall cover and what it shall not cover. Often this is unclear leading to unfocused product architecture design.
Product properties across the product program: Taking a starting point in properties being obligatory, expected or positioning in the market place, properties can be realized by implementing them as either e.g. basic properties, differentiators or delighters in the product design – depending on the level of fulfillment.
Requirements across individual and all application areas: This is important in order to scope the product families, e.g. which areas shall be covered and which shall not be covered. Similarity and

differences across application areas is in principle going to drive variety of the elements constituting the product architecture and flexibility of the production architecture.

Product family architecture definitions: This dimension is explaining which product families that shall be developed and how they cover the market grid as previously shown in e.g. Figure 1.

List of features and options: This is an important area since it is often difficult for projects to clarify how many features shall be implemented in high end, medium and low end products. It is relevant to explicitly specify which features shall be implemented and which ones shall be postponed to later launches or simply omitted from certain market segments.

List of commercial variants that shall be launched to the market: This list describes the complete list of individual products and which standard designs and features that goes into each product. This is relevant in order to identify the total development task that shall be handled.

3.2 Product architecture requirements

List of structural elements: According to [7] we distinguish between standard designs and design units. The standard designs encapsulate what is reused in several product families, whereas the design units are elements which are not reused. The distinction between standard designs and design units is of importance as their nature is different. Standard designs have to be designed in such a way that they can be used in future products, whereas design units only have the scope of one product. Consequently the application aspects are different for standard designs and design units. A standard design requires a higher degree of documentation, higher degree of maintenance, appointment of responsibility than a design unit, in order to enable ruse in future products.

List of interfaces: This area described the important mechanical, electrical, fluidal and software interfaces between standard designs and design units.

List of product families that can be derived from the architecture: This area describes the functional aspect of an architecture and includes key properties of the individual products that can be developed e.g. cost, energy efficiency, footprint, fault tolerance etc.

3.3 Production/supply chain requirements

Generic production flows: These flows describe the main production and assembly processes including the necessary production equipment. At the end the types of standard designs that can be produced shall be described. This indicates the flexibility of the production and shows what differentiates each variant and what is common.

List of equipment: This includes the production lines, cells, machinery, tools and fixtures, mapped towards future launches.

3.4 Road mapping – future launches

Future launches: Indicate which products and standard designs to be launched.

Specific product updates: This shall explain which products that shall be launched for each application area.

4 STATE OF THE ART

This section described significant contributions to the modeling of architectures in literature:

Modular Function Deployment: The modular function deployment (MFD) [3] builds largely on the methodology of the QFD method and on the formulation of eight so-called module drivers. The purpose of MFD is to enable cross functional teams (including mainly marketing, development and production personnel) to create a mapping from the physical structure of the products within a family to the functional structure of those products and to ensure that the functional structure corresponds to the demands of the customers. Modular Function Deployment method consists of five consecutive steps. Customer requirements are mapped to functional criteria and subsystem design characteristics and subsequently forming a physical design in which a modular architecture supports a carefully selected set of modularization incentives called module drivers.

Design Structure Matrix: This approach takes a starting point in the decomposition of a product into components/systems and an identification of interfaces/relations among these, Pimmler & Eppinger [13], Höltta-Otto & De Weck [14]. By the use of algorithms, it is possible to encapsulate components into modules or chunks that are closely related to each other from an interaction point of view [15].

This process is referred to as clustering. The outcome of a DSM is a proposal for a future modular product architecture.

Generic Bill of Materials: The generic BOM originate from the assemble-to-order environment [16]. The end-products typically have a number of features for which a number of options are available to choose from. Not many options are required in order to make the number of combinations (i.e. end-products) enormous. The number of end-products can easily become too large to able to define specific BOM's for every single combination. Furthermore, forecasting, BOM-storage and maintenance become unmanageable. The generic BOM is a concept that is introduced to enable creation of a specific manufacturing BOM when the customer places an order. The generic BOM is used to describe related products in one all-embracing model by using generic and specific items.

Decision tree: The decision tree [17] is used by Tiihonen & Soininen [18] as a product configuration model, which basically represents all the valid combinations of the components that can be used to obtain the desired functions for the customer. The decision tree presents the multitude of component variety within a product family and by the use of positive combinatory relationships (e.g. if "engine size"=D13 then "engine power" must be 360 or 420 hp) and/or incompatibility relations (e.g. if "engine size"=D13 then "engine power" cannot be 220 or 700 hp) it defines the possible product configurations.

Value analysis: Value Analysis is a discipline founded at General Electric in the late 1940's [19]. In short, value analysis is a methodology that has as its purpose to relate cost with functions in a product. It is a stepwise methodology in which a product is partitioned into smaller constituents for further analysis – that may be analysis of cost or value. Value is not the same as the Japanese idea of customer value we may see within the lean paradigm. Value is specifically defined as the "worth" relative to cost, i.e. value = worth/cost. Worth in this sense actually resembles the idea of customer value in lean very well. It is a denominator of those aspects, functions and features a customer wants to pay extra for. Some practitioners try to quantify worth and relate it directly to cost. Obviously cost is rather quantitative and measurable in hard currency, while "worth" is a more soft and qualitative size. Whether qualitative or quantitative, value has a focus on identifying value elements from a customer perspective and relate it directly to the functions of the product and thereby indirectly to the way the products are built.

Function structures: The function-based design methods are characterized by establishing either a function model [20] or the schematics of the product [4]. The function structure describes the flow of material, data, and energy through sub-functions of the product using a set of rules (e.g. the rules that are referred to as the functional basis which basically is a common language to describe functional elements. The schematic of the product is somewhat similar to the function model. But where the function model describes the product using functional elements the schematics on the other hand can describe both functional and physical elements, whichever being the most meaningful. Having established an understanding of the functional structure of the product some methods base identification of modules on experience and some simple guidelines, i.e. a rather qualitative approach [4], [13] and [20].Basically, these methods identify potential modules in a way similar to the way the MFD method makes use of the so-called module drivers.

Multi criteria assessment: Otto & Hölttä-Otto [21] presents a technique based on multi-criteria assessment where product architecture concepts are given a score based on a set of different weighted criteria. Although, the method is designed to be used for screening of preliminary product architecture concepts, and not - as it is the focus of this research - analysis and re-design of product families, the method include analysis aspects that should be considered. The method is based on relatively quantitative metricc adapted from the field of modularity, product architecture design, and product development in general (e.g. functional structure, DSM, commonality indices, etc.).

Value stream mapping: Most value stream mapping tools has a focus on information and physical goods passing through the supply chain. The value stream is consequently often perceived as the flow of materials through the value adding processes. There are several value stream mapping tools, e.g. by Womack & Jones [22]. This section describes the "traditional" value stream mapping tool. Other tools or methods re describe in the subsequent sections. A less graphical depiction of the value stream is a process activity map. It is a schematic representation of the critical path of a production. It is basically a matrix containing a mapping between process steps and machines, time consumption and distance along with other factors of choice. This tool may be used in conjunction with the traditional value stream map or as a preparation of that.

Conclusion: It is clear that all the above approaches can play a role in identifying structural aspects of an architecture, but the functional aspects are not explicitly described. Functional aspects of an architecture includes identification of which products that can be derived and the properties of these products, e.g. costs or energy efficiency. This is important because identification of what the architecture can support and what it cannot support concerning future launches of products is of high importance. Furthermore the structural contents of architectures are not described in terms of module or design types. This topic is relevant in order to design flexibility in product architectures. In large projects this plays an important role concerning scoping of the development task. Also visualization of multiple architectures in the market, product and production is missing.
The next section will present a proposal for the modeling of market, production and production architectures.

5 ARCHITECTURE MODELING

5.1 Market architecture

The purpose of modeling the market architecture is to bring clarity into decision making concerning the choice of which segments to cover or not cover and what properties are needed in order to do so across different business areas with different applications. A clearly defined market architecture is able to guide and control the engineering efforts towards profitability by "smart" product family design.

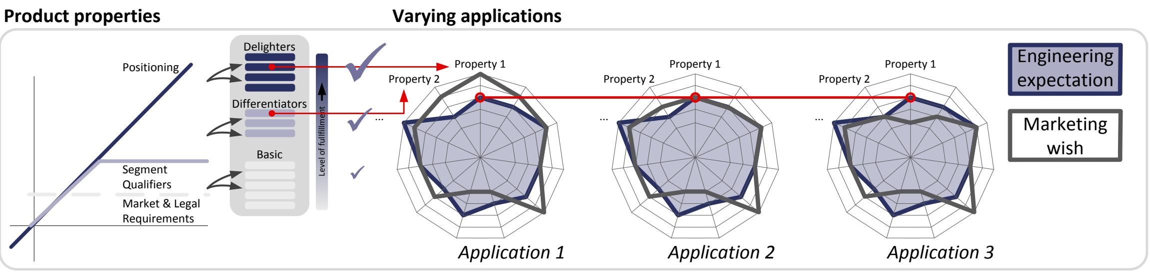


Figure 3. Product properties and their mapping towards varying market applications.

The radar diagrams show the total performance of the product by mapping the properties capable of positioning the product against competitors e.g. by differentiation. During the early phases of product scoping and requirements definition in close cooperation with competencies representing marketing, the mapping can serve as means of matching the wanted product performance from a marketing point of view with the expected product performance from an engineering point of view. Hereby, the explicit mapping can have a brokering function facilitating the meeting between sometimes unrealistic marketing wishes and best guess engineering expectations. If applied to a product family intended to cover different applications in different segments with varying requirements, it is of fundamental interest to map marketing professionals' perception of the spectrum of varying demands. As it is most often impossible to fulfill requirements for all segments, the mapping can help focusing the product architecture towards the most appropriate and favorable segments. To concretize the product properties, features and options can be modeled e.g. by the means of the "customer view" [7] mapped towards the different applications and varying the performance levels (low-, mid- and high end).

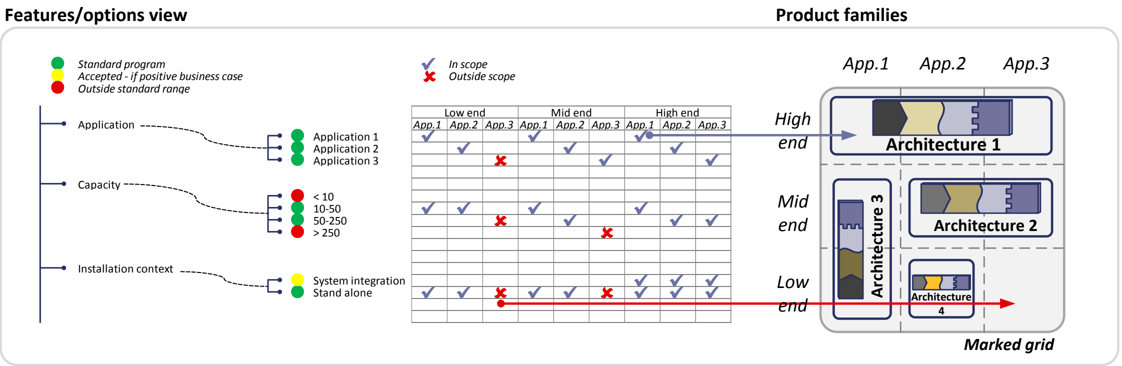


Figure 4. Features/options and their mapping towards performance levels in different applications and the identified product architecture(s).

This mapping serves to answer the questions of which product features that are in scope for the development task. Some features are too expensive or simply irrelevant for certain applications and are outside scope. Other features will be outside the standard program for all applications since they may ‘pollute’ a robust product architecture. Finally, the mapping towards one or more product architectures closes the gap towards engineering and sets the boundaries for the development task.
The detailing of the link between matching product features and identified product architecture, calls for a visualization of the commercial variants. They serve as being the ‘contract’ between engineering and marketing explicitly identifying the development task. The detailing of this list requires the product development task to be past the early stages, but major value is represented in conducting this modeling as early as possible.

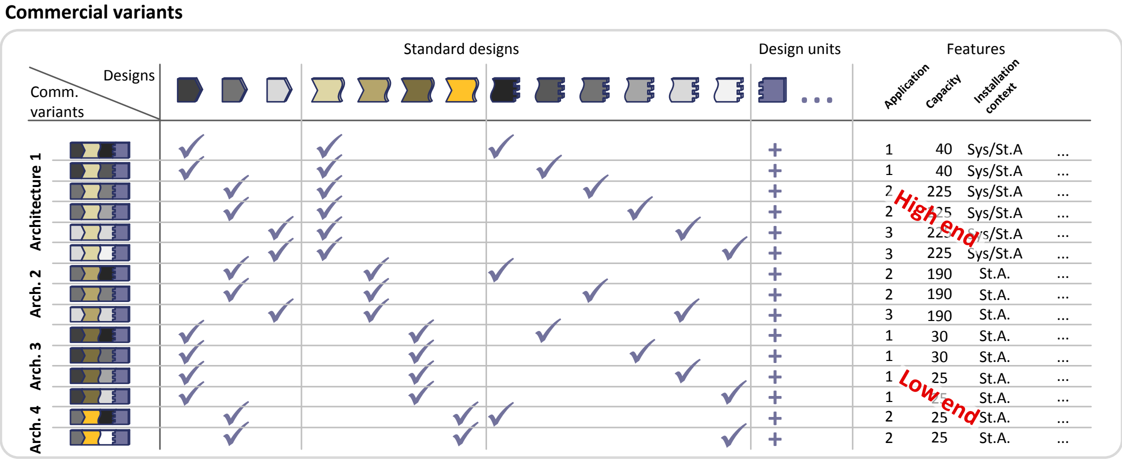


Figure 5. Commercial variants and their utilization of standard design, design units and associated product features

The modeling will vary according to the application variation, general market aspects etc., however, the models shown in Figure 3, Figure 4 and Figure 5 are made to illustrate general purposes.

5.2 Product architecture

According to the suggestions presented in this article, the modeling of product architectures encompasses the constitutive structural elements of a product architecture and the behavioral functional abilities. In other words, the aim of these modeling techniques are not solely to describe what the product architecture *is*, but also what the product architecture is able *to do*.

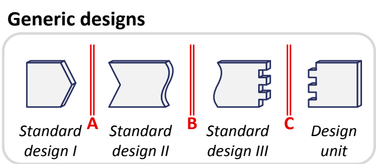


Figure 6. Generic structural elements of the product architecture: Standard designs and design units.

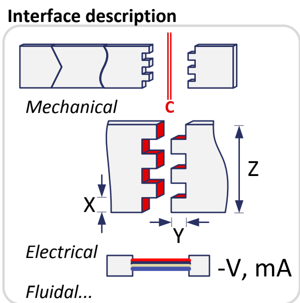


Figure 7. Modeling of interfaces between standard designs, between standard designs and design units and/or surroundings.

Equally important to the standard designs and design units, the interfaces capable of maintaining a predictable product structure, must be modeled explicitly as well.
Different standard designs can adopt different roles. Some are closely related to specific functions and/or application, while others are universal to the product architecture. Finally, design units are used for embodying functionalities that vary between individual product variants.

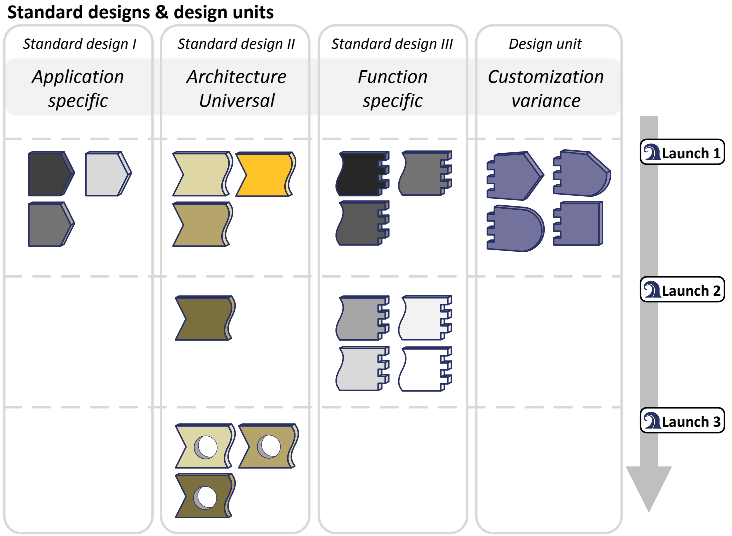


Figure 8. Standard designs & design units.

Figure 8 shows the variance across the different structural elements while incorporating the dimension of future launches: Which designs need to be prepared for which launches? Naturally, it is impossible to plan further than a certain realistic extent in rapidly changing markets, but the higher the detail this modeling can achieve, the better the basis for improving the launch preparedness is.

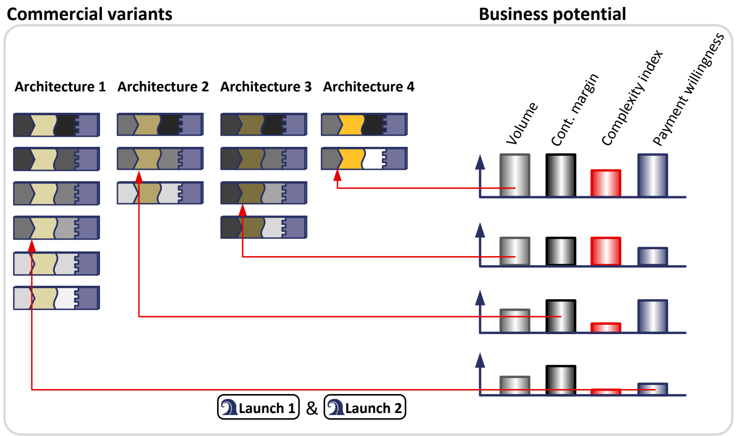


Figure 9. Assessing the business potential of commercial variants.

As described earlier, the explicit modeling of commercial variants early in the development process, act as the explicit link between the market- and product architectures, it is of fundamental commercial importance to map the expected production volumes, contribution margins, and payment willingness from customers – the payment willingness being the quantitative interpretation of the 'worth' phenomenon described earlier. If established, a measure of the complexity induced by different product variants can be included to qualify discussions with industrialization professionals with the task of freezing production architecture aspects. These four measures can help balancing out the product architecture(s), ensuring a leveled variance spectrum composed of "smart" variants with an appealing overall business justification.

5.3 Production architecture

Depending on the size of the product architecture development project, the associated production system will need either an update, a modification or a complete redesign. The production system is designed coherently, as the product architecture matures and passes from concept to detail design.

As basis for the modeling of production architecture is the generic production flow shown in Figure 10. This is capable of showing how and when the product variants are created in the production lines, which elements in the production system that are alike and which elements that differ. The relevant decoupling points (either *variant creation points* or *customer order points*, depending on the context*)* can be established and fixed [xx]+[xx]. Furthermore, an inclusion of relevant machinery, tools and capacity utilization metrics provides the opportunity of assessing key financial characteristics of the suggested setup.

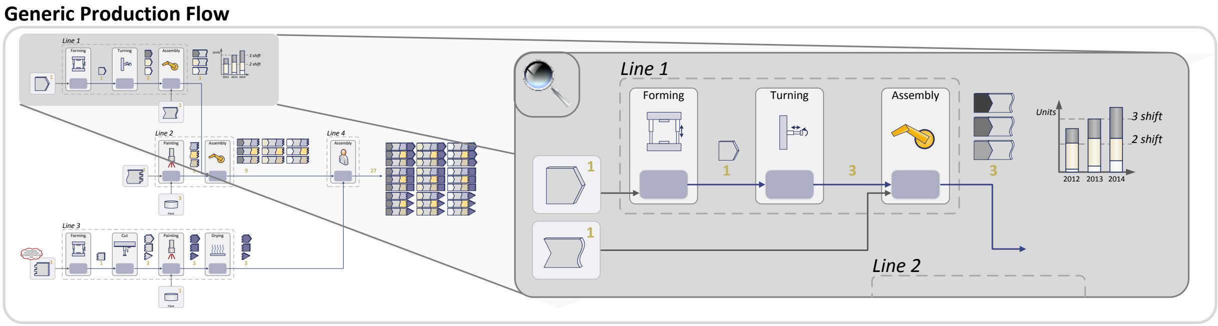


Figure 10. Generic Production Flow: Modeling the flow of all variants in one visual model

Since production equipment can require extensive capital investments, a mini roadmap of the lines, machinery and tools is valuable to map towards the suggested launch rhythm. As shown in Figure 11, the addition of further parts and components intended for launch 3 and beyond, will most likely entail a larger utilization of the production capacity, take up physical space of the production floor and require additional investments in machinery and tools downstream.

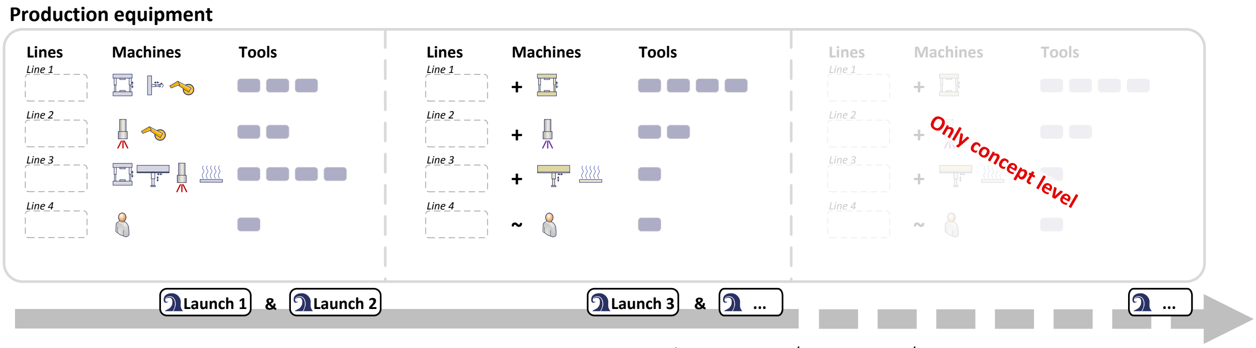


Figure 11. Production equipment needed for 1st launch, 2nd launch, 3rd launch etc.

These are all aspects that are predisposed by the design of the product architecture(s); thus requiring explicit and coherent models.

As marked in Figure 4, certain features will be part of the standard program incorporated in specific commercial variants, while other features will need an individual business case in order to be fulfilled as e.g. customizations. Setting up a global chain of supply and delivery, service levels of standard lead times, degree of local customization possible etc., are also factors predisposed by the architectures of the product and production. Figure 12 shows an example of how a global company could utilize the price of cheap labor in some regions with the local capability of customizing product (and perhaps conduct final assembly) around the world in product/distribution centres.

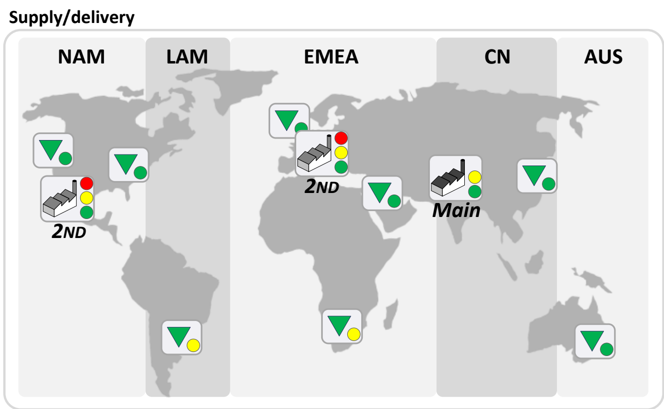


Figure 12. Global supply and delivery capabilities

5.4 Roadmap

The behavioral aspects of the market-, product- and production architecture is considered in the architectures' future launch preparedness. This is a function of the architecture, explaining what the architecture is able *to do*. This ability is modeled by visualizing the launches, derivative products and specific product updates – already planned for.

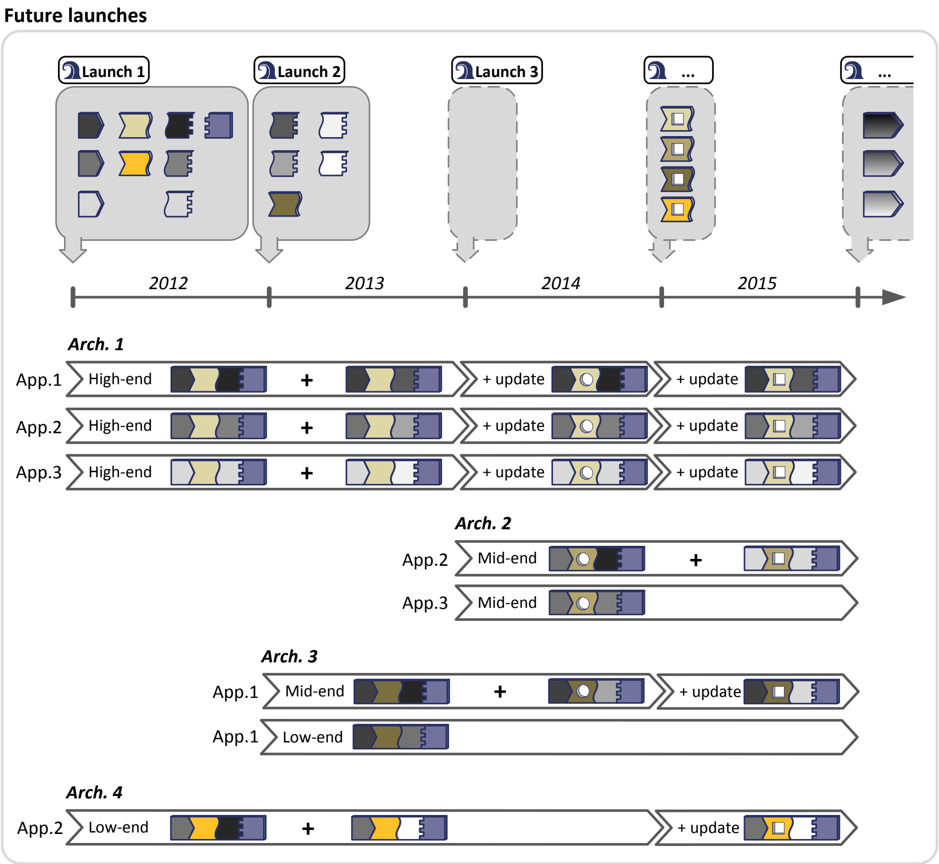


Figure 13. Future launches:
Launch preparedness, launch waves, derivative products and specific product updates.

6 EXPERIENCE FROM APPLICATION

The above architecture modeling approach has been utilized in one large scale product development project. The case company is operating globally and develops industrial products in high volumes. During the 2 year project, approximately 100 designers have been working on developing the product program. The development project has included complex fluid-dynamics, mechanics, materials, software, electronics, solid state mechanics and thermodynamics - architecture wise only mechanics and electronics have been included. In the area of production, complete new facilities have been established in Asia, Europe and the US. Market wise the product program has partly been launched. Sales is taking place through existing sales companies. The application areas include mainly existing well known areas, but also a few new applications are included, e.g. renewable energy. So far the research conclusions have not yet been approved by the management of the company and therefore the case is presented anonymously.

The PhD students have been working for more than a year and a half, several days a week in utilizing the market, product and production architecture model. During the case study, the architectures has been developed and described by the PhD students in close collaboration with employees in sales, product management, engineering, production and supply chain. Four types of architects have been responsible for the contents of the market, product and production/supply chain architecture. The PhD students have carried out the practical structuring of information. The four types of architects are named market, product, production/supply and cross functional architect. The market architect is based in product management and is responsible for the market architecture and roadmap; the product architect is based in engineering and is responsible for the product architecture; the production architect is based in production and is responsible for the contents of the production architecture; and the cross functional architect is responsible for the alignment of the market, product, production architecture and roadmap. The project manager has acted as the cross functional architect.

Two kinds of meetings have been conducted in the project: They are named architect meeting and cross functional architect meetings. The first year both architect and cross functional architect meeting was carried out each Thursday from 9.00 to 11.00. During the last period the market, product and production architect was held each week, but cross functional architect meeting was held every 2nd week. Participants in market architect meetings were program management and product management. In the product architecture meetings, senior designers from relevant specialist areas participated. In the production architect meetings new product introduction managers, tool designers and production line designers participated.

Experience from application of the market architecture: The feature/options has enabled an earlier and more explicit definition of what defines a high end, mid end and low end product, i.e. clarification of which features and options that shall go into which variants.

Experience from application of the product architecture: The interfaces have been decided much more conscious compared to previous projects in the company. It means that the next 3 product launches have been explicitly planned in such a way that the architecture is prepared for one new technology, an update of the user interface and more advanced wireless communication.

Experience from application of the production architecture: The project has had the task to establish completely new production lines with three kinds of automation levels, fully automatic, semi automatic and manual production and assembly. Particularly the full automatic production line design have benefitted from the product architecture. It has been possible to order new production and assembly equipment earlier since the product program have been decided earlier and therefore variety of each part have been known earlier. Also the product architecture specification has been beneficial to production design since flexibility and scalability is very important design properties for automatic production equipment.

Experience from cross functional application of the market, product and production architecture: The main benefit of the cross functional review meetings have been continuous scoping of the project, i.e. decisions concerning what shall be developed now and what shall be postponed. Another aspects that have been more consciously considered, is clarification of where the architecture shall be prepared and where is shall not be prepared for future launches. The performance limits concerning cost, energy, foot print and availability have also been clearly defined.

All in all the main benefits of applying the explicit modeling of market-, product-, production architectures (including the roadmap) has been a contribution to:

- Improved preparedness for future launches, e.g. US versions of the products.
- Improved synchronization between product- and production development
- Achievement of attractive cost- and technical performance level on all products in the product family
- On time launch of the generation of the product program

Concerning future application the cross functional architect role has to be reconsidered. With a traditional organisation, one could argue that “no one” or everyone is responsible. No single person or department have all the competencies necessary to handle the cross functional architect role. This will be a topic for further research and case studies. The architecture models are mainly handling technical decisions whereas business decisions is only implicitly addressed. This is another area that obviously should be improved.

7 CONCLUSIONS

The paper has presented an explicit proposal for description of contents of a market, product and production architecture. The main contribution is the distinction between structural and functional contents of architectures. By this distinction it is possible to improve the description of what the architecture is prepared for concerning future launches.

Further work includes test in two other companies. So far only the mechanical is included. It is clear that also software has to be included in the next version of the modeling method. Also other life phases such as service/aftermarket will in many cases be of high importance. A follow up case study is planned in order to study whether the intended preparedness is realized in reality.

REFERENCES

[1] Immelt, J.R, Govindarajan, V. and Trimble, C.: How GE Is Disrupting Itself, *Harvard Business Review*, 2009, 87(10), 56-65

[2] Hvam, L., Mortensen, N. H. and Riis, J., *Product Customization*, 2007 (Springer, Berlin)

[3] Ericsson, A. and Erixon, G., *Controlling design variants – Modular product platforms*, 1999 (Society of Manufacturing Engineers, Michigan, US)

[4] Ulrich, K. T. and Eppinger, S. D., *Product design and development*, 2nd ed., 2000 (McGraw-Hill, Boston, MA, USA)

[5] Meyer, M. H. & Lehnerd, A. P., *The power of product platforms – Building value and cost leadership*, 1997 (The Free Press, New York)

[6] Kvist, M., *Product Family Assessment*, 2009, Dissertation/Thesis (DTU Management, The Technical University of Denmark, Kgs. Lyngby).

[7] Harlou, U., *Developing product families based on architectures - contribution to a theory of product families*, 2006, Dissertation/Thesis (Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby).

[8] Pedersen, R., *Product Platform Modeling*, 2009, Dissertation/Thesis (DTU Management, The Technical University of Denmark, Kgs. Lyngby).

[9] Mortensen, N. H.,Hvam, L.,Haug, A.,Boelskifte, P. and Hansen, C. L. Making Product Customization Profitable, *International journal of industrial engineering*, 2010, 17(1), 25-35.

[10] Hubka, V. and Eder, W.E, *Theory of Technical Systems*, 1988 (Springer, Berlin)

[11] Andreasen, M. M., *Syntesemetoder på systemgrundlag – Bidrag til en konstruktionsteori* (in Danish), 1980, Dissertation/Thesis (Department of Machine Design, Lund Institute of Technology, Lund, Sweden)

[12] Jiao, J., & Tseng, M. M. A methodology of developing product family architecture for mass customization, *Journal of Intelligent Manufacturing*, 1999, 10(1), 3–20.

[13] Pimmler, T. U. & Eppinger, S. D., Integration analysis of product decompositions, *Proceedings of the ASME Design Theory and Methodology Conference*, 1994 (Minneapolis, MN, USA)

[14] Hölltä–Otto, K. & De Weck, O., Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints, *Concurrent Engineering: Research and Applications*, 2007 15(2), 113–126 (Sage Publications, Thousand Oaks, CA, USA)

[15] Steward, D. V., The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management*, 1981

[16] van Veen, E. A. & Wortmann, J. C., Generic bills of material in assemble-to-order manufacturing, *International Journal of Production Research*, 1st International Production Management, Conference, 1987, 25(11), 1645-58

[17] Rea, R. C., Helping a client make up his mind - The use of a "decision tree" helps a client in planning his estate, *The Journal of Accountancy*, 1965, 39-42

[18] Tiihonen, J. & Soininen, T., *Product configurators – Information system support for the configurable product*, 1997 (Product Data Management Group, Helsinki University of Technology, Helsinki, Finland)

[19] Fowler, T. C., *Value Analysis in Design*, 1990 (Van Nostrand Reinhold Company, New York)

[20] Pahl, G. & Beitz, W., *Engineering Design – A systematic approach*, 1996 (Springer, Berlin)

[21] Otto, K. and Hölttä-Otto, K., A multi-criteria assessment tool for screening preliminary product platform concepts, *Journal of Intelligent Manufacturing*, 2007, 18(1),59-75

[22] Womack, J. P. and Jones, D. T., *Lean thinking – Banish waste and create wealth in your corporation*, 2003 (Simon & Schuster UK Ltd., London)

6.3 Paper C

INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

ON THE MARKET ASPECT OF PRODUCT PROGRAM DESIGN: TOWARDS A DEFINITION OF AN ARCHITECTURE OF THE MARKET

Christian Lindschou Hansen, Niels Henrik Mortensen and Lars Hvam

Keywords: market architecture, product architecture, product program design, product family, platform

Abstract

An often overlooked aspect of architecture based product development, is the market aspect. However, without focusing the scope of the product family and ensuring an appropriate layout of product families, variants and features across the product program offerings, experiences show that architecture based product families become rigid, unfocused, prepared for yesterday's market situation, and ultimately lack profitability. This paper will propose to expand the existing notion of coordinating product and production architectures as a means to develop profitable architectures by including an architecture of the market. This is to be interpreted as the 'market perspective' of the product family referring to the design of the product family from the market's point of view. The main result of this paper is the suggestion of a definition of a market architecture with an articulation of its elements, relations, hierarchical nature and raison d'être. Three action research studies show that defining the market architecture serve as a feasible and operational means of addressing the market aspects in architecture development.

1. Introduction

Architecture based product development can basically be considered as a means of solving the conflicting task of providing variety to the market place while seeking to reduce complexity among internal company operations in order to achieve an attractive cost level of a product family. Commonality of activities is here an important ingredient, which is closely related to the commonality of the structural aspects of the product family. However, as there is a very close relationship between the variety provided and the dispositioning of costs during development, it is a fundamental challenge to maximize the variety that generates a high payment willingness without sacrificing internal complexity, and minimize the variety that does not generate any payment willingness. These two standard situations are usually not too difficult to differentiate from each other. On the other hand, the foundation of good decision-making in reality is often much more blurry to reach such unambiguous conclusions.

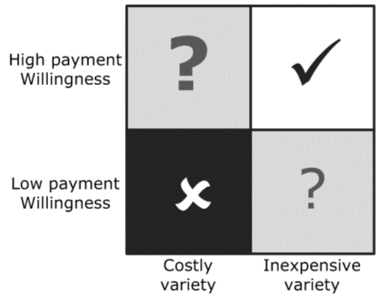


Figure 1: The four standard situations

As seen in Figure 1 the dilemma begins especially when we look upon the situation of high payment willingness with costly variety. The provision of support for decision-making in these 'grey zones' is the theme of the paper, and the contribution offers an operational suggestion for how to improve the foundation of decision-making to handle the trade-offs that arises from this dilemma.
Behind the scenes of these types of decisions are the balancing of the offerings to the market towards the design of the architecture of the product program and the production setup.
In order to account for the hierarchical relations between the meetings encountered by a product family through its life cycle phases, structures can be defined for every life cycle phases, which are to be taken into account during development [Andreasen et al. 1996].

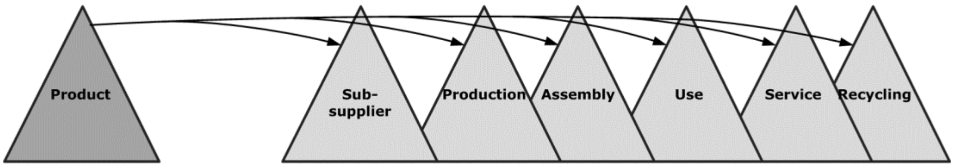


Figure 2: Structure of the product life cycle phases
(redrawn from Andreasen et al., 1996)

From Andreasen et al. [2004] one definition of an architecture is that it is a "*purposefully aligned structure of a system"*. Hence, the deliberate alignment of the structures of the life cycle phases may denominated as architectures. Yet, the architectures vary a lot depending on what life stage is under consideration, ranging from architectures mainly constituted of structural elements (e.g. production) to architectures mainly constituted of behavioral character (e.g. service).

While the production stage usually carries most of the costs, the product and production architectures are previously proposed to be developed in coordination with each other [Mortensen et al. 2011]. However, as an extension to this, product and production architectures do not become profitable if the derived product family is not targeted the market in a coherent and appropriate way. Therefore, we propose the concept of an architecture of the market, as a systematic description of the hierarchical aspects that define the meeting between the product family and the launch on the market [Mortensen et al. 2008]. As with product and production architectures, the key challenge here is to create an optimal *fit* between the market, product and production architectures, which is done through *alignment* [Andreasen et al. 2004]. The three domains of market, product and production follow the classic partitioning from *Integrated Product Development* [Andreasen and Hein 1987].

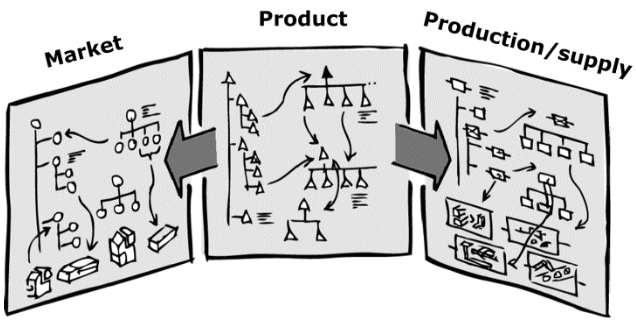


Figure 3: Three architectures: Market, product and production/supply
(redrawn by Kvist, 2009, originally from Harlou, 2006)

The classic paradox of variety and commonality is largely at stake here [Andreasen et al. 2004]. While variety can be considered a relational property between product families, variants and features, commonality is a relational property between different life phase views, referring to the resemblance of the meetings encountered by the products during these phases [Andreasen and Olesen 1990]. Hence, the deliberate balancing of variety and commonality calls for a modeling of the critical aspects deciding the outcome of the trade-offs between these paradoxical goals. This paper suggests to expand the current notion of coordinating the development of product and production architectures as a means to develop profitable architectures by including an architecture of the market.

The paper will briefly describe the motivation for such a market architecture and discuss the requirements for this. Subsequently the literature is screened for the state-of-the-art, followed by a proposition towards the definition of an architecture of the market. Lastly, the experiences from applying the framework are commented upon and conclusions are drawn.

2. Why do we need an architecture of the market?

2.1. The classic pitfalls

From a company perspective, we need an architecture of the market, in order to avoid these classic pitfalls:

2.1.1. Market cannibalization

This is the phenomenon of new product introductions becoming unprofitable, due to significantly overlapping market coverage between product families. This results in lost sales of existing product families that does not justify the new introduction.

2.1.2. The 'sandwich' phenomenon

This phenomenon describes the 'trapped' situation encountered by companies in industries where growth is centered in the high-end and low-end market tiers, and the product families developed lack performance to compete in the high-end markets and lack cost competitiveness to compete in the low-end markets. This leaves them unfocused in the mid-end tier with decreasing sales.

2.1.3. Dead end scaling strategies

A dead end scaling strategy is characterized by the company having no profitable scaling strategy in place, thus using the development efforts on new product introductions without prospects for follow-up releases, upgrades or continuous multi-launches. A dead end is encountered when no natural continuation is planned.

2.1.4. Uneven mix of product properties

If there is no clear differentiation or distinction between which product properties the market expects to be in the product, and which product properties that is capable of positioning the products in the targeted segment, a common result is an uneven mix of product properties eventually leading to over and underspecified product variants, which drive internal complexity and lack profitability.

2.1.5. Lack of application knowledge

As the variance across product applications can be considered the basic variant driver of a product family, the lack of structuring of knowledge about the requirements from these applications, can distort the focus of which application to target as primary and which application to deprioritize as secondary.

2.1.6. Sub-optimal price and cost points

Even though price and cost in principle always should be separated, the layout of optimal and suboptimal cost-points of the product program should be aligned with price points suggested from a marketing perspective. The pitfall here is to end up with high volume variants positioned in suboptimal price and cost points that do not fulfill the product program target contribution margins, thus jeopardizing profitability of the whole product family.

2.1.7. The jungle of free-text requirements

Many companies have improved their management of requirements, but the classic pitfall here is the lack of classification of requirements that appear in free-text fields with no clear sender or recipient, no differentiation between need/nice to have, and no links to the product architecture. Also, these tend to grow beyond 1-5.000 requirements even for smaller mechanical products, increasing complexity without providing an overview of the dependencies between the requirements.

These pitfalls are recorded through a number of case studies within the research group of the authors, representing a comprehensive challenge for the majority of companies engaged in with product customization, variant management and mass customization.

There is no simple solution to avoid ending up in the situations described above, but the next section will go through some of the basic requirements for defining an architecture of the market with the aim of improving companies' decision-making. The underlying hypothesis here is that by improving the foundation of decision-making, the risk of ending up in these pitfalls will decrease.

3. What should the market architecture enable us to do?

In order avoid the classic pitfalls described in section 2 the definition of the market architecture should enable companies to fulfill these five overall tasks:

3.1. Requirements for a market architecture

3.1.1. Scope the development of product and production architectures

The market architecture should support the scoping the product and production architectures from a marketing point of view. This could include the focusing of which segments and applications to cover and which not to cover.

3.1.2. Elaborating the product applications within these business areas

In order to account for the product applications, the market architectures should support to provide an overview of the similarities and differences among the intended product applications, e.g. by visualizing the requirements of the critical performance parameters.

3.1.3. Make clear and differentiate the product properties

The market architecture should support the allocation of product features across the product variants and ensure an appropriate mix of different product properties across these.

3.1.4. Match the layout of product features with the layout of commercial variants

With multiple intended applications, multiple features to satisfy these, and multiple product variants to carry these features, the market architecture should provide a comprehensive overview of this “layout”.

3.1.5. Guide market pricing and match with balanced performance steps (optimal price and cost points)

The market architecture should support to harmonize the feature ranges with performance steps and match with the underlying cost levels in order to maximize the average contribution margins and avoid inappropriately scaled feature levels of product variants.

4. State of the art

Significant contributions have been made to clarify the market aspect of architecture-based product development. These include:

4.1. Adjacent fields of research

4.1.1. Platform strategies

Meyer and Lehnerd [1997] were the first to formulate three fundamental types of strategies in the market segmentation grid combining product segments with price/performance tiers; horizontal leveraging, vertical leveraging, and the beachhead approach. Kristjansson and Hildre [2004] formulated 17 influencing factors on which platform strategy to choose, and divided them into 4 categories: Core competencies, industry situation, market situation, and competitive strategy.

4.1.2. Product planning

Andreasen and Hein [1987] formulated product planning as the continuous parallel activity of determing the product strategy, conduct business search, follow up and supervise on product development activities through coordinating activities. Recent contributions include the challenges of variant management within these efforts [Jonas and Krause 2011]. Also, Riitahuhta et al. [2011] suggests the modeling of a Company Strategic Landscape combining aspects of product, value chain and strategy structuring as means of product-process synchronization

4.1.3. Enterprise Systems Engineering

American literature is oriented towards a wider definition of the concept of architectures [Rebovich and White 2011], working with a practical definition of an architecture as a model that details a system’s constitutive and behavioural characteristics in the form of activities, processes, functions, roles, taxonomy and framework. The notion here is that architectures are often rendered through views of deliberate perspectives to overcome human cognitive limitations.

4.1.4. Product properties vs. Customer preferences

Original contributions, as e.g. the Kano model, seek to characterize product attributes from the meeting between product and the customer preferences. This is done by differentiating between basic/threshold/obligatory attributes, performance/positioning attributes and excitement/delighting attributes (some variations of the model includes expected attributes as a sub-kind of performance/positioning attributes that can only be optimized to a certain limit, e.g. noise level). Other coherent frameworks exist for this partitioning of product attributes.

4.1.5. Et cetera

In addition to the fields mentioned above, requirements management, concurrent engineering, and related product management disciplines all mention the subject of the market aspect of architecture-based product development, but it is out of scope of this paper to go into further details here.

4.2. Gap

The current state-of-the-art lacks a coherent description of the elements described in the requirements listed in section 3. Individual elements are touched upon from different theoretical angles and with different aims, but these are not consolidated from a product architecture-based viewpoint. Aside from the lack of coherence, most contributions within this field consider the market perspective of architecture development as ‘focusing on maximum variety’ by default, without going into details about optimal *fit* of product applications and product features. Hence, some contributions become isolated in the product domain by e.g. developing advanced numerical optimization algorithms that seek to optimize the configuration of product families based on very simplistic product models. These methods might satisfy analytical needs, but they do not fulfill the requirements described in section 3.
Based on this sub conclusion, section 5 will elaborate on the suggested proposal of an architecture of the market.

5. The market architecture

5.1. Towards a definition

To overcome the challenges listed in section 2 and fulfill the requirements from section 3, a description of an architecture of the market is suggested. The architecture of the market should serve the development product programs by describing them from the market’s point of view, while maintaining a hierarchical structure that can act as a malleable object of alignment towards the product and production architectures. Figure 4 shows the three architectures with their five levels. Section 5.4 will elaborate on the definition of the included elements.

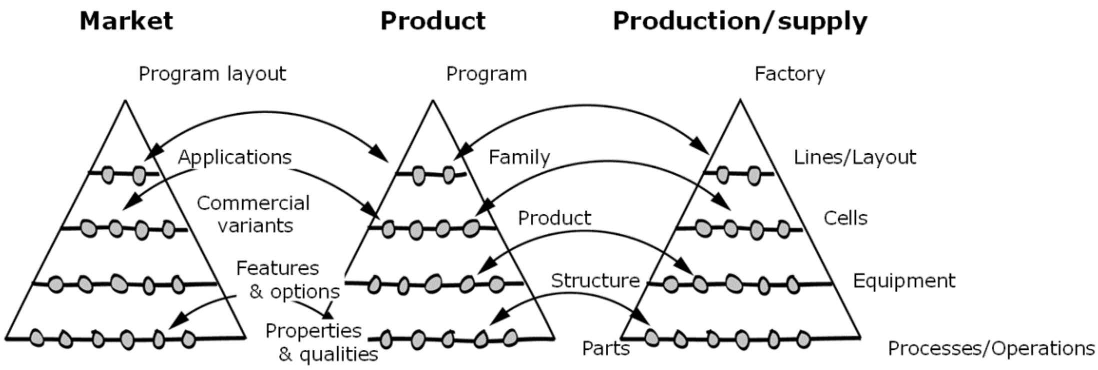


Figure 4: Alignment of market, product and production architectures
(expanded from Mortensen et al., 2008)

5.2 Use

The argument here is that modeling these three aspects concurrently during development is a prerequisite of creating attractive product programs. Alignment is seen as the mutual phenomenon of creating an optimal fit between the different architectures through activities of synthesis. However, in some cases of e.g. redesign or DFM activities, certain architectures can remain stable.
The architecture of the market is suggested to be applied both for mapping the market aspect of a product program for analytical purposes, and for maintaining an overview of decision-making during updates or new product program development.

5.3. Visualization

A visualization approach is chosen as means of staging the definition of architectures as boundary objects between the involved domains. Architectures, being a rather abstract phenomenon, can be very difficult to manage without appropriate models. In the attempt of bridging coordination between multiple domains with multiple levels of understanding, visual modeling is considered a prerequisite of intervention and malleability.

5.4. The five levels

In Figure 4, the three architectures are shown with their five levels. Figure 5 shows an elaboration of the five levels of the market architecture. As this is a general presentation, naturally the levels vary from case to case. However, a key aspect presented here is that the market architecture needs concurrent definition of all five levels.

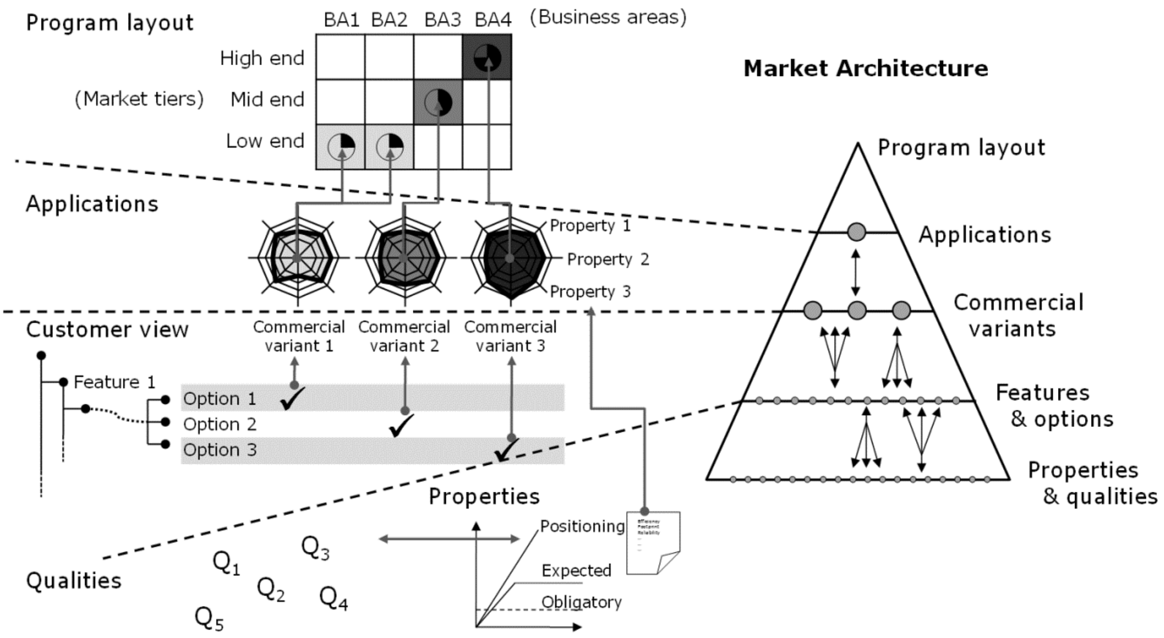


Figure 5: The 5 levels of the market architecture

The arrows between the levels to the right are indicating that there can be a one-to-many relationship between the three lower levels in both directions.

5.4.1. Program layout

Based on the market grid introduced earlier, the overall task of the program layout is to describe which business areas/segments to serve and which not to serve. The program layout expands this concept by adding the market life cycle stage to indicate whether a product family is newly introduced, maturing, or declining. The horizontal structure can indicate business areas or segments usually having a simple fit to the product applications. The mapping can help focusing the product architecture towards the most appropriate and favorable segments. Included in the program layout is also an indication of future derivate product families in order to avoid the pitfall of dead end program scaling.

5.4.2. Applications

The applications of the product are basically a segmentation of the market based on common use situations. The visualization of the requirements from each application (e.g. by radar diagrams) can serve to prioritize which applications to target the product program towards, while serving as a valuable input for differentiating what is variable between applications and what is common. The application overview is used as a mediating function between marketing and engineering for balancing wishes and possibilities, but also as a benchmarking tool for assessing the innovation height of a new product program compared to the recent product program. Applications *can* be similar within business areas/segments and across market tiers; in this case, it is the level of fulfillment that differentiates the product families from low-end to high-end.

5.4.3. Commercial variants

The commercial variants are the actual product variants of which the marketing department usually carries the market responsibility. In engineer-to-order companies, these do only exist retrospectively as commercial variants are customized for individual customers' specific needs. The projected cost of the commercial variants are mapped towards marketing's best guess of market price ranges. This allows for a comparison of the contribution ratios across the product program to evaluate the grouping and allocation of features and options across the commercial variants and improve product program contribution margins.

5.4.4. Features and options

The overview of the commercial variants are combined with the customer view [Harlou 2006]. Here the mapping of features and options are done towards these variants, and it is relevant to focus which features and options to include during the first product launch and which to postpone for future launches. Some features might be de-scoped and omitted if the means of realization is not in place, payment willingness is considered absent, or if the overall market offerings are regarded as being too wide and in need of focus. The main task of linking the features and options towards the varying applications is to separate the cost-effective and reusable 'core' from the variations provided by these.

5.4.5. Properties and qualities

The lowest level contains the individual product attributes of interest to the customer, namely product properties and product qualities. A well-working partitioning here is the differentiation between obligatory (must), expected (improvement to a certain level) and positioning properties (differentiating from competing products), or simply just need/nice-to-have. In some industries the existence of excitement properties (*delighters* that surprise the customers if included) are just as important. This definition might seem loose, but due to variation between industries and products, no general partitioning is suggested here. This information is often stored in requirements lists, but it is important to link these directly to the features and options fulfilling these requirements.

5.5. Linking the architectures

As time-to-market is mostly decided by the size of the engineering efforts, and investments are mostly decided by changes implemented in production, it is of fundamental importance to link the mapping of the market architecture to the product and production architectures, and optimally develop these three concurrently. The market architecture constitutes the basis of a focused product architecture, thus making all efforts of focusing the production/supply tasks dependent on the ability to focus the market architecture.

6. Experiences from application

The concept of the market architecture has been tested, refined and developed through a number of action-based research studies. Three of them are shortly commented here:

6.1 Early-stage architecture development

In the context of a larger industrial manufacturer of mechanical products, a new promising technology was considered the corner-stone of a new generation of product families. To ensure forward compatibility of the technology and prepare for laying out a path of potential future launches, architecture work was engaged. Since the project was in its early stages, the work was focused on the interplay between the market and product architecture. Here, the market architecture provided an overview of the product applications, the commercial variants and the possible features and options. The main task here was to separate the application dependent options from the reusable core to prepare it for mass production and attractive cost levels. The results was a proposal for the first generation of a product family architecture with prospects of a line of possible future derivatives, matched with an overview of the variations between proposed variants and selectable options. Also, the market architecture helped to select prototype installations that represented the total spectrum of possible future variants, and to focus the basic scaling principles of the product architecture in alignment with the market architecture.

6.2 Performance critical OEM-supplier

In a world leading OEM supplier of performance critical components for the energy industry, research work was initiated in order to clarify how the company could benefit from architecture thinking and to test and improve the modeling techniques. The company had severe difficulties with their time-to-market, and no reuse existed between engineer-to-order customer projects. Also, the investment level and resource consumption of a standard development project was too high to serve other than a few large OEM customers. The definition of a market architecture helped to scope the definition of coherent product and production architectures. By creating an overview of the dynamics in the program layout and by systematically listing the requirements from lead applications, the market architecture helped to focus engineering efforts. This created a basis for reuse of engineering resources and production equipment, shorten time-to-market for derivative products, and remove the risk elements from high initial investments in production equipment having the tendency to scare away OEM customers unwilling to co-finance such start-up activities.

6.3 Fundamental architecture selection

In an industry leading electronics company, a major program development project was severely postponed. As the company is a result of mergers in the past, different product architecture strategies were present alongside each other, and the major dilemma was whether to switch to a fast-track development program using current technology with market launch in 2012/2013, or accept the postponement and develop the product program as initially proposed being ready for launch in 2015. With contradictory interests and different perceptions of the market situation, the modeling of the market and product architecture of the development program was initiated to evaluate which fundamental architecture selection options would serve the company the best. The classic pitfalls of multiple development projects competing for the same sales (market cannibalization), and dead end scaling strategies were largely at stake here. The basic trade-off between maintaining the market position with the fast-track alternative, or wait and improve the feature offerings with possible loss of market share, was elaborated in the dimensions described in section 5. This modeling of the alternative market and product architectures served to improve the decision foundation of the company, e.g. by aligning the scaling strategy of the product architecture with the scaling strategy of the market architecture.

7. Discussion

Only dispersed bodies of literature have treated the market aspects of architecture-based development of product programs and families systematically. This contribution should be regarded as another important piece of a puzzle outlining a suggested framework based on the authors' practical experiences within this challenging area. Thus, the contribution presented here does not represent a complete framework on its own, but serve as a contribution to the framework of the authors (represented in Figure 4) and the scientific body of knowledge.

An important strength in this contribution is that the application of the concepts presented can be applied without the need of crossing huge barriers. Many companies might have some elements of the market architecture well documented and under control when looking retrospectively at current product programs, but the contributions presented here underline the importance of modeling the market architecture proactively during development and in coherence with the product and production architecture. A possible deficiency with the concepts presented here arises from the same situation, as the need of adaptation (and competence to do this) is needed in order to integrate the work with the market architecture successfully.

8. Conclusion

This paper has presented and elaborated on the definition of an architecture of the market. Is has been described how the definition can support the difficult decision-making of providing sufficient variety in product programs to maximize payment willingness from customers without sacrificing internal complexity. The market architecture definition has been motivated through the outlining of seven classic pitfalls encountered by companies failing to scope and fit product programs appropriately from the market's point of view. The response to these challenges was formulated through five requirements for the definition of an architecture of the market, and the state-of-the art was screened and briefly summarized to identify the knowledge gap. Subsequently a proposal towards a definition of a market architecture was described shortly including five levels, and the successful application through three case studies was shortly reported.

Implementation of architecture thinking across market, product and production domains, however, is a very challenging task. Many preconditions and prerequisites exist for successful implementation, e.g. a modern IT infrastructure, organizational ownership, sufficient resources/competences and high-level anchoring of the initiatives.

Regarding further work, the detailing of the modeling elements included in this presentation can be mentioned. As this has been a short and general presentation, the further detailing, testing and refinement of the modeling techniques behind the market architecture are relevant activities to succeed this paper. The generalizability can be considered a strength of this presentation as well as a weakness, since reality in many cases needs a higher resolution of detailing that this presentation format allows for.

References

Andreasen, M.M., Hansen, C.T., Mortensen, N.H., 1996. The Structuring of Products and Product Programmes. Proceedings of WDK2, 4, 28

Andreasen, M., Mortensen, N., Harlou, U., 2004. Multi Product Development–New Models and Concepts, 75-86

Andreasen, M.M., Hein, L., 1987. Integrated Product Development. Springer, Berlin

Andreasen, M.M., Olesen, J., 1990. The Concept of Dispositions. Journal of Engineering Design. 1, 17-36

Harlou, U., 2006. Developing Product Families Based on Architectures - Contribution to a Theory of Product Families. Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby

Jonas, H., Krause, D., 2011. Strategic planning for modular product families. 5, 112-359

Kristjansson, A.H., Hildre, H., 2004. Platform strategy: A study of influencing factors. Proceedings of Norddesign 2004, 114-123

Kvist, M., 2009. Product Family Assessment, Ph.d. thesis. DTU Management, The Technical University of Denmark, Kgs. Lyngby

Meyer, M.H., Lehnerd, A.P., 1997. The Power of Product Platforms - Building Value and Cost Leadership. The Free Press, New York

Mortensen, N.H., Hansen, C.L., Hvam, L., Andreasen, M.M., 2011. Proactive modeling of market, product and production architectures. Proceedings of the 18th International Conference on Engineering Design : Impacting Society through Engineering Design, 133-144

Mortensen, N.H., Pedersen, R., Kvist, M., 2008. Modeling and Visualizing Modular Product Architectures for Mass Customization. International Journal of Mass Customisation. 2, 216-239

Rebovich, G., White, B.E., 2011. Enterprise systems engineering : advances in the theory and practice. CRC Press/Taylor & Francis, Boca Raton

Riitahuhta, A., Lehtonen, T., Pulkkinen, A., Huhtala, P., 2011. Open Product Development, in: . Springer, pp. 135

M.Sc.Eng. Christian Lindschou Hansen
PhD student
The Technical University of Denmark
DTU Mechanical Engineering
Produktionstorvet
DK-2800 Kgs. Lyngby
Denmark
Direct: +45 4525 1646
Mobile: +45 4046 4233
chrlh@mek.dtu.dk
http://www.kp.mek.dtu.dk/

6.4 Paper D

PROACTIVE IDENTIFICATION OF SCALABLE PROGRAM ARCHITECTURES:
HOW TO ACHIEVE A QUANTUM-LEAP IN TIME-TO-MARKET

International Journal of Industrial Engineering 2014, Vol. 21 Issue 2, p74-85. 12p.

Christian Lindschou Hansen & Niels Henrik Mortensen

Department of Mechanical Engineering
Product Architecture Group
The Section of Engineering Design & Product Development
Technical University of Denmark
Building 426
DK-2800 Kgs. Lyngby
Email: chrlh@mek.dtu.dk, nhmo@mek.dtu.dk,

This paper presents the Architecture Framework for Product Family Master Plan. This framework supports the identification of a program architecture (the way cost competitive variance is provided for a full range of products) for a product program for product-based companies during the early stages of a product development project. The framework consists of three basic aspects: the market, product program, production and a time aspect – captured in the multi-level roadmap. One of the unique features is that these aspects are linked, allowing for an early clarification of critical issues through a structured process. The framework enables companies to identify a program architecture as the basis for improving time-to-market and R&D efficiency for products derived from the architecture. Case studies show that significant reductions of development lead time up to 50% is possible.

Significance: Many companies are front-loading different activities when designing new product programs. This paper suggests an operational framework for identifying a program architecture during the early development phases, to enable a significantly improved ability to launch new competitive products with fewer resources.

Keywords: Product architecture, program architecture, product family, platform, time-to-market, scalability

1. INTRODUCTION

Many industrial companies are experiencing significant challenges in maintaining competitiveness. There are many individual explanations behind these, but some of the common challenges that are often recorded from companies are:

- Need to reduce time-to-market in R&D:
    - Shorter product life cycles are increasing the demand for faster renewal of the product program in order to postpone price drops and maintain competitive offerings (Manohar et al., 2010)
    - Loss of market share in highly competitive markets call for improved launch responsiveness to match and surpass the offerings of competitors (Chesbrough, 2013)
    - Protection of niche markets and their attractive price levels requires continuous multi-launches of competitive products (Hultink et al., 1997)
- Need for achieving attractive cost and technical performance levels for the entire product program
    - Increased competitiveness requires all products to be attractive both cost wise and performance wise (Mortensen et al., 2010)
    - Focusing of engineering resources requires companies to scale solutions to fit across the product program (by sharing) and prepare them for future product launches (by reuse) (Kester et al., 2013)

- Sales forecasts from global markets are affected by an increasing number of external influences making it more and more difficult to predict the sales of individual product variants, thus leaving no room for compromising competitive cost and performance for certain product variants (Panda and Mohanty, 2013)

These externally induced challenges pose a major task to the whole company. As such, many approaches exist to handle these challenges which are of organizational- , process-, tool-, and competence nature originating within research from sciences across business, marketing, organization, technology, socio-technical, and engineering design. The research presented here originates within engineering design and product development focusing on the development of a program architecture for a company. Although originating from the engineering design domain which is naturally centered in the R&D function of a company, the development of program architectures have relations that stretches far into the marketing, product planning, sourcing, production, and supply chain domains as well as into the companies' overall product strategy.

Several original contributions exist from (Lindemann et al., 2009; Harlou, 2006; Ulrich and Eppinger, 2000; Ericsson and Erixon, 1999; Meyer and Lehnerd, 1997; Ulrich, 1995) outlining important dimensions of architectures. Also, many recent contributions exist (Johannesson, 2013; MacDuffie, 2013; Campagnolo and Camuffo, 2010), but companies are still struggling with the adaption of various concepts and methods while the challenges from the competitive surroundings are not decreasing (Krause et al., 2013). One element is the lack of operational approaches to support in particular the synthesis, instead of merely analysis, of product programs (Jiao et al., 2007). By 2004, two-thirds of contributions presuppose the a priori existence of a product family or product program on which to apply sophisticated optimization techniques and algorithms (Simpson, 2004), and later literature studies do not report a change here (Simpson et al., 2014; Campagnolo and Camuffo, 2010). In many cases, however, it is not appropriate to take a starting point in the optimization of old structures and building blocks, as the integration of new technology and new solutions may provide the basis for developing better products.

This paper presents the Architecture Framework for Product Family Master Plan enabling companies to create an overview of the current state of the market, product, production, and roadmap aspects of the future program architecture during the early development phases where CAD drawings and other product documentation does not yet exist. The state can be a mix of tentative and/or decided during earlier phases, and stabilized and/or verified during later phases. But in addition to an overview, the framework also enables the substantiated evaluation of e.g.:

- Adding or removing a feature
- Decoupling or integrating a module
- Precipitate or postpone point of customization
- Adding or removing a performance step etc.

The overview and evaluation does not only cover the scope of the next product launch, but also includes the preparation towards future launches to be derived from the architecture under development. Thus not only including the constituting structural description of an architecture (parts, modules etc.), i.e. what the architecture *is*, but also the behavioral description (derived features, future launches etc.), i.e. what the architecture is able *to do*.

Proactive is mentioned in the title for two reasons:

- There is a need for identifying the program architecture before the specification of individual variants are *frozen* making it impossible to rationalize decision-making on a program level.
- There is a need to include the next two, three or four product launches to be derived from the architecture and take these into consideration explicitly – meaning that they should be included in modeling activities to the extent meaningful input can be provided to support their definition.

The next section will briefly present the research method followed by an outline of the current barriers and challenges in identifying program architectures to overcome the challenges presented here. Subsequently, the state-of-the-art is reviewed including a discussion of the methodical gap that the authors have identified. After this, the framework is presented followed by the description of four industrial case applications.

2. RESEARCH METHOD

The framework presented here is developed using literature and experience. The framework is developed by researchers with an applied research background within product architecture, product modeling, product configuration, mass customization combining traditional domains of mechanical engineering with operations management. Experience has been built up by 15-20 years of case studies conducted in companies developing, producing and marketing products of a mainly mechanical character. The case studies have matured the framework's modeling techniques to their current state through the reflective use in a number of different companies and situations. This paper will include four recent cases demonstrating the wide application of the framework, but also to show the need for adapting the framework and modeling techniques to fit the exact requirements arising from the company specific challenges. These four case studies represent prescriptive research studies including different levels of descriptive evaluation of the framework's support to the given case.

The basic hypothesis behind the framework is that the application of explicit modeling techniques promotes the identification of an architecture, based on the assumption that explicit modeling promotes the basis for decision-making. Yet, a number of requirements exist to ensure this result – here are two of them:

- The framework's models must be staged as boundary objects (Star and Griesemer, 1989) between competent and influential professionals from the market, product and production domain
- The framework's models are properly prepared for constructive intervention and malleability – through structured review sessions (Cooper, 1992)

The models may be intra-domain specific for clarification of details, e.g. a proposal for engineering building blocks only to be reviewed by R&D, or cross-domain hybrids to promote alignment, e.g. the inclusion of a feature and its impact on production.

3. BARRIERS AND CHALLENGES

There are a number of context specific reasons why companies do not succeed in identifying profitable program architectures during the early phases of development projects. However, a number of explanations are recurring.

Below is a list of reasons repeatedly recorded during interviews with R&D management teams from 2010-2013 across 12 different companies producing products of mainly mechanical character. The companies operate in seven different industries. The quotes were recorded during the interviews.

"What we measure is what we promote"
Project managers are not measured upon the preparedness towards future launches. Most often, business cases do not include any additional sales figures, development costs or lead-time beyond the impact of first product launch, making it very difficult for project managers to justify the preparation towards future product launches if this requires additional investments – even though this is often not the case. Concurrently with companies being more and more project oriented, the forward-looking thinking beyond the strict project boundaries is made increasingly difficult, leaving it up to the skilled or far-sighted project managers to push and drive this on their own.

"Our project management techniques can compromise architecture thinking"
Most companies are organizing their R&D activities according to phase-gate models (e.g. Stage-Gate®) of many different variations. This is not a problem in itself. Although in some cases, phase-gate models assume linearity in the sense that market requirements must be almost completely finalized before product and production concept work can start. This can be lethal to architecture work, as no meaningful prioritization of market requirements can be carried out without knowing the product and production consequences. For instance, how to prioritize the implementation of a feature if the development lead time, production investment level and module consequences are unknown?

"We have no forward-looking scaling strategies in place"
Solutions are developed to meet dedicated specification by discrete values without considering how to scale in terms of added functionality or a different performance level. Therefore it becomes difficult to ensure that appropriate performance and cost points can be realized for future product variants. Solution principles are often not stretched across the full spectrum to make sure that the best performance to cost ratios are met for the first, second, third and fourth launch wave of products. The consequence is that solutions are not scalable in the sense that they cannot deliver competitive performance beyond their primary product implementation.

“Our development projects become too focused too soon”
In the effort of streamlining projects to make them capable of meeting deadlines and hand-in deliverables in due time, projects sometimes become too focused too soon. Often this ends up having the complete opposite effect as projects constantly challenge their scope and boundaries, unsure about whether to include e.g. neighboring product applications or whether to support the next generation of technology. Instead, the experience of the authors is that the inclusion of neighboring product applications and future generations of technology from the beginning of the program architecture work helps to improve the scoping of the lead development project, as the scope and boundaries for the lead product are now clearly defined (see section 5). However, this requires a program overview of target and neighboring product applications, which often does not exist.

“The amount of information available is overwhelming and obtaining an overview is difficult”
The amount of information in most modern companies is often overwhelming in itself representing a huge barrier in itself. Access to ERP and PDM/PLM systems is no assurance for carrying out successful data mining in order to structure data into a format that allows comprehensive decision-making. As the data complexity is increasing, the need for creating simple hybrid visualizations (by combining data from different sources) increase.

“There is a lack of tool support”
Many phase-gate models require the concept of the product program to be done in order to pass Gate 0, but often do not provide any tools or ways for project managers to carry out this sort of overview. Therefore, it is up to project participants to think up new ways of representing early product program concepts in parallel to their existing tasks. Similarly, very few models support the evaluation of consequences across market, product and production areas. Furthermore, the responsibility of highlighting these consequences is often redistributed across different project participants with diverse backgrounds and varying tool support.

“It is difficult to work with a program of solutions – it is easier to work with just one product”
There is plenty of complexity to deal with in most development projects, and the scope of a program architecture can often scare competent resources from the complexity in this widened task. Additionally, only few resources are dedicated to the projects during the front-loading phase (phase before Gate 0), making it even more attractive to focus the development project on a few product variants – simply to cut down complexity. The effect of this de-scoping is the lack of preparation towards future launches resulting in increased time-to-market and decreased utilization of resources.

“We lack responsibility – no one is given the task”
In many companies, only upper management has the actual architecture responsibility, as project managers are very often only responsible for the first product launch, and no other functional departments in practice own this cross-functional area. Therefore, even though program managers exist in certain companies to facilitate coordination between project managers, very little preparation towards future product launches actually exists. On a portfolio level, the program responsibility is often found in the market domain, whereas no one is responsible for establishing a clear link between the portfolio level and the program level where solutions are developed and maintained by the R&D function. Silo thinking is still predominant when it comes to linking the portfolio plans with projects’ ability to think ahead towards future product launches. However, elements of architecture responsibility has begun to see the light of day in some companies, but the challenge here is that the role of program architects is often not sufficiently defined leaving them with difficulties in taking critical decisions.

“How to select the point of departure – we have difficulties getting started”
Previously, a central concept to architecture work was the separation of preparation and execution activities. A clear architecture makes this possible in principle, but companies often dedicate their best resources for the execution projects, making the preparation projects fail dealing with the very early and highly strategic decision-making taking place here. This is often too big a challenge for less experienced resources. Therefore, it is the experience of the authors that the architecture preparation activities should be carried out as a front-loading activity to a larger development project – during the early development phases.

4. STATE-OF-THE-ART

The review of the state-of-the-art includes a review of five different groups of supporting methods for the identification of architecture for a product program. Function-based models taking their starting point in functional structures by focusing on the clear mapping between functions and modules as a basis for a modular architecture. Matrix-based models take their starting point in an existing structure, using sophisticated algorithms to identify functional clusters that can later become module candidates. Concurrent engineering takes it starting point in the concurrent design of product and production architectures, while DFM provides a set of specific tools to promote this alignment – however, without providing adequate support for identifying an architecture for an entire product program, but focusing on single products. Methods based on mathematical models are also briefly reviewed.

Function-based models
Methods describing the development of modular product architectures often choose to start with the conscious mapping of functional structures into physical modules (Pahl et al., 1996). Functions can be represented in function-based models, e.g. functions-and-means trees (Andreasen, 1980), or by schematics of the product including physical elements to a meaningful extent (Ulrich and Eppinger, 2000).

The understanding product functions can be used in different ways to identify possible modules. To improve the identification of modules and make sure that the modular architecture will serve its objectives, Ericsson and Erixon (1999) define a set of module drivers. The module drivers can support the reasoning behind the module identification by elaborating the justification of the modules' existence, e.g. 'planned product changes' module, 'process' module, 'different specification' module, 'technology evolution' module etc. The module drivers are a part of a comprehensive framework called Modular Function Deployment (MFD), which in analogue to the QFD method provides support for the linking of relationship between the module drivers and technical solutions.

Matrix-based models
Another approach to identify modules is the application of design structure matrices (DSM). This approach takes its point of departure in the decomposition of a product into parts and/or subsystems while identifying the relations (and possible future interfaces) among these (Pimmler and Eppinger, 1994). By applying different algorithms and clustering techniques, it is possible to encapsulate functional 'chunks' that have the potential of becoming physical modules, due to their functional interrelations. DSM techniques are the subject of many research initiatives and serve as the basis for an array of derived methodologies. A recent example of this is the Multi-Domain-Matrix (Lindemann et al., 2009; Lindemann and Maurer, 2007). Alternatively, other design tools focus more on the specific task of examining different functional flows with the aim of identifying modules (Stone et al., 2000; Otto and Wood, 1998). These methods are heuristically based.

Other general methods focus on the identification of common features in the existing product program in order to point out the basis of the product architecture (Gonzalez-Zugasti et al., 2000). By formulating the design task as a quantitative problem, which can be subject to optimization, this method is balancing inputs from requirements and product variants design with data models of performance and costs. By iteration, the optimal product variants are designed and evaluated through quantitative performance metrics.

Concurrent Engineering
From the associated area of Concurrent Engineering, one can also find research into the concurrent development of product and production architectures, with phrasings such as 'methods supporting the development of product platforms'. Nevertheless, interesting contributions are submitted within this area. Fine (1998) introduces a three-dimensional methodology superimposing the traditional domains of concurrent engineering, by suggesting the linking of technology, architecture and focus relations in the process, product and supply chain domains. Fixson (2005) proposes an important step of operationalization of this 3D-Concurrent Engineering approach (3D-CE) by developing a multi-dimensional framework that enables comprehensive assessment of alternative product architectures.

The concept of Architecture for Product Family (APF) is introduced as a conceptual structure, proposing logics for the generation of product families (Du et al., 2001; Jiao and Tseng, 1999). The Generic Product Structure (GPS) is then proposed as the platform for tailoring products to individual customer needs. Ko and Kuo (2010) presents another systematic method for concurrent development of product families, by combining QFD-based methods with quantified DSM-techniques and morphology analysis to visualize concepts.

Design-for-Manufacture (DFM)
Original contributions from Olesen et al. (1992) proposed a framework for the concurrent development of manufacturing supported by the Theory of Dispositions (Andreasen and Olesen, 1990). This is done by proposing a set of models aligning the product design and the product life system phase of manufacturing to create a fit. However, the case with DFA and

DFM methodologies, the main focus is single product development. Herrmann et al. (2004) comments that an extension of the DFM tools to comprise multi-product development will hold the key to achievement of competitiveness.

Mathematical models
Some researchers have undergone the task of developing methods based on mathematical models. Some methods are based on measures of modularity, which act as subjects of optimization using different techniques (Guo and Gershenson, 2007). Others seek to integrate product platform, manufacturing process and supply chain decisions through the application of mathematical models, thus extending the concept of the Generic Bills of Materials (GBOM) by quantifying relations between decisions from the different domains (Huang et al., 2005).

Conclusion
It is evident that the contributions mentioned above can play a role in the identification of program architectures.

Situated in this cross-functional research field, it is clear how research centered within either the product or production domain, tend to leave out important aspects of the adjacent fields, and considering the identification of program architecture this is a deficiency considering the contributions listed above.

Extensive research is also found within the reengineering of business processes and different means of optimization of operations, but these areas exclude necessary details within the field of architectures. They are simply not concrete enough, or deal with suboptimization of operations and processes leaving out the product domain.

The mathematically based models are centered on analytical activities, thus omitting support of synthesis activities, which also would require a different form of representation. For the mathematical models to be applicable, a very delicate balance between concretization and data basis has to be present. Skinner (1978) formulated this dilemma a long time ago as:*"Mathematical models that include enough variables to be realistic can become fantastically difficult to formulate and program. And oversimplified models, eliminating vital factors, are inaccurate and misleading"*.

Furthermore, evaluating a product architecture based on the product itself, not including the market or production architecture, misses the important relations between these three, making it impossible to capture the benefits of *alignment*. Also, no basis is found to support the definition of the behavioral aspects of program architecture, that is, which products can and cannot be derived from the architectures and the performance properties of these. Lastly, the consequence evaluation described in section 1 is not supported in order to enable a substantiated evaluation of the program architecture in its entirety.

5. ARCHITECTURE FRAMEWORK FOR PRODUCT FAMILY MASTER PLAN

Figure 1 shows an overview of the modeling techniques making up the framework. The framework allows for the identification and definition of a program architecture during the early phases of a development project for a full product program. The framework consists of an architecture of the market, product and production (including supply and distribution) as well as the time-aspect captured in the multi-level roadmap.

The term "program architecture" is reflecting the result of aligning the market, product and production architectures and coordinating these with the multi-level roadmap in order to ensure an improved preparedness towards future launches. Thus the term differentiates itself from a product architecture in the sense that a program architecture requires the alignment with market and production architectures to achieve both competitiveness *and* attractive cost levels – from first launch and over time. The framework can be considered an extension of the basic framework of the Product Family Master Plan (Harlou, 2006), following the basic partitioning in market-product-production from the research in integrated product development (Andreasen and Hein, 1987).

The term "scalable" from the title of the paper refers to the scalability of solutions by development of modules that are designed to be scalable in the desired range and in relation to key design characteristics. This enables the achievement of attractive cost of system performance optimized for price, value or performance.

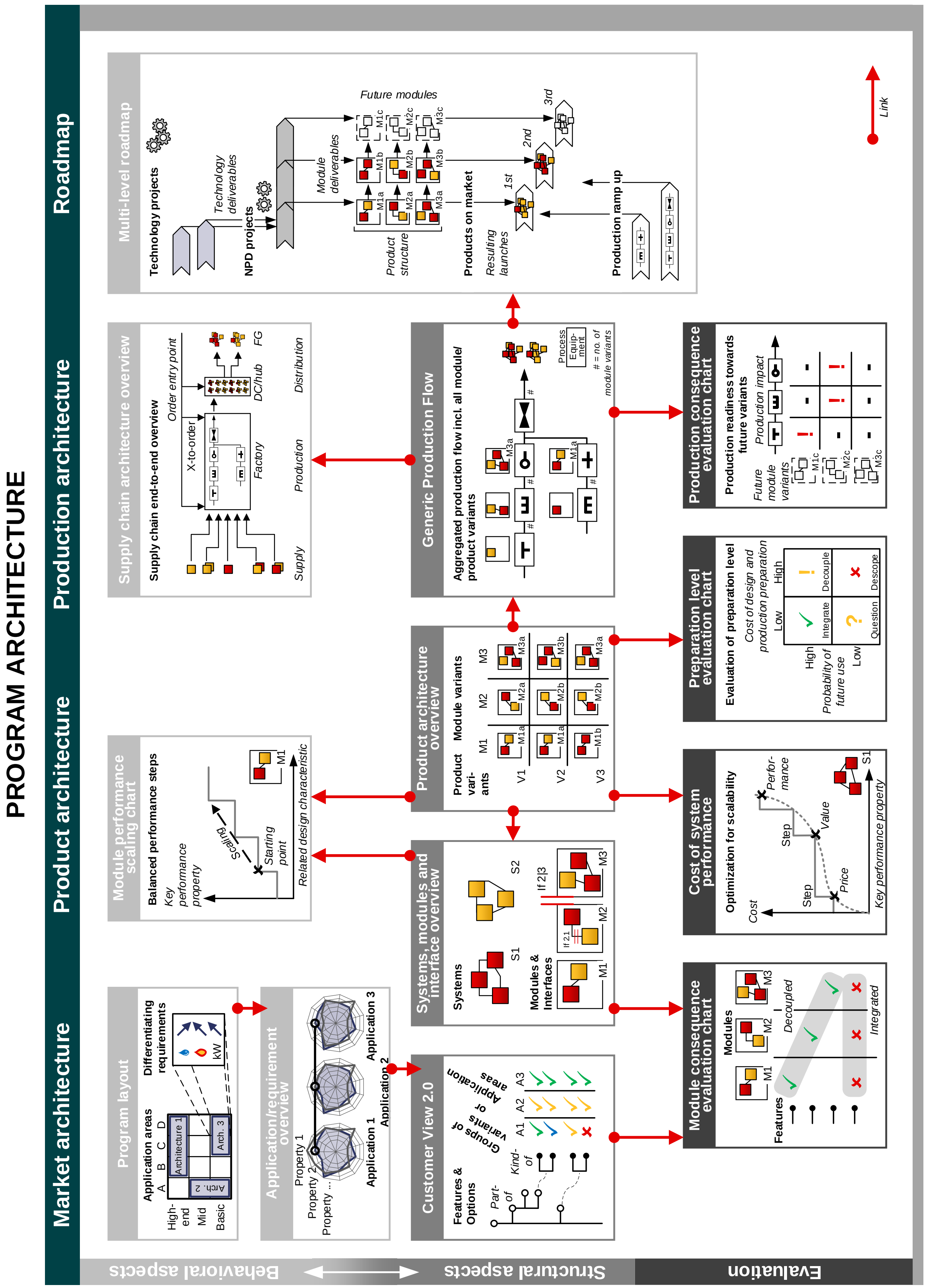


Figure 1 – Architecture Framework for Product Family Master Plan

As indicated on the vertical left, each part of the framework consists of a set of models that capture constitutional aspects – what the architecture *is*, as well as behavioral aspects – what the architecture is able *to do*. Besides these, a set of evaluation models supplement the constitutional and behavioral models by allowing for a structured evaluation of the consequences that the program architecture lead to. The framework consists of a mix between intra-domain views that focus on e.g. the product aspects, and cross-domain views that focus on the link between for example product and production. The intra-domain views are to be applied for clarifying aspects within a specific domain, whereas cross-domain views are to be applied for creating alignment between domains.

The *market architecture* is the market perspective on the product program reflecting the design of the product program from the market's point of view (Hansen et al., 2012; Mortensen et al., 2008). The identification and definition of the market architecture serves to avoid a number of classic pitfalls concerning the market perspective of product program development: Unplanned market cannibalization due to overlapping offerings, lack of differentiation between high-end and basic market tiers, dead end scaling strategies with no natural continuation of product launches planned, an uneven mix of product properties across the product program and price and cost-points that are out of balance. Another pitfall is the exponential growth in requirements formatted in a free-text format that are difficult to apply to gain an overview of differences between different product applications.

Program Layout
The Program Layout serves two important objectives: The clear scoping of what the program architecture should cover and what not to cover, and the identification of the requirements that differentiate the market segment and market tier. The Program Layout resembles a market grid (Meyer and Lehnerd, 1997), but is extended by inclusion of the differentiating requirements which hold the key to competitiveness in the specific market segment. Also, the modeling of the layout on a program level (instead of product family level) allows for a clear differentiation of individual product families.

Application/requirement overview
The application/requirement overview serves the objective of identifying which requirements that are common to all product applications and which vary. This overview can also take on a mediating function between marketing and R&D to balance market wishes towards engineering possibilities, or be used as a benchmarking tool to identify where exactly the new product program should differentiate from the previous one. The clear definition of this is a central prerequisite for focusing the engineering resources during development.

Customer view 2.0
The Customer view 2.0 serves the aim of detailing the feature and option layout across the product program. In addition to the original customer view (Harlou, 2006), this expanded version allows for detailing the feature and option layout to suit the state of the development project by either modeling application areas or groups of product variants in horizontal columns. Mapping of features and options directly towards the specific commercial variants may result in an overly detailed level of granularity during the early phases of product program development.

The *product architecture* is the constitutional and behavioral elements of the program architecture concentrating specifically on the physical realizations of the product program in terms of systems, modules (including key components) and interfaces.

Systems, modules and interface overview
The systems are the carriers of functionality and the viewpoint is applicable and important where it is impossible to encapsulate functionality into separate modules (e.g. control, hydraulics). The functionality of systems should be independently defined and strive for a simple mapping towards the modules implementing the systems to allow for a flexible decoupling. The modules are encapsulated to achieve beneficial effects in the value chain (e.g. development, sourcing, production) or the entire product life cycle (e.g. installation, upgrade, recycling). The most critical interfaces are identified and described in this overview to make sure that the stabilization of these is considered a key goal during architecture conceptualization.

Product architecture overview
The product architecture overview serves the aim of outlining the sharing of modules between main product variants to create an overview of where variants of modules are needed and where modules can be used across several product variants. This provides a powerful tool to oversee the status of the program development project.

Module performance scaling chart
In order to obtain the desired steps of performance for the main key properties, the scaling principle of relevant modules can be outlined in order to ensure that the performance steps are balanced. The performance of key properties is mapped towards defining design characteristics (e.g. *power* or *efficiency* in relation to *weight* or *size*), in order to clarify how to

achieve balanced steps in relation to the specific design characteristics that are most sensitive to achieving the desired performance.

Evaluation: Module consequence evaluation chart
This chart evaluates how the feature layout is mapped towards the modules. If the implementation of a feature is dispersed across several modules the design is integrated and an attempt should be made to limit the implementation of a feature to fewer modules (shown by red crosses). If the implementation of a feature is a 1:1 relation with a module, functional encapsulation has succeeded and the functionality is decoupled (shown by green tick marks).

Evaluation: Cost of system performance
This chart evaluates how the relation between performance and cost is for main systems. Three optimization points are suggested here, namely price, value and performance. *Price* is the low-cost version where only necessary functionality is implemented, *value* is where system performance reaches a market competitive level before prices start to climb, and *performance* is where the maximum possible system performance can be reached without compromising the program architecture.

Evaluation: Preparation level evaluation chart
This chart supports the evaluation of whether to include design and production preparation for implementation of features and functionality to be utilized beyond the first product launch. Depending on the probability of future use and the cost of preparation, it is suggested to either integrate the feature (high probability, low cost), decouple the feature (high probability, high cost), question the feature (low probability, low cost) or descope the feature (low probability, high cost). Furthermore, the ratio between the resources needed to include the preparation now or postpone it to a later development project should be considered carefully. If this ratio is 1:1, it is reasonable to postpone the implementation, if the program architecture is not compromised. If this ratio is 1:10, a considerable amount of R&D resources can saved by implementing the feature now.

The *production architecture* describes the constitutional elements of the production setup along with the behavioral elements associated with the function of the production setup in relation to supply and distribution/delivery.

Generic Production Flow
This view captures the *variant creation points* of the production flow, the *order entry points* of the meeting between the planning/order flow and the production flow, and the *customization points* where goods become unique to specific product variants and/or customers. The Generic Production Flow (Mortensen et al., 2011) contains an aggregated view of all main production equipment, modules and product variants and enables the assessment of whether variants are created in the most appropriate sequence, timing and by which impact on production equipment.

Supply chain architecture overview
In relation to the supply chain, the performance of the production flow sets requirements for delivery sizes and frequencies of inbound materials and the need or possibility for postponed variant creation in distribution centers or hubs. This overview captures the significant elements in relation to the overall supply chain design, in order to explore opportunities of complexity reduction during production and bringing down work-in-progress, inventories and delivery times.

Evaluation: Production consequence evaluation chart
This chart supports the evaluation of production equipment readiness towards future product variants by identifying the production impact of future module variants that are not decided yet. In close relation to the preparation level evaluation chart, this chart supports the specific evaluation of the production equipment to evaluate which production lines to remain dedicated to and which to include flexibility towards future expansions.

The *roadmap* described the time aspect of the program architecture. The view of the *multi-level roadmap* combined the timing, duration and delivery from technology projects, development projects with a roadmap of products on the market and production ramp-up activities. The unique aspect here is explicit inclusion of the product structure to enable road mapping on a modular product level in order to ensure the alignment between the product architecture and careful planning of multiple product launches.

Applying the framework
Four important basic conditions are identified as critical for successfully applying the framework. The framework is developed to be applied as part of a program architecture review process:

- There is need for *appropriate staging* of the modeling techniques in order to ensure sufficient interaction between relevant competences. The staging can be improved by using A0-format posters to capture the vast amount of details necessary to support architecture decision making.

- It is necessary that practitioners have sufficient *practice in modeling techniques*. The modeling of critical aspects across market, product and production domains requires sufficient theoretical understanding of the dependencies across practical training but also practical training.
- In order for the modeling techniques to be usable and pertinent, it is important to *orchestrate reviews* on a frequent basis to ensure that the architecture models reflect the most recent status of the development project. Reviews must be thoroughly prepared by the respective resources in order to facilitate an efficient exchange of information from different stakeholders.
- It is of fundamental importance that *key competences are included* in the architecture review process as architecture decision-making normally requires the highest competence level available in a company.

6. INDUSTRY APPLICATION: FOUR CASES

The Architecture Framework for the Product Family Master Plan has been used and evaluated in a number of cases whereof four of them are reported here. Not all cases use the full framework. Instead the framework was tailored to meet specific needs by focusing on gaining a significant impact from the use of a few models and views rather than forcing these through the full framework.

In a global company developing and producing mechanical solutions for industrial applications, a subset of the Architecture Framework has been used in more than 50 development projects. The overall goal is to reduce complexity in development project and optimize product development investments to create room for focused innovation. The Architecture Framework is used to create an alignment between market representatives, R&D engineers and production technicians by initially focusing on commitment towards requirements (market architecture) to ensure that development activities fulfill the goals specified from market side and to make sure that R&D and production can deliver faster and with a higher certainty. The Architecture Framework has succeeded in providing an overview for the project participants to identify the project critical path, determine which solutions to use across the product program and by which scalability these solutions can be developed. The Architecture Framework is owned by the product responsible while the ownership of individual views can be distributed with specific responsibility. The application of the framework is targeted to improve time-to-market by more than 40% (more than 50% of early phases and 30% of later phases) and recent projects show promising results, where several derivative launches have been prepared for during the main project phase. These derivative launches would normally have been postponed for subsequent development projects.
Views included: Customer view 2.0, product architecture overview, supply chain architecture overview/generic production flow, and the multi-level roadmap.

In another global company developing and producing medical devices, another subset of the Architecture Framework has been used in the two largest development projects. The overall goal is to coordinate the development of the production equipment with the development of the product program whilst making sure complexity of the product offerings does not increase exponentially. Historically, the product assortment has grown in small steps triggering a need for constant rebuilding and expansion of the production setup followed by high cost and unacceptable downtime of production. The Architecture Framework served to explore the boundaries of the product offerings during the early development phase to ensure that the preparation level of design and production equipment was optimized for quick responsiveness where needed and de-scoped where the inclusion would have resulted in over-investments and loss of productivity for a large part of the production setup. The result has been that more products variants are developed simultaneously than ever before, including the very important preparation of the production architecture. This has resulted in a time-to-market reduction for the product variants that would normally have been postponed for a later development project, as well as an improved overall R&D efficiency due to the wider bandwidth of the product program to be derived from the program architecture.
Views included: Customer view 2.0, module consequence evaluation chart, product architecture overview, preparation level evaluation chart (partly), generic production flow, production consequence evaluation chart, and the multi-level roadmap.

In a third global company developing and producing machinery for industrial, commercial, and consumer applications, the Architecture Framework has been used and tested in two larger development projects. The overall goal has been to move in the direction of multiple market launches that are derived from a program architecture over a period of 8-10 years, instead of renewing the entire portfolio every 3-4 years with large investments and R&D resource spend as a consequence. The Architecture Framework served to challenge the initially proposed product structure to see how robust it is to embrace the features that are planned to be introduced beyond the first product launch. Together with an overview of the program layout and an application/requirement overview the framework succeeded in bridging these to the cost of system performance and module performance scaling chart. This was done to ensure that main performance properties were directly targeted the use applications where they provide the most value – in the most cost-effective way. To exemplify, a number of different

solutions existed to provide the Watts needed from the customer's point of view. The framework made it possible to identify the most cost-effective option, which at the same time made it possible to scale the performance for all the product variants in scope, without compromising the balanced achievement of performance- and cost points. Overall, the result of the framework being applied is a significant reduction in time-to-market for all the features that would traditionally have been de-scoped from the development project and postponed for implementation later on. To be exact, the early phases of a similar product development had a standard duration between 12-18 months, which was now reduced to 6 months. It was evaluated in formal workshop that the later development phases including production preparation etc. would not change dramatically in duration. However, the program architecture was evaluated to improve the precision of the project duration by avoiding downstream non-value adding iterations.
Views included: Program layout, application/requirement overview, Customer view 2.0, module consequence evaluation chart, product architecture overview, module performance scaling chart, cost of system performance, and the roadmap.

In a fourth global company developing and producing electronics for businesses and consumers, the Architecture Framework has been used in two large development projects. The overall goal in this context is to be able to launch 50% more products using the same amount of R&D resources. The key to achieve this is to move from the development of single or very small families of products to develop entire product programs. To be able to do so, the identification and definition of a program architecture is key in order to ensure that multiple product launches not just become individual development projects that are carried out in sequence without improving time-to-market or R&D resource utilization. The Architecture Framework enabled the structured overview of features/options to ensure adequate product differentiation from a market point of view without an unnecessary increase of modules and interfaces across product variants. The result was the parallel development of almost twice as many product variants than previous projects had managed within a comparable time horizon. To be exact, five product variants were included in the program architecture where a standard scope within the company would be one to three. Another effect was that the early phases preceding the stage-gate development phases was reduced by more than 50%, based on detailed measurements conducted by the company's program office.
Views included: Customer view 2.0, module performance scaling chart, product architecture overview, and the multi-level roadmap.

7. CONCLUSION

The main result is the Architecture Framework for Product Family Master Plan. The Architecture Framework contains a set of modeling techniques to provide views that enable the identification and definition of a program architecture during the early phases of a product development project. The framework is unique in the sense that it includes not only constitutional aspects of a program architecture, but also includes behavioral aspects while enabling the optimization of the program architecture through a number of evaluation models. The framework has been applied in a number of cases studies, resulting in four different global companies indicating a significant improvement in time-to-market for product launches derived from the program architecture. This is enabled through the improved utilization of R&D resources by the means of enhancing the preparation level or ability to launch of the product program. This is done by preparing the program architecture for the second, third and fourth product launches following the first product launch. The preparation level is achieved through the coordinated development of an architecture of the market, product, and production to enable a long-term perspective on product program decision-making during the early phases of a development project. Further works and implications of the framework include the coordination with quantification of complexity costs to ensure a measurable impact on not only innovation related benefits but also complexity cost reduction.

8. REFERENCES

Andreasen, M.M., 1980. Machine Design Methods based on a Systematic Approach - contribution to a design theory, Ph.d. thesis. (Danish) Department of Machine Design, Lund University, Sweden

Andreasen, M M, Hein, L. (Eds.), (1987). Integrated Product Development. Springer, Berlin

Andreasen, M.M., Olesen, J., (1990). The Concept of Dispositions. Journal of Engineering Design. 1, 17-36

Campagnolo, D., Camuffo, A., (2010). The Concept of Modularity in Management Studies: A Literature Review. International Journal of Management Reviews. 12, 259-283

Chesbrough, H (Ed.), (2013). Open business models: How to thrive in the new innovation landscape. Harvard Business Press

Cooper, R., (1992). Formal organization as representation: remote control, displacement and abbreviation. Rethinking Organization: new directions in organization theory and analysis, 254–72

Du, X., Jiao, J., Tseng, M.M., (2001). Architecture of Product Family: Fundamentals and Methodology. Concurrent Engineering. 9, 309-325

Ericsson, A, Erixon, G. (Eds.), (1999). Controlling Design Variants - Modular Product Platforms. Society of Manufacturing Engineers, Dearborn, Michigan

Fine, C (Ed.), (1998). Clockspeed - Winning Industry Control in the Age of Temporary Advantage. Perseus Books, Reading, MA

Fixson, S.K., (2005). Product Architecture Assessment: A Tool to Link Product, Process, and Supply Chain Design Decisions. Journal of Operations Management. 23, 345-369

Gonzalez-Zugasti, J., Otto, K.N., Baker, J.D., (2000). A Method for Architecting Product Platforms. Research in Engineering Design. 12, 61-72

Guo, F., Gershenson, J.K., (2007). Discovering Relationships Between Modularity and Cost. Journal of Intelligent Manufacturing. 18, 143-157

Hansen, C.L., Mortensen, N.H., Hvam, L., (2012). On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market. 12th International Design Conference - Design 2012

Harlou, U., 2006. Developing Product Families Based on Architectures - Contribution to a Theory of Product Families. Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby

Herrmann, J.W., Cooper, J., Gupta, S.K., Hayes, C.C., Ishii, K., Kazmer, D., Sandborn, P.A., Wood, W.H., (2004). New Directions in Design for Manufacturing. IDETC/CIE, 1-9

Huang, G.Q., Zhang, X.Y., Liang, L., (2005). Towards Integrated Optimal Configuration of Platform Products, Manufacturing Processes, and Supply Chains. Journal of Operations Management. 23, 267-290

Hultink, E.J., Griffin, A., Hart, S., Robben, H.S., (1997). Industrial new product launch strategies and product development performance. J. Prod. Innovation Manage. 14, 243-257

Jiao, J., Simpson, T.W., Siddique, Z., (2007). Product Family Design and Platform-based Product Development: A State-of-the-Art Review. Journal of Intelligent Manufacturing. 18, 5-29

Jiao, J., Tseng, M.M., (1999). A Methodology of Developing Product Family Architecture for Mass Customization. Journal of Intelligent Manufacturing. 10, 3-20

Johannesson, H., (2013). Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms, in: Simpson, T.W., Jiao, J.(., Siddique, Z., Höltta-Otto, K. (Eds.), . Springer New York, pp. 119-146

Kester, L., Hultink, E.J., Griffin, A., (2013). An Empirical Exploration of the Antecedents and Outcomes of NPD Portfolio Success

Ko, Y., Kuo, P., (2010). Modeling Concurrent Design Method for Product Variety. Concurrent Engineering. 18, 207-217

Krause, D., Eilmus, S., Jonas, H., (2013). Developing Modular Product Families with Perspectives for the Product Program, in: Smart Product Engineering. Springer, pp. 543-552

Lindemann, U., Maurer, M., (2007). Facing Multi-domain Complexity in Product Development. The Future of Product Development. 6, 351-361

Lindemann, U, Maurer, M., Braun, T. (Eds.), (2009). Structural Complexity Management: An Approach for the Field of Product Design. Springer, Berlin

MacDuffie, J.P., (2013). Modularity-as-Property, Modularization-as-Process, and 'Modularity'-as-Frame: Lessons from Product Architecture Initiatives in the Global Automotive Industry. Global Strategy Journal. 3, 8-40

Manohar, K., Abell, J., Beiter, K., Gonzales, D., (2010). Analytical Framework to Define Product Architecture, 285-291

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

Mortensen, N.H., Hansen, C.L., Hvam, L., Andreasen, M.M., (2011). Proactive modeling of market, product and production architectures. Proceedings of the 18th International Conference on Engineering Design : Impacting Society through Engineering Design, 133-144

Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P., Hansen, C.L., Frobenius, S., (2010). Making Product Customization Profitable. International journal of industrial engineering. 17, 25-35

Mortensen, N.H., Pedersen, R., Kvist, M., (2008). Modeling and Visualizing Modular Product Architectures for Mass Customization. International Journal of Mass Customisation. 2, 216-239

Olesen, J., 1992. Concurrent Development in Manufacturing - based on dispositional mechanisms, Ph.d. thesis. Department of Engineering Design, Technical University of Denmark, Kgs. Lyngby

Otto, K.N., Wood, K.L., (1998). Product Evolution: A Reverse Engineering and Redesign Methodology. Research in Engineering Design. 10, 226-243

Pahl, G, Beitz, W., Wallace, K. (Eds.), (1996). Engineering Design - A systematic approach. Springer, London

Panda, T.K., Mohanty, P.K., (2013). Adoption of Supply Chain Management Strategies as a Response to Bull Whip Effect: From the Perspective of Indian Retailers. IUP Journal of Supply Chain Management. 10

Pimmler, T.U., Eppinger, S.D., (1994). Integration Analysis of Product Decompositions. American Society of Mechanical Engineers, Design Engineering Division. 68, 343-351

Simpson, (2004). Product platform design and customization: status and promise. (AI EDAM) Artif. Intell. Eng. Des. Anal. Manuf. (UK) Artificial intelligence for engineering design, analysis and manufacturing. 18, 3-20

Simpson, T W, Jiao, J., Siddique, Z., Hölttä-Otto, K. (Eds.), (2014). Advances in Product Family and Product Platform Design: Methods & Applications. Springer New York

Skinner, W (Ed.), (1978). Manufacturing in the Corporate Strategy. Wiley, New York, N.Y.

Star, S.L., Griesemer, J.R., (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. Social studies of science. 19

Stone, R.B., Wood, K.L., Crawford, R.H., (2000). Using Quantitative Functional Models to Develop Product Architectures. Design Studies. 21, 239-260

Ulrich, K., (1995). The Role of Product Architecture in the Manufacturing Firm. Research Policy. 24, 419-440

Ulrich, K, Eppinger, S. (Eds.), (2000). Product Design and Development. McGraw-Hill, Boston

6.5 Paper E

IDENTIFICATION OF A SCALABLE ARCHITECTURE FOR CUSTOMIZATION OF COMPLEX PARTS

Christian Lindschou Hansen[1*], **Hans Peter Lomholt Bruun**[1], **Niels Henrik Mortensen**[1] & **Lars Hvam**[2]

Department of Mechanical Engineering[1]
Product Architecture Group
The Section of Engineering Design & Product Development
Technical University of Denmark
Building 426
DK-2800 Kgs. Lyngby, Denmark
Email: chrlh@mek.dtu.dk, hplb@mek.dtu.dk, nhmo@mek.dtu.dk
Telephone of corresponding author: +45 4046 4233

Department of Management Engineering[2]
Center for Product Modeling
Technical University of Denmark
Building 424
DK-2800 Kgs. Lyngby, Denmark
Email: lhv@man.dtu.dk

** corresponding author*

Abstract

Many Original Equipment Manufacturers (OEMs) are experiencing an ever increasing pressure on their ability to develop solutions at a faster pace and at competitive prices. In particular, OEMs developing and manufacturing high-performance complex parts with highly integrated product structures are struggling with time-to-market and unpredictable quality levels. Orders are often fulfilled in an engineer-to-order workflow and very little commonality exists between solutions across the product program. These OEMs also experience significant challenges in applying modularization to their product programs as a means to overcome the challenge, due to the reason that highly integrated product structures of complex parts are not easily modularized using traditional methods for modularization. In such cases resulting compromises on performance and cost are most often difficult to unite in a competitive product. This paper presents a framework that enables such companies to overcome these challenges by identifying an architecture of the product and production setup. The architectures enable the companies to scale their solutions and production setup in a profitable way, and at the same time maintain a sufficient degree of commonality to significantly improve time-to-market, R&D resource utilization and the level of quality.

Significance: Many companies experience problems with applying modularization to highly integrated product structures of complex parts. This paper suggests a framework for identifying modularity even beyond the traditional physical product interfaces by an architecture that allow these companies to harvest the benefits of modularization without compromising functionality, performance, or cost.

Keywords: Product architecture, product platform, product customization, scalable architecture, product complexity

1 Introduction

Many Original Equipment Manufacturers (OEMs) experience an increasing pressure on their ability to develop solutions faster and at lower costs. It is not unusual that large companies expose their OEMs to cost cutting strategies forcing them to cut 5-10% of cost every year (Zoia, 2013; Bickerstaffe, 2012). This leaves OEMs with no choice but to innovate. Only through developing new solutions with improved performance can they postpone the pending price drops and remain profitable. As many businesses also become more and more project-oriented, large companies are minimizing their risks and become reluctant to co-finance R&D activities at their OEMs, while at the same time expecting their OEMs to supply new solutions at a faster pace. This is a tendency which has increased in the wake of the global financial crisis.

These changed circumstances put the OEMs under pressure, requiring them to increase their R&D efficiency to improve time-to-market by providing new competitive solutions faster; and at the same time cut costs by improving product quality and productivity in production. The globalization of most large companies has also forced OEMs to act in the global stage, making the need for local variants increase, while the fluctuation of demand between the different variants makes production planning more difficult as the production volume of each product variant decreases.

Requirements on OEMs	
Decrease	*Increase*
Time-to-market CAPEX (Capital Expenditure) Volume per product variant	R&D Efficiency Production efficiency Quality level No. of product variants

Table 1 – Conflicting requirements on OEMs

The requirements listed in Table 1 are conflicting. And they apply challenge an OEMs agility of supplying new solutions, whilst at the same time making it very costly to expand the current portfolio using traditional approaches. Developing new solutions in a traditional engineer-to-order workflow would severely compromise R&D efficiency while imposing negative consequences on production ramp-up times and product quality, as every solution is often new from an R&D and production point of view. At the same time the OEM customers require that OEMs push the envelope of the products' performance to enable them to achieve higher efficiency and improve their own offerings.

Traditionally the notion has been that applying modularization can make it possible to develop a modular architecture for the product program (Meyer and Lehnerd, 1997; Ulrich, 1995). Modularization when appropriately applied could serve as a means to provide the variety needed from a customer point of view and at the same time reuse sub-solutions across different products to improve time-to-market, and maintain predictable product quality. However, many companies do not succeed in this, as modules are not easy to identify in products where key functionality is highly integrated and distributed across the product structure (Hölttä-Otto and de Weck, 2007). Examples of these types of complex parts include e.g. rotor blades, complex manifolds and engine parts, hulls etc. that in addition contain a high degree of engineering from several disciplines as fluid mechanical engineering, solid mechanical engineering, process engineering and chemical engineering. In these cases, the compromises on performance often become too significant following traditional modularization approaches (Guo and Gershenson, 2007; Gershenson et al., 2003). In the development of highly performance oriented critical components – no extra 3-5% of material/weight can be added to the products. Such over-engineering is most often not an option.

This paper is based on the assumption that the definition of an architecture can enable the OEMs to overcome the challenging situation described above in Table 1. So far, traditional modularization has played an important role in architecture-based product development (Ericsson and Erixon, 1999). But as traditional modularization takes its starting point in functional decoupling in order to isolate functionality in modules (Jiao and Tseng, 1999), there is a need for a new approach applicable to products where functionality is highly integrated and distributed across the product structure.

This paper presents a framework seeking to expand the recent body of knowledge within modularization and architecture-based development of product programs by including the situation of OEMs supplying engineer-to-order solutions with integrated functionality distributed across the product structure of complex parts. Thus, the framework disclaims the current notion of modularization and architectures being a compromise to achieving competitive performance in highly integrated designs. The framework contains a set of coherent models and a stepwise approach.

This paper will continue by elaborating the motivation and requirements for the framework. This will be followed by a review of the state-of-the-art methodology and a presentation of the proposed framework. The framework has recently been applied in an industrial case study with promising results, which are also presented here. The paper is concluded by a discussion, reflection and a conclusion.

2 Challenges and barriers

In the experience of the authors no OEM management disagrees with facing challenges in meeting the requirements mentioned in Table 1. Many of these challenges are closely related to the current ways OEMs are developing unique solutions to every customer. The observations below are based on a collection of interviews with management of OEM companies:

Customers are dictating solutions: As many OEMs do not proactively show their customers which solutions they would prefer, the natural result is that OEMs end up providing solutions that are unique to every individual customer. In the pursuit for customer satisfaction many OEMs enter a sales dialogue with a mindset of accepting a level of customization close to 100% – even regarding sub-solutions that are not critical to the overall performance of the product.

No proactive go-to-market approach: Many OEMs satisfy themselves with their upstream position in the value chain as being a rather passive supplier in the market place, not recognizing the need for an active marketing effort to analyze market developments and predict which particular segments are growing. Therefore they tend to fall into a reactive role resulting in lack of responsiveness towards technological trends and changing requirements from customers.

Very limited development outside customer projects: Many OEMs do not start development before the customer has accepted to cooperate. This leaves customer projects being notoriously behind schedule and leaves little-to-none opportunity for including forward-looking development work in customer projects.

Customer projects are fulfilled with zero outlook: The natural consequences of the reactive market approach and customer dictated solutions are that customer projects often fail to look beyond their first delivery. The focus on the first delivery results in solutions that are not prepared for upgrades or predisposed for future variants.

Solutions are developed in individual work streams: In the attempt of becoming customer focused, OEMs organize development teams in individual work streams, with only little or no coordination between them. The result is often lack of solution overview and sub-solutions that vary from customer project to customer project, making it difficult to harvest benefits of common solutions across customer projects while prolonging time-to-market.

Complex interplay between specialized engineering disciplines: The high performance products in scope here require very specialized engineering design work from different disciplines. Even though every engineering discipline has a separate optimization task closely related to the unique customer requirements many dependencies exist between them, making it difficult for individual disciplines to predict and see through their influence to other disciplines.

R&D resources are tied up to individual customer projects: OEMs tend to tie a relatively large share of their R&D resources into specific customer projects, when customers are willing to co-finance development costs. However, “renting out” R&D resources for e.g. EUR70-80 per hour makes a very low return on investment compared to investing in R&D resources developing solutions that could be common for a range of customers. The issue becomes particularly crucial when customers exit the projects before time and the OEM is left behind with a customer specific solution and zero orders.

Lack of task definition: It is the experience of the authors that the challenges mentioned above is a natural consequence of the way OEMs orchestrate their engineering resources and define their tasks and priorities. There are often very few or no cross-functional initiatives that aim to overcome these challenges as they span all the way from initial customer dialogue to the engineering execution and project delivery. Moreover customer projects are often kept isolated and 100% focused on customer dictated requirements making it difficult to prioritize efforts in order to change the condition mentioned above.

These challenges end up becoming barriers for OEMs to fulfill the requirements listed in Table 1.

3 Requirements for the framework

To overcome the challenges and barriers presented in the previous section, there is a need for OEMs to address their development of customer specific solutions in a different manner. As the problem reaches all the way from early sales dialogue to the delivery of engineering solutions, it is not sufficient to make small and local changes in order to turn the situation around – there is a need for a more profound change. A change that needs the support of a framework, in order to

- Identify an architecture for the product and production setup that allows for profitable customization of complex customer specific solutions with highly integrated product structures by improving the utilization of R&D resources and improving time-to-market.

In order to do so, the framework should enable OEMs to

- Scope an architecture from a market point of view. This includes focusing upon which segments and applications with which performance steps to cover and which not to cover. This again determines the market envelope i.e. the collective of the preferred offered product variants and their desired performance steps.
- Identify the defining design characteristics and properties based on primary market parameters. This includes the scaling principles for design characteristics and performance properties.
- Identify an architecture that utilizes R&D resources to provide value instead of solving similar requirements for every customer project as repetitive work. This includes decoupling of work tasks to enable different disciplines to work in parallel.
- Identify an architecture that allows for scaling of sub-solutions to allow for reuse between customer projects.
- Identify an architecture with product scaling principles that are coordinated with the production scaling principles with respect to geometry and volume.
- Identify an architecture that can serve as foundation for deriving several future product variants beyond the first delivery

With these requirements literature has been reviewed for contributions that address the challenges described and fulfill the requirements listed here.

4 State-of-the-art

This section covers significant contributions in literature to: The modeling of architectures, the support for developing modular product families, and methods for improving the development process in terms of reducing lead time.

Architectures based on design process theory: According to (Andreasen and Hein, 1987) the development process can be described in terms of single models on four levels: product planning, product development, product synthesis and problem solving. The design process theory is based on descriptive and prescriptive models from (Pahl et al., 1996; Hubka and Eder, 1988). Product planning is related to activities where decisions regarding introduction of new products and phasing out existing products are made. Concerning application of architectures and platforms, it will have impact on all levels above, i.e. product planning, product development, product synthesis and problem solving. In order to succeed with a platform it has to be fitted to the nature of product development, meaning that solutions are determined gradually, and platforms have to include certain flexibility in order to match different design projects and product variety.

The concept of Concurrent Engineering (CE) is a parallel approach, replacing the linear process of serial engineering. CE is intended to encourage the product developers, from the start, to consider the total job (Prasad, 1999; Prasad, 1996). The concept of CE is very much linked to the concepts of architecture based development. In a modular architecture, there is a division of labor between architects who first split a product into modules, and those who work within the parameters of a specific module. The latter group needs to know only about the specific module and the design rules which ensure that the module can be integrated into the larger system. Modules can then be developed in parallel, which again lowers lead time in development. A prerequisite is that architects possess the requisite knowledge of parameter and task interdependencies of the whole product.

The concept of Architecture for Product Family (APF) is introduced as a conceptual structure, proposing logics for the synthesis of product families (Du et al., 2001; Jiao and Tseng, 1999) . The Generic Product Structure (GPS) is then proposed as the platform for tailoring products to individual customer needs. (Ko and Kuo, 2010) presents another systematic method for concurrent development of product families, by combining QFD-based methods with quantified DSM-techniques and morphology analysis to visualize concepts.

Product Family Master Plan (PFMP): The Product Family Master Plan (Mortensen et al., 2010; Harlou, 2006), describes a product assortment from three points of view: Customer, engineering and part view, equivalent to the partitioning in market, product and production domain of the Integrated Product Development framework (Andreasen and Hein, 1987). Each of the views is causally linked meaning that certain types of traceability can be described. The relation between customer view and engineering view describes how certain customer features are realized by means of certain functional units. The relation between engineering view and part view explains how functionality is realized by means of physical parts and sub-assemblies. Reading the PFMP from the part view to the engineering view explains how a certain part contributes to delivering functionality to the products. From the engineering view to customer view the relation describes how functional units deliver customer features and, value to the customer. Another important aspect of the PFMP is that it enables a professional dialogue between three very important stakeholders, namely sales, engineering and production. There must also exist a professional media for communication and decisions concerning the exact scope and content of possible platforms.

Modular Function Deployment (MFD): The Modular Function Deployment (Ericsson and Erixon, 1999) builds on the methodology of the Quality Function Deployment (QFD) and on the formulation of eight so-called module drivers. The purpose of MFD is to enable cross functional teams (including mainly marketing, development and production) to create a mapping from the physical structure of the products within a family to the functional structure of those products and to ensure that the functional structure corresponds to the demands of the customers. Modular Function Deployment method consists of five consecutive steps. Customer requirements are mapped to functional criteria and subsystem design characteristics and subsequently form a physical design in which a modular architecture supports a carefully selected set of modularization incentives called module drivers.

Design Structure Matrix (DSM): This approach takes a starting point in the decomposition of a product into components/systems and an identification of interfaces/relations among these (Hölttä-Otto and de Weck, 2007; Pimmler and Eppinger, 1994). By the use of algorithms, it is possible to encapsulate components into modules or chunks that are closely related to each other from an interaction point of view (Steward, 1981). This process is referred to as clustering. The outcome of a DSM is a proposal for a future modular product architecture.

The function-based design methods (Function structures) are characterized by establishing either a function model (Otto and Wood, 2001; Pahl et al., 1996) or the schematics of the product (Ulrich and Eppinger, 2000). Both approaches have a visual representation as an outcome. The function structure describes the flow of material, data, and energy through sub-functions of the product using a set of rules (e.g. the rules that are referred to as the functional basis which basically is a common language to describe functional elements). The schematic of the product is somewhat similar to the function model. But where the function model describes the product using functional elements, the schematics can describe both functional and physical elements — whichever is the most meaningful for the purpose of the representation. The functional structure forms the basis for several different approaches to design or re-design the products.

The German school of Variant Management provides a number of methods and techniques to optimize the design of variance in product families (Krause et al., 2013). The methods and techniques form an integrated approach, which aims to reduce internal variety of product programs. The overall idea of the approach is that non-value adding variety of the product program is both constituted by elements belonging to product variety and process variety. In order to reduce internal non-value adding variety of the product program, one has to address technical-functional and product-strategic module drivers along the product life cycle phases, and to redesign components to enable a modular (variety optimized) product structure.

Configurable Component framework: The configurable component concept (CC) is a means of representing systems and their subsystems using a generic building block, the configurable component (Claesson, 2006). The original purpose of the concept is to handle data, information and knowledge sharing, as well as managing the conflict between commonality and reuse, while having the ability to represent variant-rich and complex products and, more generally, entire product platforms. The CC concept 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 throughout the development process. The CC framework has been implemented and tested in the automotive and aerospace industry. The framework has been enlarged to cover an approach for integrating modeling of products and production systems (Gedell and Johannesson, 2013; Levandowski et al., 2013).

Impact of product configuration in engineering oriented companies: Studies have shown that engineering oriented companies (companies in which each customer order requires some engineering work) can gain significant reduction of lead time in quotation and production by implementing product configurators (Haug et al., 2011). Product modularization and configurations are used to structure and model the product assortment in order to configure a customer tailored product unambiguously. However, the creation and use of configurators is often a risky and highly time-consuming project. Thus, although for example for a 90 percent reduction of lead time and man-hours achieved, this may still be an unprofitable project if the costs of achieving this are too high.

4.1 Gap

The contributions definitely all play an important role in identifying architectures. However, very few contributions have dealt with the definition of architecture initiatives for highly integrated products or complex parts. For these product types it is often not possible to consider integration or balancing of different modules, because modules (in a classical understanding) cannot be decoupled from the rest of the integrated product structure. It is simply not possible to identify such. Integrated products or complex parts can be characterized as functional feature-based products in which the customer's perceived value in the products is based on properties such as peak performance and efficiency in operation. To optimize such functionality of a product family, it is necessary to consider compromises between the product variants (total systems) instead of compromises between modules (sub-systems), because the performance of the technical system is dependent on the balancing of design characteristics between product variants. The current frameworks, methods and models proposing to support architecture initiatives, do not enable companies to overcome these challenges. The gap in prior contributions is centered on the task of identifying flexible and scalable architectures, for highly integrated products, for the product and production setup. Yet, the theoretical basis mentioned in section 4 provides a thorough basis for deriving a framework integrating the PFMP-based methodology of concurrent design of market, product and production aspects merged with a function-oriented modeling of performance properties to identify coherent feature, performance and production scalability.

5 Research methodology

The framework presented in Figure 1 is the result of experience derived from several previous case studies conducted with OEMs producing high performance mechanical products for a relatively small amount of large customers. These studies have matured the framework till its current state and subsequently it has been tested in yet another case study with very promising results.

According to (Joergensen, 1992), research is both problem and theory based. The problems in the industrial practice is described in section *Challenges and barriers* and which is based on several interviews with managers and decision makers in European engineering companies experiencing these problems. Therefore, many descriptive research activities lie ahead of this framework presentation, which can be characterized as being prescriptive.

5.1 Type of inquiry

Different types of inquiries were used while engaging in the practical setting. During the analysis phase of the study, the inquiries were rather exploratory and diagnostically based, helping the researchers to understand the situation and assess the applicability of the framework. Moving on to the synthesis phase, the inquiries changed to being more of confronting in character and directly prescriptive. The last type challenges the company to see their products from a new perspective, and was absolutely necessary in order to make them adopt the framework and ensure a successful intervention.

5.2 Visualization as working method

As the research aims to bridge information from sales and marketing with engineering and production development, there is a need to create a *boundary object* enabling the different competences to interact, exchange ideas, and understand each other's work challenges (Latour, 1986). Thus visualization has been used to create such a boundary object to facilitate collective alignment among sales/marketing, engineering and production professionals. From the early stages of the project a concept architecture has been illustrated on A0 sized posters, allowing professionals with different backgrounds to gather around a large poster and make review meetings efficient and by taking advantage of the optical consistency such a visualization represents. This approach enables participants to lay aside their daily working habits and see the challenges in the project as being of the 'same type'.

6 Framework

The illustration presented in Figure 1 represents the framework for developing a scalable architecture including a step-wise approach. It is not the intention to present a complete framework, but instead emphasis has been put on the elements of crucial importance when an OEM wishes to:

- Move away from a dedicated engineer-to-order workflow where unique solutions feature in every customer project
- Explore their design envelope to investigate the potential for generating solutions with lower lead time
- Develop an architecture to enable faster development of new and competitive solutions
- Improve the efficiency and effectiveness of R&D resources
- Enhance the preparation level towards generating future derivative variants

The framework follows the classic partitioning in a market, product and production domain known from (Andreasen and Hein, 1987). The 9 steps represent sequenced excerpts from the framework that was discussed, reviewed and documented on large A0 format posters to keep a coherent overview of the architecture’s status.

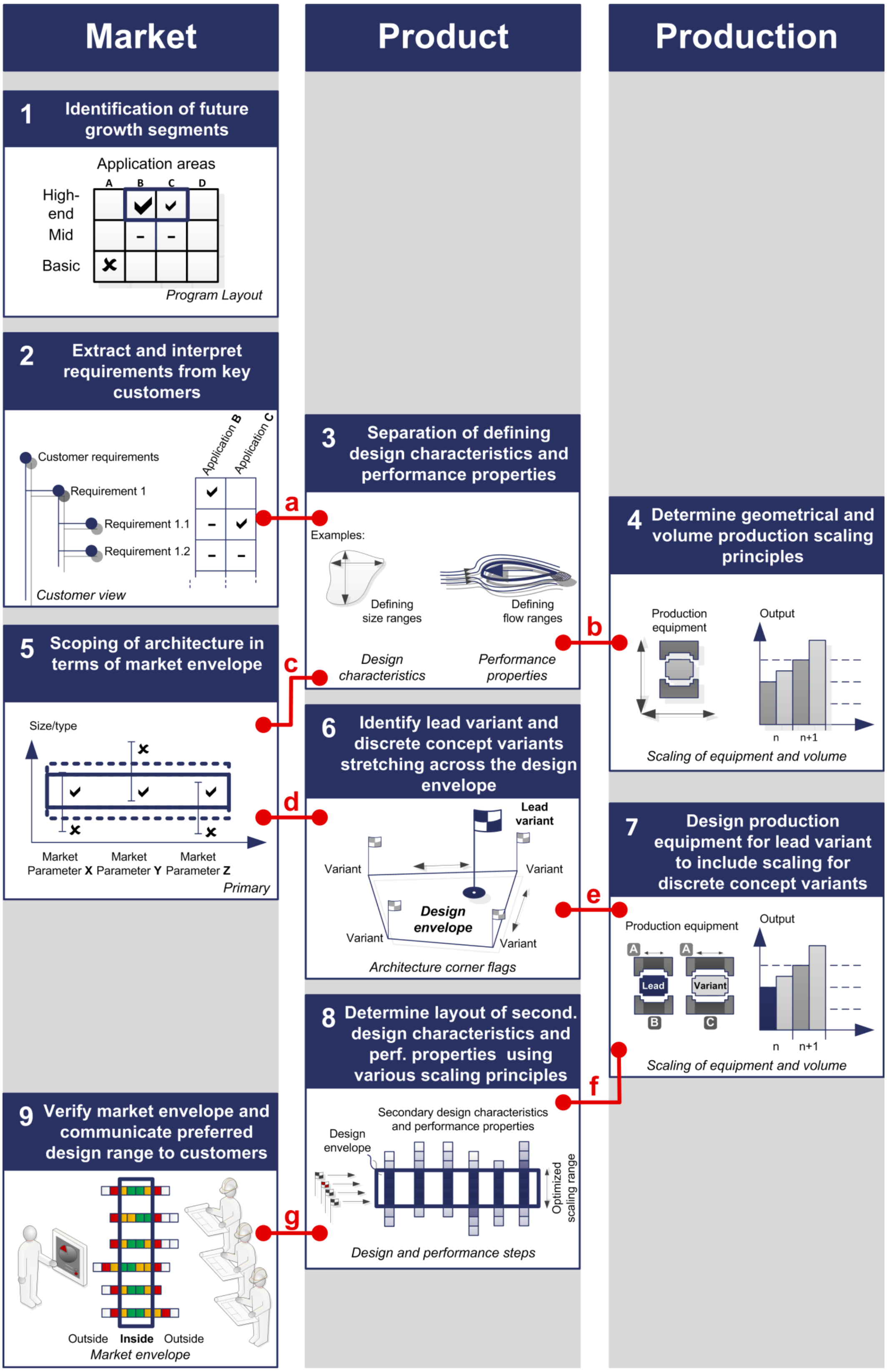


Figure 1 – Framework including approach

1. Identification of future growth segments

Developing an architecture which is prepared for the future, requires a much more systematic approach to analyzing and interpreting the growth of the most favorable market segments than traditional OEMs are used to. It is not necessarily good to be 100% customer driven in the sense that customers might not include the OEM in their 3-5 year product roadmaps thus leaving the OEM behind with shorter time to react. Therefore it is necessary to use for example the Program Layout to outline the growth in all relevant business areas across different tiers from high-end to basic (Hansen et al., 2012). This can form a basis for identifying which market segments to focus on and isolate the application areas of relevance.

2. **Extract and interpret requirements from key customers**

From the analysis of the future growth segments, the next step is to select a few key customers within these segments and extract and interpret their requirements. The Customer View can serve to support the systematic mapping of these requirements based on for example interviews with senior representatives within technical sales (Harlou, 2006), and categorize these requirements dependent on which application they are applicable for.

3. **Separation of defining design characteristics and performance properties**

The next step is to separate those design characteristics and performance properties that answer to the primary market parameters. These are derived as parameters fulfilling the most important customer requirements – e.g. efficiency, load limits or other characteristics and properties. This is a very critical step, as it is necessary to initiate the development of the architecture by solely focusing on those fulfilling the few requirements that are capable of positioning the products in the market place.

Link a: The interpretation of the most important application specific requirements into the few design characteristics and performance properties that can fulfill these.

4. **Determine geometrical and volume production scaling principles**

Using the direct input of the defining design characteristics (e.g. primary geometries) the scaling principles for production can now be determined. As it is critical to achieve scaling towards future variants and upgrades, this must be determined in coordination with the geometrical possibilities and limitations outlined by the defining design characteristics.

Link b: The physical scaling of production equipment is coordinated directly with the defining design characteristics and scaling of volume is taken into account.

5. **Scoping of architecture in terms of market envelope**

As it is impossible to satisfy all primary market parameters in their full range, it is necessary to scope the architecture in terms of which will be boundaries of the market parameters seen from the customer's point of view. The fulfillment of these parameters will often be subject to trade-offs, as the fulfillment of one parameter can be increased by compromising the fulfillment of another.

Link c: The market parameters might be direct market oriented translations of certain design characteristics (e.g. size) and performance properties (e.g. thrust, power, efficiency).

6. **Identify lead variant and discrete concept variants stretching across the design envelope**

Identifying the lead product variant, meaning the first product variant of the new product family, is the next step. The identification of this would normally be the result of close dialogue with a lead customer. While specifying the lead variant it is of crucial importance to specify a number of additional concept variants. These concept variants are not going to be completely designed in detail just yet. They serve as important instantiations of the architecture in order to investigate scaling principles of design characteristics between the different variants, including the lead variant. This is in order to achieve appropriately balanced performance steps of the most critical properties.

The sum of these architecture variants spans the total design envelope of the architecture.

Link d: The architecture variants are closely linked to the market parameters in the way that the design envelope matches a certain market envelope, where individual market parameters are covered within a certain range.

7. **Design production equipment for lead variant to include scaling for discrete concept variants**

In terms of production the basic scaling principles are already determined (step 4), but taking the starting point in the lead variant the actual equipment can now be designed and specified. The discrete concept variants are taken directly into account deciding the geometrical scaling principles as it is now possible to prepare production equipment (moulds, fixtures etc.) for the exact design characteristics of these. However, no capital is invested covering the discrete concept variants – only preparation activities to maximize future reuse of production equipment for the lead variant

Link e: Production equipment for the lead variant is prepared for scalability towards the design envelope represented by the discrete concept variants.

8. **Determine layout of secondary design characteristics and performance properties using various scaling principles**

Until now, the design only contains the defining design characteristics and performance properties. The secondary ones are now to be balanced within the design envelope to make a good fit in balanced design and performance steps. These secondary design characteristics and performance properties describe the sub-solutions. They do not interfere with primary performance, as the task is to optimize the whole architecture and product family. There is no value in varying these sub-solutions unnecessarily across the architecture.

Link f: Certain sub-solutions might have close ties to the production equipment and desired future volume scalability. Therefore, it should be ensured sub-solutions are aligned with the critical choices of production technology and volume scaling principles.

9. **Verify market envelope and communicate preferred design range to customers**

To close the loop and harvest benefits of the preparation of the architecture, the next step is to verify the design and resulting market envelope to make sure that it fits with the needs of potential future customers. If this can be verified the next challenge is to communicate the preferred design range to customers. The preferred design range is the market translation of the architecture variants prepared in the lead design and discrete concept designs.

Link g: As the market envelope reflects an optimized scaling range of the designs (“inside” the envelope), it is of critical importance to communicate the preferred range of designs to customers, as there might not be a good product match if the customer falls “outside” the market envelope.

Nomenclature: Green means “inside”, Yellow means “inside with compromises/adaptions”, Red means that larger changes to the architecture has to be made in order to offer a competitive product.

The exact sequence of the 9 steps varies slightly between individual cases, but the sequence presented here was the one used during the case study.

7 Case

The framework was applied in an engineering oriented OEM company serving the global energy industry with performance critical components used for critical energy generation processes. The company has a global production footprint enabling the OEM to serve a number of customers having only regional production facility of their own and to help them expand their market reach to win larger orders on a global scale. The case company is anonymized as a result of competitive reasons and in order to be able to report more interesting details than a public case allows for.

7.1 Situation

In the wake of the global financial crisis the customers of the OEM company have experienced financial problems with financing their energy solutions. Their financial shortcomings put pressure on their OEMs, as they cannot co-finance R&D activities to the extent they have done earlier, and cannot commit themselves to larger production volumes as the business is becoming increasingly project oriented. The OEM is experiencing severe price cuts for their high volume products forcing them to find new growth segments, where their engineering expertise can be valued in terms of a higher price level.

7.2 Framework

The first engagement with the company was an assessment project to evaluate the potential of architecture based product development in the company. As the products of the company cannot be modularized in the traditional sense by visible and physical structural decoupling, certain skepticism was expressed by many stakeholders in the company. However, the result of the assessment project was to apply the framework to a new promising development project, where a few potential large lead customers had shown interest.

As specified in ***Step 1***, the potential customers constituted a future growth segment for the OEM, making it an appropriate business area to apply the framework to.

The next step, ***Step 2***, was to extract and interpret the key requirements from the most likely future customers – including the potential lead customers. The Customer View was used to create an overview of which requirements were common to all customers and which vary between them, and the most important market parameters were isolated.

In ***Step 3***, taking the starting point in the Customer View, the defining design characteristics and performance parameters were separated out in order to create early concept designs that fit directly to the market parameters. Thus it was ensured that the primary requirements of efficiency, load limits and preparation for scalability was taken into account during these very early stages of designing. Fulfilling these was directly associated with the fluid dynamic nature of the products. This made the fluid dynamic trade-offs and design considerations primary for defining design characteristics and performance properties.

Figure 2 exemplifies defining design characteristics and their relations to performance properties for an engine manifold. The performance properties of flow, pressure, and strength are realized in a complex interplay between the inlet/outlet size, the tube wall thickness, the diameter of the tube, and multiple bending radiuses of the tubes.

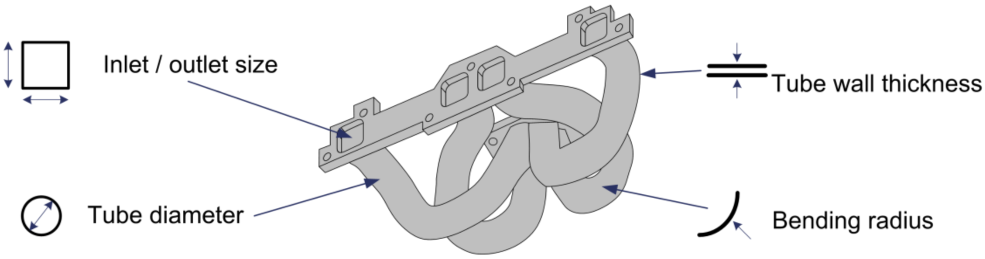


Figure 2 – Identified performance critical design characteristics for an engine manifold

The performance critical design characteristics were used in ***Step 4*** as input to determining the geometrical and performance scaling principles of the production equipment.

Figure 3 exemplifies the concept of scaling principles. The defining design characteristics of the engine manifold have been identified i.e. the defining structure and its attributes based upon desired performance properties. Some geometry is fixed for all variants in the design envelope (red areas) in order to support modularity in the production equipment, while others are flexible (green areas). The performance properties are related to flow, pressure, and strength. In order to meet the requirements to performance properties in the design envelope, the design characteristics have been determined in principles and ranges of scaling i.e. tube diameter, inlet/outlet size, bending radius in tubes, and the tube wall thickness. The relations between design characteristics and performance properties have been analytically verified, and even if no physical parts yet exist, the design is fully scalable within the defined ranges.

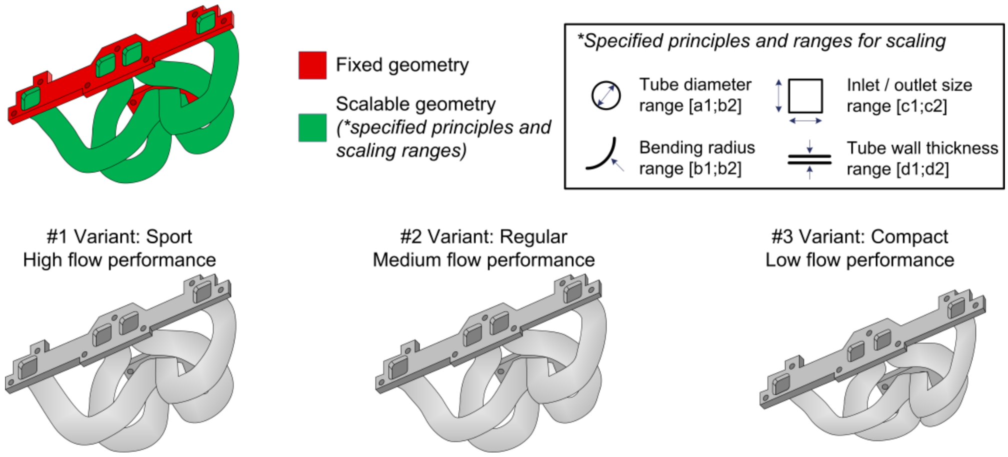


Figure 3 – Engine manifold used as example for illustrating scaling principles and scaling ranges

Modular thinking was incorporated early and applied to the production equipment to ensure future reuse of most the costly and lead time defining production equipment.

The next step, ***Step 5***, was to scope the architecture from a market point of view based on the work from Steps 1-4, as it was not possible to make a one-fits-all design. Clear decisions were made regarding which performance ranges to support, and which not to support. As the customers of the OEM use different technologies, several decisions were made to include support for certain ones while excluding the support of others (e.g. different support system technologies).

In ***Step 6***, the lead variant was identified together with three other discrete concept variants – one variant with lower specification and two with higher specifications. In total now four architecture variants. The lead variant was intended for a large customer. The discrete concept variants were aligned with the preferred scaling principles (from Step 4) and targeted to fulfill market parameters in the surrounding areas of the chosen ranges (from Step 5). Throughout the rest of the architecture design process, the concept variants were used as modifiable instantiations of the architecture to put structure to the design envelope. These concept variants do not limit the future design of variants to match these exact discrete specifications, but they prepare the architecture and the scaling principles applied to fulfill any requests for variants inside the design envelope of the architecture.

Step 7 was the inclusion of production equipment to include the actual lead variant while preparing for the future scaling of the discrete concept variants. Modularization was applied to ensure the preparation towards the required scalability of future variants by decoupling the variable equipment from the costly and lead time defining equipment.

In ***Step 8***, the sub-solutions were included now taking all the secondary design characteristics and performance properties into account. For instance, the steps between sub-solutions do not have to follow the same steps as the four architecture variants – the sharing of sub-solutions could be independent hereof, and in certain cases, only two variants of sub-solutions were chosen for the four variants in scope. An example of this was layers of strengthening material that were completely shared within the common parts of the architecture variants decoupling the variance needed to other more flexible areas of the design.

In ***Step 9***, the architecture including the design and market envelopes was communicated to customers. This approach represented a large shift of paradigm for the OEM company, enabling a much more qualified early dialogue, as the OEM could now enter specific design discussions with customers *before* having started an actual customer project with them. This has so far enabled the OEM to influence and impact design decisions of their customers – a side effect that only very few company representatives believed possible when the project started.

7.3 Results

An extensive validation effort was initiated after the framework was applied. All participants in the architecture design process were interviewed and their best estimate of their resource requirements (in hours) and lead time (in days) was collected. The investigation showed that

- An architecture variant (succeeding the lead variant) can be developed using:
  - R&D resources: From index 100 to 30
- Lead time from:
  - First customer dialogue to finished design: From index 100 to 58
  - Finished design to prototype delivery. From index 100 to 50

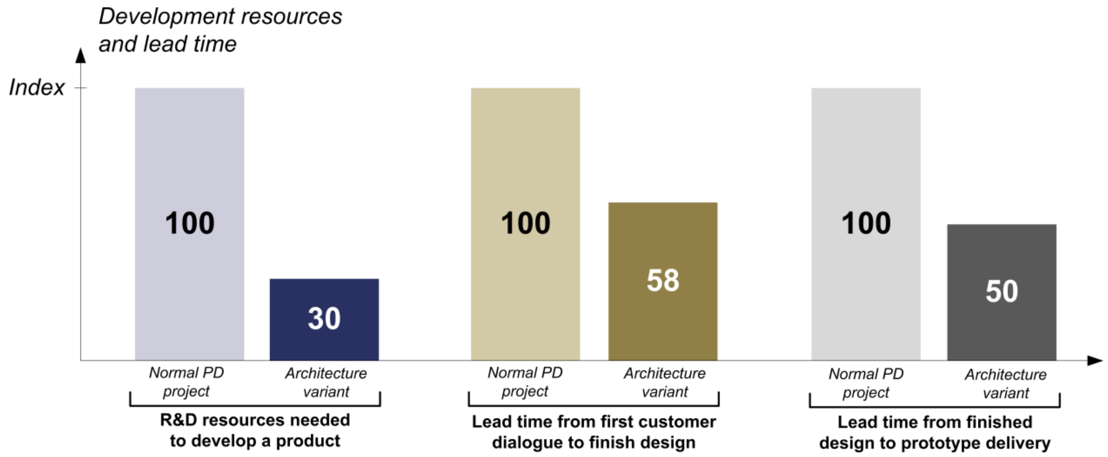


Figure 4 – Savings in design lead time and R&D resources for one variant

The index 100 is reflecting a traditional customer project where a unique and dedicated design is developed from scratch.

This estimate is conservative and includes all resources relevant – including project management itself, while taking critical path and normal project uncertainties into account.

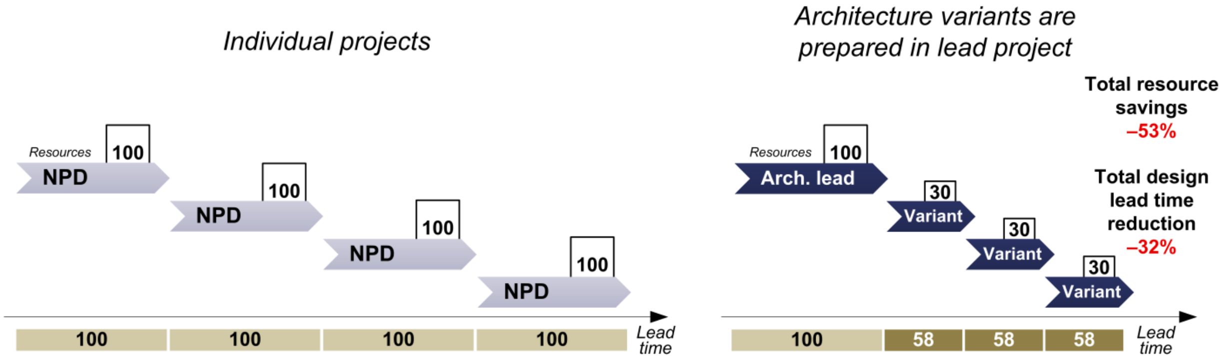


Figure 5 – Total savings in design lead time and resource consumption for four variants

Figure 5 illustrates the savings in design lead time (–32%) and R&D resource consumption (–53%) by using the new architecture approach instead of the company's previous new product development (NPD) process. The architecture project covered four variants and the savings are calculated on this premise. The OEM's ability of supplying new solutions at a faster pace has increased, without increasing R&D resources in the same step. The R&D efficiency has actually improved by reducing the resources needed to develop four variants by as much as 53%. The savings compared to normal projects will moreover increase when more variants are developed inside the already specified design envelope.

The dramatically improved responsiveness towards customer requests inside the architecture has been a game changer for the OEM company. Customers are now even sharing their own roadmaps with the

OEM company in order to ensure closer collaboration and make sure that their product requests falls inside the scope of the further development of the architecture. In other words, the benefits from requesting new products inside the envelope of the architecture are obvious enough to enable the early customer interaction as needed.

Today, the OEM company is working with marketing their architecture in order to harvest the full potential, and recent conclusions are that customers find their approach proactive and constructive. Also the organizational anchoring of the architecture thinking patterns, methods etc. utilized in this project is also a derived activity as are the implications to portfolio management, roadmapping, product and production technology etc.

8 Discussion and reflection

8.1 Case

Reducing time-to-market by almost half and saving 53% on R&D resources is a drastic improvement on current performance. However, limitations of the case results do exist.

The framework only covers the development of a lead variant and preparation for derived architecture variants. The actual execution of architecture variants was estimated through interviews with all relevant stakeholders, but the actual execution of these are in the making at the time of writing. However, the projected realization time, on which basis 70% of the savings were derived, was accepted and found credible by the OEM management. This covers the internal validation of the results.

External validation is more difficult. The researchers have undertaken many architecture projects for OEMs developing mechanical solutions for large customers, which provide certain evidence that this OEM company is comparable to many other OEM companies. Their challenges are similar to the challenges experienced in many other OEM companies. The OEM is a global player, capable of attracting many skilled employees with an annual turnover in the range of EUR 0.5-1billion, and there are no indications that the leap of performance achieved in the case study, could be explained with lack of professionalism or a competence level below average.

The reliability of the results are therefore of course contingent upon many contextual factors of the company where it will be applied. Also the mere presence and attendance of the researchers in kick-off meeting, review meetings and evaluation meetings is impossible to isolate from the results – as of course, a certain competence within the field is necessary in order to create such positive results.

8.2 Theory evaluation

As reported in the case the framework proved useful and solved the challenge of identifying an architecture for the OEM case company. The theoretical gap mentioned in section 4.1 has therefore been challenged by the framework proposed, which is integrating the PFMP-based methodology of concurrent design of market, product and production aspects merged with the function-oriented focus on performance properties to identify coherent features, performance and production scalability. The framework's inclusion of behavioral aspects of architectures across market, product and production domains, namely what the architecture enables the company *to do* in terms of preparation and responsiveness towards future launches (instead of limiting the focus to what the architecture *is*) has also been an important parameter differentiating the framework from previous works, and proved useful in highlighting the relevance of the work in the industrial setting of the case study. This angle of attack seemed a powerful response to the traditional skepticism which can be found by practitioners who doubt the industrial relevance of such a framework.

8.3 Further works

The framework applied here may be altered to fit the exact needs of other engineering companies. For example, the focus on the lead customer and lead design could be carried out earlier than described in this paper. Also, as high performance mechanical products might experience the need for very different optimization loops, the generic and general inclusion of these is difficult and therefore left out of this work. However, the transferability of the results presented here is generally assessed to be good, as long as industrial practitioners can mobilize the necessary driving force to ensure the architecture work is progressing and is aligned among the major stakeholders.

An improvement area to focus further works on is the creation of quantitative trade-off models to help support designers in critical decision making of where to apply dedicated design and where to allocate efforts for incorporating scalability into the architecture. The aim would be to minimize the costs of complexity while improving time-to-market and the responsiveness towards customer requests. Some early models were applied in this and earlier case studies, but it was out of scope to generalize these for this publication.

Another improvement could be a general assessment model to apply for another case company experiencing similar challenges. The model should support the assessment of the level of readiness to profit from the development of an architecture to serve the customers instead of always proposing dedicated designs. A result could be a maturity model that prescribed which areas to focus a pre-project on before developing the actual architecture in a lead customer project.

9 Conclusion

This paper has presented a framework including a step-wise approach to develop an architecture, particularly suited for OEMs developing mechanically and highly integrated performance products, where traditional modularization is not enough to achieve reuse, scalability, reduce time-to-market while improving R&D resource utilization. The framework and its approach takes it starting point in bridging the few but defining design characteristics and performance parameters of the products with the market parameters that are critical to achieving competitiveness within a target segment. By systematically implementing scalability for main- and sub-solutions an architecture for the product and production setup is developed. This ensures an increased responsiveness towards customer requests and a case study shows promising and significant reductions in time-to-market (almost 50%) and savings in R&D resources used per customer project to develop derived product variants based on the architecture (70%).

The authors would like to thank the case company for sharing challenges, resources and competences with the research team.

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

10 References

Andreasen, M M, Hein, L. (Eds.), (1987). Integrated Product Development. Springer, Berlin

Bickerstaffe, S., (2012). Comment. Automotive Engineer. 37

Claesson, A., (2006). A configurable component framework supporting platform-based product development. Doktorsavh. Chalmers. Tek. Hogsk., 1-159

Du, X., Jiao, J., Tseng, M.M., (2001). Architecture of Product Family: Fundamentals and Methodology. Concurrent Engineering. 9, 309-325

Ericsson, A, Erixon, G. (Eds.), (1999). Controlling Design Variants - Modular Product Platforms. Society of Manufacturing Engineers, Dearborn, Michigan

Gedell, S., Johannesson, H., (2013). Design rationale and system description aspects in product platform design: Focusing reuse in the design lifecycle phase. Concurrent engineering, research and applications. 21, 39-53

Gershenson, Prasad, Zhang, (2003). Product modularity: definitions and benefits. J. Eng. Des. (UK). 14, 295-313

Guo, F., Gershenson, J.K., (2007). Discovering Relationships Between Modularity and Cost. Journal of Intelligent Manufacturing. 18, 143-157

Hansen, C.L., Mortensen, N.H., Hvam, L., (2012). On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market. 12th International Design Conference - Design 2012

Harlou, U., 2006. Developing Product Families Based on Architectures - Contribution to a Theory of Product Families. Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby

Haug, A., Hvam, L., Mortensen, N.H., (2011). The impact of product configurators on lead times in engineering-oriented companies. AIEDAM. 25, 197-206

Hölttä-Otto, K., de Weck, O., (2007). Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. Concurrent engineering, research and applications. 15, 113-125

Hubka, V., Eder, W.E., (1988). Theory of technical systems: a total concept theory for engineering design. Berlin and New York, Springer-Verlag, 1988, 291 p. 1

Jiao, J., Tseng, M.M., (1999). A Methodology of Developing Product Family Architecture for Mass Customization. Journal of Intelligent Manufacturing. 10, 3-20

Joergensen, K., (1992). Videnskabelige arbejdsparadigmer (In Danish). Institute for Production, Aalborg University of Technology, Denmark

Ko, Y., Kuo, P., (2010). Modeling Concurrent Design Method for Product Variety. Concurrent Engineering. 18, 207-217

Krause, D., Eilmus, S., Jonas, H., (2013). Developing Modular Product Families with Perspectives for the Product Program, in: Smart Product Engineering. Springer, pp. 543-552

Latour, B., (1986). Visualization and cognition. Knowledge and society. 6, 1-40

Levandowski, C.E., Corin-Stig, D., Bergsjö, D., Forslund, A., Högman, U., Söderberg, R., Johannesson, H., (2013). An integrated approach to technology platform and product platform development. Concurrent engineering, research and applications. 21, 65-83

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

Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P., Hansen, C.L., (2010). Making Product Customization Profitable. International journal of industrial engineering. 17, 25-35

Otto, K N, Wood, K.L. (Eds.), (2001). Product design: techniques in reverse engineering and new product development. Tsinghua University Press

Pahl, G, Beitz, W., Wallace, K. (Eds.), (1996). Engineering Design - A systematic approach. Springer, London

Pimmler, T.U., Eppinger, S.D., (1994). Integration Analysis of Product Decompositions. American Society of Mechanical Engineers, Design Engineering Division. 68, 343-351

Prasad, (1999). Enabling principles of concurrency and simultaneity in concurrent engineering. (AI EDAM) Artif. Intell. Eng. Des. Anal. Manuf. (UK) Artificial intelligence for engineering design, analysis and manufacturing. 13, 185-204

Prasad, B (Ed.), (1996). Concurrent engineering fundamentals. Prentice Hall PTR

Steward, D.V., (1981). The design structure system: A method for managing the design of complex systems. IEEE Trans. Eng. Manage. EM-28, 71-74

Ulrich, K., (1995). The Role of Product Architecture in the Manufacturing Firm. Research Policy. 24, 419-440

Ulrich, K, Eppinger, S. (Eds.), (2000). Product Design and Development. McGraw-Hill, Boston

Zoia, D.E., (2013). Global Suppliers Favor German OEMs, Survey Finds. Ward's Auto World. 49

6.6 Paper F

NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

Towards a Classification of Architecture Initiatives: Outlining the External Factors

Christian Lindschou Hansen*
PhD stud., Engineering Design and Product Development
DTU Mechanical Engineering
chrlh@mek.dtu.dk

Niels Henrik Mortensen
Prof., Engineering Design and Product Development
DTU Mechanical Engineering
nhmo@mek.dtu.dk

Lars Hvam
Prof., Operations Management
DTU Management Engineering
lhv@man.dtu.dk

Ulf Harlou, PhD
uha@pfmp.com

*Corresponding author

Abstract
This paper introduced a set of external factors capturing the contextual differences that set the stage for architecture initiatives. These are derived from a systems theoretical approach recognizing the fact that architecture initiatives should respond the challenges posed by the external environment in which the company and the future product program is operating. The outlining of the factors are based on the conviction that no one-fits-all exists, when it comes to architecture initiatives, and the notion that it is impossible to truly evaluate whether an architecture initiative is good or bad, without including the contextual differences. The purpose of the external factors is to improve scoping and goal setting of architecture initiatives, and improve comparability between- and transferability of knowledge from architecture initiatives. The external factors are a first step towards an actual classification of architecture initiatives.

Keywords: Product architecture; product platform; systems theory; contingency factors

Introduction

Challenges

A vast array of new methods and techniques for successful implementation of product architecture initiatives are presented every year in various conferences. The contributions are based on experiences from many different companies and research work. However, due to the fact that product architectures are a complex phenomenon in itself, the findings and discoveries reported from research originating from specific architecture initiatives, can be difficult to transfer from one context to another. And while researchers often neglect to include the contextual differences that set the boundaries and conditions for the architecture initiative, it is difficult for practitioners to adapt towards and benefit from the latest ideas and concepts.

In close relation to these challenges is that the lack of inclusion of contextual differences makes it difficult to evaluate whether an architecture initiative is good or bad. There is no one-fits-all when it comes to the tailoring of architecture initiatives to a specific situation of a company. The lack of inclusion of contextual characteristics simply poses a risk for unsuccessful scoping and goal setting of architecture initiatives eventually leading to underperforming product programs.

This paper addresses this challenge by outlining a set of external factors that capture the most significant contextual differences, as a first step towards an actual classification of architecture initiatives. This is based on the basic hypothesis that proper classification is a prerequisite for improving the maturity of research within product architecture initiatives. Here, the aim is to improve the scoping and goal setting of architecture initiatives, and improve generalizability of research in architecture initiatives as a prerequisite of comparability and transferability.

The external factors are proposed based on the experiences collected from various action-based research studies, which leads towards a later publication of an overall classification.

A systems theory view

In order to address the challenges described above systems theory is applied.

Definition of the system and its boundaries

When developing a new product architecture, is has previously been presented how it is necessary to define the product architecture in close coordination with the market aspects and the production setup. This approach is captured in the DTU framework for architecture initiatives (see Figure 1) [1], [2].

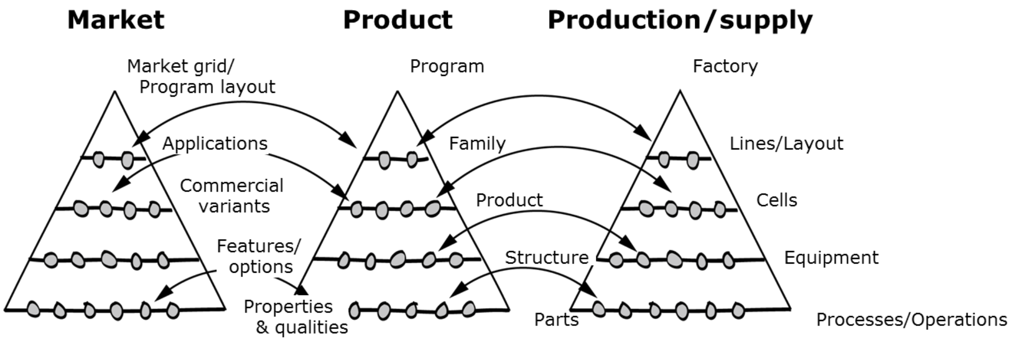


Figure 1 – DTU framework for architecture initiatives

The framework builds upon the classical partitioning of the market, product and production/supply domains [3]. This is most recently presented and described as the market architecture, product architecture and production/supply architecture. The elements described in each pyramid, can be seen as the behavioral and constitutive elements of an architecture that an architecture initiative can change and affect.

In order to separate the architecture initiative from its surroundings, it is necessary to consider the architecture under development as a system. The architecture initiative can change elements *within* the system (the architecture) as a response to the external factors.

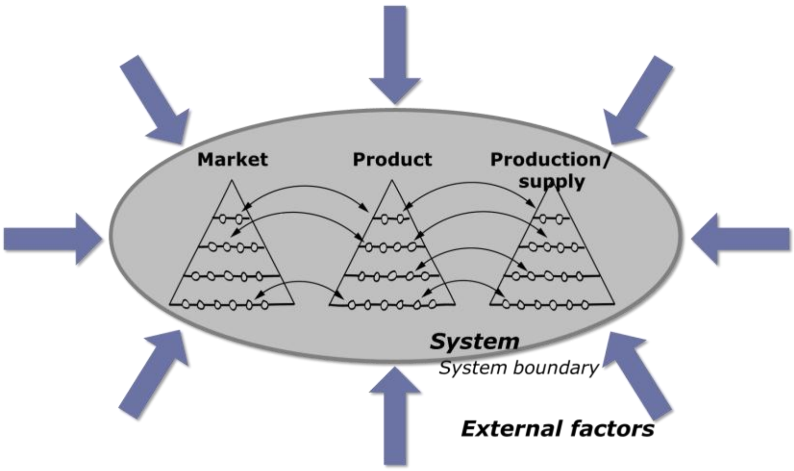


Figure 2 – System, boundary and external factors

Figure 2 shows the architecture as a system. The task of the architecture initiative is to provide a response to the external factors, in order to improve competitiveness of the company where the architecture is developed. The important aspect here is that the external factors provide the conditions for the environment in which the system is performing. Therefore, there are no such “absolute truths” when it comes to architecture initiatives. From contingency theory Galbraith (1973) described this phenomenon years ago [4], stating such design of complex systems, not one single design can be the best in all cases and that the best design depends on the character of the environment, which the system is going to interact with.

This paper will address the challenges above by taking an important first step towards a classification of architecture initiatives. Firstly, the motivation for the classification is described in headlines including small examples; secondly, the external factors are outlined to describe the contextual differences setting the boundary conditions for an architecture initiative; and lastly, a few examples are shown including a short reflection and a conclusion.

Why do we need to classify architecture initiatives?

Motivation

Scoping and goal setting of architecture initiatives
The primary motivation for classifying architecture initiatives, is to improve the scoping *and goal setting* of architecture related initiatives in general. The authors repeatedly experience a mismatch between the definition of the initiative and the situation the initiative should respond to. In other words, often, the concepts and ideas of the architecture initiatives do not match the challenges companies are facing. Therefore, many companies end up in classic pitfalls [1], due to the lack of proper scoping *and goal setting of the architecture initiative eventually leading to architectures that are not appropriately tailored to the situation.*

Improve comparability of initiatives
A secondary motivation for classifying architecture initiatives is the difficulty experienced when comparing different architecture initiatives with each other. Often, attempts to generalize experiences end up in rough simplifications (when the abstraction level gets too high) or alternatively too much “contextual noise” (when the abstraction level is too low). A classification of architecture initiatives should serve to overcome these challenges by providing a common “language” for either making comparison possible, or support and clarify why comparison is not possible.

Improve transferability of experiences
The comparability of initiatives should also serve to improve transferability of experiences between researchers and between the academic societies and industrial practitioners. Many concepts and ideas from academia are more or less randomly dispersed across different industries, often not optimally scoped or tailored to suit the needs of individual companies being in different situations.

State of the art

Adjacent fields of research

External complexity drivers
Bliss (2000) defined three external complexity drivers determining the “market complexity”, namely demand-, competitive- and technological complexity [5], and argues that companies must adapt their internal complexity to match these external complexity drivers.

Contingency theory
Zeithaml et al. (1988) formulated a number of principle solutions or responses that can be used to satisfy the requirements of a given competitive environment [6].

Strategy definition
One of the most significant contributions within this area is still Porter's (1980) generic strategies [7]: Market segmentation/focus, cost leadership or product differentiation. However, an appropriately scoped and successful architecture initiative can serve to *combine* these strategies, e.g. enabling differentiation and cost leadership at the same time.

The production task
Skinner (1974) argued decades ago, a blind-spot for most production managers is the attempt to design a production setup that has to compete with an impossible mix of demands [8]. Many additional contributions from Skinner emphasized the strategic definition of the production task as a central aspect of the corporate strategy, and a powerful response to external competition. As the requirements for a production system are dynamic, the production task is not stabile and changes over time

The specification task
Hansen (2003) contributed similarly by describing the need for analyzing and defining the task of the variant specification system [9].

Business structures vs. competitive conditions
Sant (1988) linked typical business structures, competitive conditions and product offerings in relation to the market life-cycle phase of a product program [10]. This was a continuation of Skinner's definition of the production task to include the domain of product development and place this into a business perspective.

Industry life-cycle and game rules
Johnson and Scholes (2008) proposed life-cycle model of an industry and merged this with various business structures to derive a set of basic game rules [11]. The game rules highlight the most important competitive parameters of the different industry life-cycle phases.

Product architecture
Mortensen et al. (2005) argued the need for modeling of *opportunity roadmaps* to capture the need for future changes of features, technologies, standard designs and products [12].

Open innovation
Riitahuhta et al. (2011) defines a Company Strategic Landscape, within which the product structure must be aligned with the value chain structuring, strategy structuring, process and service structuring, and organizational structuring [13].

A life-cycle view
The Design-for-X life-cycle perspective also entails a line of contributions centered on integrating life-cycle knowledge in product development by recognizing the need for e.g. time-to-market focus by corresponding Design for time-to-market methods [14].

Variant management
The German school of variant management provides a vast number of methods and techniques to optimize the design of variance in product families as a response to the external factors from a competitive environment.

Gap
Very few contributions have dealt with the definition of external factors that act as boundary conditions of an architecture initiative. The clear focus on the task definition and the

influencing factors of this seems to be rather isolated to the production domain, and there is a need to implement this thinking in architecture development in order to enable the improvement of scoping, comparability and transferability of architecture initiatives. For example, a large number of contributions focus on modularity as a goal in itself, even though modularity will always remain *a means* to achieve desirable effects in response to the challenges imposed by external factors.

Towards a classification: The external factors

In order to present a classification of architecture initiatives, this paper will propose a set of external factors that the classification has to take into account. The actual classification will be presented in a later publication.

Framework of reference

To be able to capture the complex aspects of architecture initiatives, it is proposed to refer to an architecture framework recently proposed by the authors (see Figure 1). This is in compliance with many of the contributions presented in the previous section thus creating a solid foundation for deriving the set of external factors.

The external market factors

Market launch clock speed

The frequency of market launches has a large impact on the planning of new product introductions. Certain companies are situated in industries with e.g. yearly trade exhibitions that make it necessary to target new product introductions to these. Other companies are operating in industries where continuous product launches and upgrades are expected to keep the attention of the main markets. The market launch clock speed can be determined strictly by external parameters in certain companies, whereas it can be the sole decision of a strong marketing department in other companies. Again, the market launch clock speed has a huge impact on the stability it is possible to implement in the architecture development, as the need for an *evolving and transforming* architecture arises when the clock speed increases.

Marketing channel/supply chain position

The channel of which the products are delivered to the market place is another influencing factor of which architecture initiative to work with. In situations where products are delivered directly to end-users, the architecture should be prepared for a clear differentiation of offerings through *features with positioning properties*.

In other situations sales are carried out through sales subsidiaries to wholesalers, where *range completeness* and a leveled distribution of variants throughout the offerings spectrum can be achieved with *balanced performance steps*. The architecture has to be prepared for that, as this has an impact on e.g. basic technological scaling principle.

Thirdly, other companies sell their products to contractors or technical advisors that focus solely on sales price and minimum required performance. In these cases the *optimal distribution of cost- and price points* and the use of *proven technology* are of fundamental importance.

Lastly, other companies sell their products through public procurement agencies (e.g. medical products) that need a strong *formal justification of incrementally added value*, compared to previous product generations, as formal documentation – often accompanied by passing formal test procedures. In these cases, the product and production architecture is strongly assigned to accomplishing these obligatory properties in order to be part of public tenders etc.

Market positioning
The market share and bargaining power are important factors closely related to the factor of product customization.

If customers are in possession of the bargaining power, the architecture can either only be prepared to a certain level (CTO/ETO), or a strong cost focus must be applied throughout the architecture development. On the other hand, if the company has the bargaining power, the architecture initiative can be focused to maximize feature multiplicity and launch clock speed.

The market share can be equally decisive for the focus of the architecture initiative, as factors as range completeness can be important to maintain a large market share, and as unique differentiation can be important to maintain a niche market share.

The external product and production factors

Primary driver for product positioning
The primary positioning driver of products to be derived from the architecture is of course specific from company to company. However, in general there is often a focus on *sheer performance* or *feature multiplicity* in order to position the products ahead of those of competitors.

A focus on sheer performance sets a number of physical constraints on the architecture, as mechanical compromises can be difficult to match with functional encapsulation and modularization. In most cases, the challenge is here for the architecture to enable the functional and physical encapsulation around the performance critical parts or modules, or isolate the modularization efforts to the production domain. In other cases, the architecture initiative could be centered on defining an integrated but scalable structure of the products, in order to reduce lead time of the development task and production ramp-up.

A focus on feature multiplicity can be ideal as a driver for modularization as part of the architecture initiative, and the focus will be the balancing of feature variety and payment willingness towards incremental production investments and development lead time.

Product customization
The type of product customization is included as an influencing factor. Here, a distinction between whether the market can be served with a *definite solution space* or an *open solution space* is made.

In companies having a definite solution space, pre-defined product variants are developed in discrete instances. These companies are also nominated product-based companies, and the focus of the architecture initiative should be the preparation of multiple planned product launches while minimizing the internal resource consumption.

In companies having an open solution space, configurable product variants are customized. The architecture developed here is focusing on isolating the reusable standard designs from the customer-specific design units, while focusing on preparing the architecture for short development lead times. Here, the challenge of the architecture is to enable a controlled specification of customized products (e.g. with configurators) to guide customers towards similar solutions in order to reduce internal complexity. The solution space can be more or less defined often differentiating between Configure-to-order (CTO) or Engineer-to-order approaches.

Product and production technology clock speed
The frequency of technology renewal has a large impact on the *stability* it is possible to implement in the architecture development. High technology clock speeds often rule out

physical reuse, thus focusing the architectural potential at a higher structural level. For instance are elements on parts and process level not standardized, but product structures and product equipment might be reused across product generations and families. Or, physical reuse can only be obtained by thorough encapsulation (e.g. by isolating functionality completely).

High technology clock speeds can result in very short market life cycles of products or short life cycles of production equipment leaving a small room for architecture initiatives focused on traditional reuse. Also, technology clock speed is a high determinant for the dependency towards technology development centers and external suppliers of key components.

Volume per variant
In continuation of the influencing parameter of product customization, the volume per variant is another parameter differentiating architecture initiatives.

Companies following an ETO approach are developing *one-off* products in some cases. In these situations the architecture initiatives focuses of interface management, decoupling of the development task and a close integration of requirements from the installation and commissioning phase.

Other companies following a CTO approach are often manufacturing a relatively low volume of each variant. In these cases, the architecture initiative cannot pursue benefits from economies of scale between the low volume variants, but the development of a robust production architecture can be another good way of ensuring competitiveness as long as preferred solutions are implemented in configuration systems to control the specification of new variants within the boundaries of the production capabilities.

In high volume production, the architecture initiatives should strive to accomplish the virtues of a traditional mass customization paradigm.

Macro-economic environment
It is necessary to mention the macro-economic environment as well. The interest rate, currency exchange rate, customs duties, logistics costs, market accessibility and legislation, raw material prices all play a huge role for the placement of production sites, sourcing of parts, supply chain design and choice of materials etc. Globalization has made the importance of macro-economic factors even more evident, and most factors remain relatively unstable.

Experiences from application

It is the experience of the authors, that it is immensely important to take the external factors into account while scoping architecture initiatives. The central point here is that different contexts require different solutions. There are no one-fits-all when it comes to the scoping and definition of powerful architecture initiatives, and many parallels can be drawn to the research conducted within the production domain on the definition of the production task. However, these aspects become even more important concerning architecture initiatives, as the product and production architectures share a number of relations. Therefore, it is of fundamental importance to include the external factors and provide a clear definition of the task that the architecture initiative should solve:

- Map the external factors of importance
- Prioritize which factors to take into account
- Concretize and quantify how to address the factors
- Design the architecture initiative to respond to the external factors

The experiences are gained through numerous action-based research studies within primarily Scandinavian industrial companies.

Reflection and further work

This paper is just a beginning. A structured and systematic ongoing work with the factors lies ahead of the authors. In addition to this, the next step of this research is to develop the actual classification of the initiatives. It is the ambition to develop not only a framework, but also a "guide" for researchers and industrial practitioners. A central aspect here is to map the external factors towards a set of generic types of initiatives and outline a set of practical and action-oriented solution recommendations. The understanding of the external factors presented here is seen as a prerequisite of this next step.

Conclusion

The paper has introduced a set of external factors capturing the contextual differences that set the stage for architecture initiatives. These are derived from a systems theoretical approach recognizing the fact that architecture initiatives should respond the challenges posed by the external environment in which the company and the future product program is operating. The purpose of the external factors is to improve scoping and goal setting of architecture initiatives, and improve comparability between- and transferability of knowledge from architecture initiatives. The external factors are a first step towards an actual classification of architecture initiatives.

References

[1] Hansen, C.L., Mortensen, N.H., Hvam, L., 2012. On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market. 12th International Design Conference - Design 2012

[2] Mortensen, N.H., Hansen, C.L., Hvam, L., Andreasen, M.M., 2011. Proactive modeling of market, product and production architectures. Proceedings of the 18th International Conference on Engineering Design : Impacting Society through Engineering Design, 133-144

[3] Andreasen, M.M., Hein, L., 1987. Integrated Product Development. Springer, Berlin

[4] Galbraith, J.R., 1973. Designing complex organizations. Addison-Wesley Longman Publishing Co., Inc.

[5] Bliss, C., 2000. Management von Komplexität. Gabler, Germany

[6] Zeithaml, V.A., Varadarajan, P."., Zeithaml, C.P., 1988. The Contingency Approach: Its Foundations and Relevance to Theory Building and Research in Marketing. eur j mark. 22, 37-64

[7] Porter, M.E., 1980. Competitive strategy: Techniques for analyzing industry and competitors. Free Press, New York

[8] Skinner, W., 1974. The Focused Factory. Harvard Business Review. 52, 113-121

[9] Hansen, B.L., 2003. Development of industrial variant specification systems. Department of Manufacturing Engineering and Management, Technical University of Denmark, Lyngby

[10] Sant, K., 1988. Udviklingsfunktionen: Fastlaeggelse af udviklingssystemet (in Danish). Jernets Arbejdsgiverforening; The Technical University of Denmark, Lyngby

[11] Johnson, G., Scholes, K., Whittington, R., 2008. Exploring corporate strategy: text & cases. Prentice Hall

[12] Mortensen, N.H., Harlou, U., Andreasen, M.M., 2005. Identification of platform levels in product development. ICED 05: 15th International Conference on Engineering Design: Engineering Design and the Global Economy

[13] Riitahuhta, A, Lehtonen, T., Pulkkinen, A., Huhtala, P. (Eds.), 2011. Open Product Development. Springer

[14] Simpson, T.W., Siddique, Z., Jiao, J., 2006. Product platform and product family design: Methods and applications. Springer, New York

6.7 Paper G

NordDesign 2012
August 22 – 24, 2012
Aalborg, Denmark

Calculation of Complexity Costs – An Approach for Rationalizing a Product Program

Christian Lindschou Hansen
PhD stud., Engineering Design and Product Development
DTU Mechanical Engineering
chrlh@mek.dtu.dk

Niels Henrik Mortensen
Prof., Engineering Design and Product Development
DTU Mechanical Engineering
nhmo@mek.dtu.dk

Lars Hvam
Prof., Operations Management
DTU Management Engineering
lhv@man.dtu.dk

Ulf Harlou, PhD
uha@pfmp.com

Abstract
This paper proposes an operational method for rationalizing a product program based on the calculation of complexity costs. The method takes its starting point in the calculation of complexity costs on a product program level. This is done throughout the value chain ranging from component inventories at the factory sites, all the way to the distribution of finished goods from distribution centers to the customers. The method proposes a step-wise approach including the analysis, quantification and allocation of product program complexity costs by the means of identifying of a number of suggested Life Cycle Complexity Factors (LCCFs). The suggested method has been tested in an action based research study with promising results. The case study shows how the allocation of complexity costs on individual product variants provides previously unknown insights into the true cost structure of a product program. These findings represent an improved decision basis for the planning of reactive and proactive initiatives of rationalizing a product program.

Keywords: Complexity costs; product program; rationalization; product architecture

Introduction

Challenges
Most industrial companies offering a multitude of product variants to the market have accepted that this situation comes with a price – it is not free to handle many product variants. In fact, it has been shown in numerous examples that all stages in the life cycle are affected by the variance in a product program. To diminish the negative effect of this "necessary evil", many efforts have been made especially during the last 15-20 years to improve the marketing, design, production and management of product programs. One unavoidable means within this area is the application of architecture-based development of product platforms. Significant contributions are found in this research field, but very little research has been centered on the actual quantification of the benefits to be achieved from the architecture-based approach. There are many reasons for this. One is that established accounting systems (e.g. in ERP-systems) focus on the direct product costs alone (e.g. standard unit cost). This is done with only sparsely including indirect costs (often equally distributed on all variants) and also without focusing on the performance of the processes delivering the product program. Therefore, the experience of the authors is that a number of promising product architecture concepts are never implemented, due to the lack of quantification in order to justify the positive effects (or diminishing of negative effects) associated with the product architecture concept throughout the product life cycle.

The method

This paper proposes an operational approach for calculating complexity costs in order to rationalize a product program. The complexity costs are calculated throughout the product life cycle and are allocated to individual product variants. This is done in order to understand the negative effects of the product variants within a product program and to obtain a better measure of the profitability of individual product variants. The approach has two important objectives:

Reactive

- Reach an understanding of the product program complexity costs
- Rationalize the product program

Proactive

- Scoping of product architecture initiatives
- Improve program (or portfolio) management

The approach has been tested in an action based research study providing not only a basis for achieving short term gains in terms of rationalizing the product program (increased EBIT margin), but also a basis for reconfiguring the supply chain to achieve a number of beneficial effects.

The paper will continue by outlining the motivation for calculating complexity costs and put the topic into an industry perspective. Subsequently, the research perspective is elaborated by compactly treating the current state-of-the-art within this composed area. Finally the 5-step approach is presented, reflected and concluded upon.

Why calculate the complexity costs of a product program?

Motivation

As mentioned in the previous section, there are two main reasons to calculate the complexity costs of a product program:

Reactive use

In order to make room for new product introductions, it is often required to rationalize the product program on a frequent basis by eliminating and/or substituting product variants. Sometimes there is a 1:1 substitution of old product families with new product families, but often the situation is much more distorted than that, making it difficult to choose which variants to phase out. And since these discontinuation activities often rely on data readily available, the product variants to eliminate are often based on revenue or unit sales alone. This is without considering the total profitability of individual product variants, and without an overview of the complexity costs associated with the product variants in the product program. In other words, there is often a lack of cost transparency across the product program resulting in portfolio decision-making made without knowledge about the profitability of single product variants, thus also without knowledge about the burden with which individual product variants impact the indirect costs.

Therefore, there is a need to understand the profitability of individual product variants and include the calculation of complexity costs on a product program level, in order to:

- Obtain a more “true” product variant performance measurement
- Obtain an overview of the product life cycle complexity costs

This knowledge can enable the rationalization of a product program, by:

- Substituting unprofitable product variants with profitable ones where possible
- Eliminating remaining unprofitable product variants

As most companies have introduced stage-gate models or equivalent to control the introduction of new products, not much attention has been paid to professionalize the discontinuation task in an equal manner. Reactive use of this approach can contribute to professionalize this task.

Proactive use
Besides being a valuable input for reactive use, the knowledge about profitability of product variants and their associated complexity costs should be used proactively to avoid the increase of complexity costs to maintain a continuous increase in profitability.

As mentioned in the introduction, architecture-based development of product families are centered on providing the right variance to the market place while at the same time diminishing the negative effects experienced internally in the company's operations. As such initiatives cannot improve all processes at the same time; there is a strong need of scoping such architecture initiatives in order to reach the desired effects. As these effects are often found in a decrease of indirect costs, there is a need of identifying, allocating and analyzing these costs to surpass the paradigm of one-sided focus on the direct variable costs and step away from even distribution of indirect costs.

Therefore, in terms of scoping architecture initiatives, there is a need for utilizing the knowledge of product variant profitability and the calculation of complexity costs of the product program, in order to:

- Identify the most profitable product variants and families (and learn from these)
- Identify the least profitable product variants and families (and *fix* these)
- Identify the largest complexity costs and their associated factors (in order to know which ones to address)

This can be exemplified by:

- Design-for-X focus: Which life phase or universal virtue has the largest potential for complexity cost savings?
- Order fulfillment strategy: Where to place the customer order decoupling point?
- Guide market pricing: Which are the optimal price and cost points across the product program?

Besides from improvement projects, the continuous tasks of program (or portfolio) management can be improved by:

- Performance measurement: Introduce product program complexity key performance indicators to take complexity costs into account (and formulate new minimum profit thresholds etc.)
- Product planning: Guide product launch and discontinuation strategies

State of the art

A literature study was undertaken to screen the research landscape for recent contributions within this field. The literature study was broadened to comprise recent studies from supply chain engineering and management, and the most relevant ones are commented here. The focus has been contributions utilizing calculation of complexity-related costs in order to rationalize product programs.

Adjacent fields of research

Activity-based costing
Cooper and Kaplan [1] among others suggested Activity-based costing as a new method to avoid the deficiencies of arbitrary allocation of overhead costs. Activity based costing allocates indirect costs first to the activities performed by shared company resources, and hereafter assigns these to individual orders, customer or even products. Thus, the method takes its starting point in the resources used, links these to activities, and *then* to cost objects.

Anderson and Kaplan [2] proposes a more accurate and efficient cost modeling principle called Time-Driven Activity Based Costing (TD-ABC) that assigns resources (e.g. all costs of a customer service department) directly to cost objects (e.g. order handling). This is done to achieve a simple cost rate measure based on time consumption.

Supply chain engineering and variant management
Lechner et al. [3] proposes the method Variety-driven Activity-based Costing (VD-ABC) to quantify the impact from adding or removing product variants in automotive logistics, based on the use of hypothetical zero-variant scenarios. This is an expansion of the TD-ABC framework allowing for the calculation of incremental complexity costs associated with variants in different logistical operations.

Mass customization
Zhang and Tseng [4] propose a modeling approach to analyze cost implication of product variety in mass customization by bridging product variety with process variety. This is done by identifying cost drivers within the product design, and the method is confined to include manufacturing costs.

Product and product program complexity dimensions and indicators
Orfi et. al [5] proposes a set of product complexity dimensions (variety, functionality index, structural index, design index, and production index), and along with these associated indicators considering the cost impact of the product complexity dimensions. Nielsen and Hvam [6] showed that product program complexity is not just harming order management, procurement and inventory costs but also has a negative effect on delivery performance and product quality.

Complexity costs definitions
It is out of scope of this short paper to review the different definitions of complexity costs.

Complexity Management
Sivadasan et al. [7] (among others) describe two types of complexity in the supply chain, *structural complexity* (increasing with the no. of elements) and *operational complexity* (increasing with uncertainty of information and element flows). Many methods within supply chain research have suggested methods to eliminate and control this complexity from a supply chain point of view. Wilson and Perumal [8] are among recent contributions offering several top-down approaches to attack interrelated product-process-organizational complexity from a

managerial view point by diving complexity costs into value adding (good complexity) and non-value adding (bad complexity).

Also, a number of product variant rationalization approaches focus solely on different “tail-cutting” methods (often named SKU-rationalization), the most interesting ones showing that there is no relation between the number of stock keeping units and market share [9].

Gap
Acknowledging the “complex” landscape of contributions within this field, one can mention that much research is centered on the definition and calculation of one might call “internal” product complexity (inside the product) and much research (especially from the supply chain area) are centered on the control of complexity from a process point of view.

Product complexity methods
The deficiency with these methods most often overlook the fact that complexity is a relative phenomenon arising between a product and a process, and thus cannot be assessed meaningfully with regards to the product and its properties itself.

Process complexity methods
The deficiency with these approached are their exclusion of the product domain in order to derive advanced numerical approaches to the calculation of process complexity costs. This has its relevance for detailed optimization tasks, but is very difficult to use as input for proactive avoidance of complexity costs.

Conclusion on gap
Considering complexity as a relational phenomenon between e.g. a product and a process (within any life cycle phase), it is in principle impossible to derive universal metrics of complexity *before* an actual investigation of the unique product/process setup had been carried out, in order to assess the actual realized costs of complexity associated with the setup. Not before understanding the *realized* costs of complexity, efficient means to reduce the future costs of complexity can be defined. For example, having 10.000 variants might not be a problem if software is configuring the variance.

It is the aim of this approach, to take a first step towards the bridging of the supply chain based attempts to quantify complexity from a process point of view, with the architecture-based approach to product program design and the elimination of negative effects associated with the handling of many product variants.

The method

Introduction
The method presented here takes its starting point in the calculation of complexity costs on a product program level entailing a focus on product variants as the complexity cost allocation objects.

Step1: Scoping of analysis
In order to determine the focus of the analysis, the scope within the product program must be decided upon. It is advisable to select a confined number of product families to include produced on a limited number of production sites and perhaps sold in a limited amount of regions worldwide. It is possible to include product families from both high-end and low-end market tiers, as long as there is adequate resemblance in the way the products are produced and handled internally in order to analyze them concurrently.

Considering the costs of complexity within the product program, it is recommendable to choose product families experiencing a gap between projected profitability and realized profitability (if known to the project group).

Step 2: ABC analysis of product profitability

The first evaluation of the gross profitability of the product variants are made by collecting the realized revenues of all variants from all sales companies with a given time period. The direct product costs are subtracted from this figure resulting in a measure of the gross contribution margin. This can be plotted for all variants on the vertical axis leaving the horizontal axis to display the net revenue per variant – as shown in Figure 1.

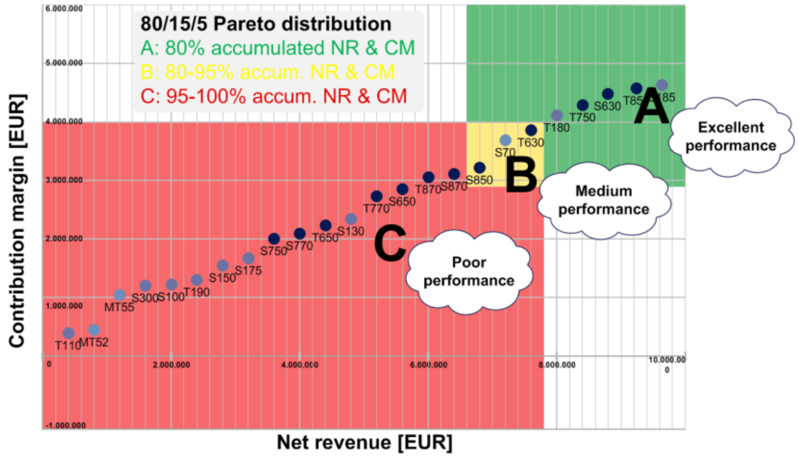


Figure 1 – ABC analysis of product profitability

A Pareto-distribution can be used to highlight the variants contributing only to the last 5 % (C-variants) and 15% (B-variants) of the total contribution margin and net revenue.

Step 3: Life Cycle Complexity Factors (LCCFs)

Identification

With Step 2's early indication of the least contributing variants, the aim of Step 3 is to investigate which factors throughout the product life cycle that holds the largest complexity costs and find out whether these complexity costs distort the picture shown in Figure 1. A central aspect here is to look after LCCFs that could represent an *asymmetric cost distribution* across the product variants. In other words, look for pools of resources that are consumed differently among the product variants. An example of a LCCF includes cost of inventory (of materials, components, sub-assemblies and/or finished goods).

As LCCFs vary greatly between industries and company types, this paper will not go into details here (a future publication will expand the concept of Life Cycle Complexity Factors). However, their identification requires iterating between a top-down and a bottom-up approach:

- *Top down: Cost structure view*
  Looking at the overall cost structure of the business area, where do we see the largest unallocated cost pools with a potential variant impact?
- *Bottom up: Hypotheses of cost asymmetry*
  Based on the experiences of key resources, where is it likely that product variants contribute unevenly to the indirect costs?

Analysis, quantification and allocation
The basic idea here is analyze the LCCFs and find quantification objects that allows for approximations of the indirect costs in order to allocate them directly to product variants where applicable. By dividing all costs with the net revenue recorded on each variant – all costs are comparable as percentages. Often, it is necessary to settle with incomplete data extracts, and be creative in applying unconventional quantification objects to bring forward reliable approximations (inspiration can be found in [2] and [3]).

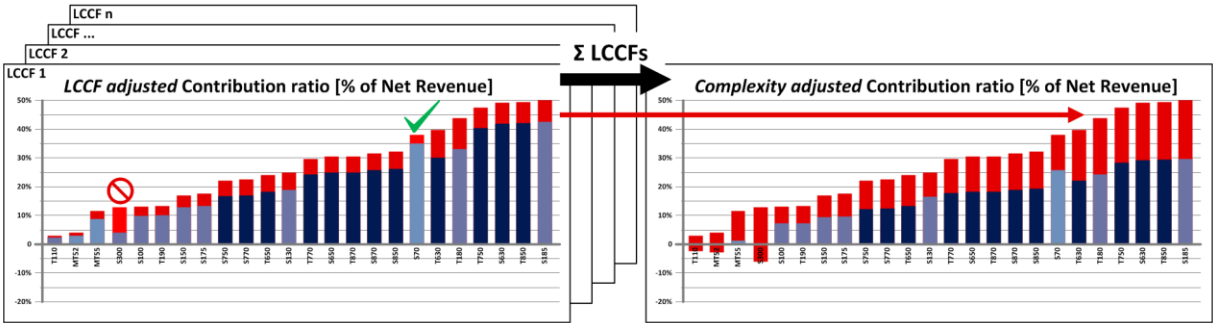


Figure 2 – Adjusting contribution ratios for LCCF induced complexity costs

As seen in Figure 2 the costs allocated from the analysis and quantification of LCCFs can be accumulated to give an overview of the complexity adjusted contribution ratios [%] (as well as the complexity adjusted contribution margins [EUR]). The right side of Figure 2 is the best possible estimation of the true profitability of the product variants.

Step 4: Short-term fixing
Based on the insights from step 2-3, it is now possible to calculate different scenarios of "fixing" the product program by the means of the reactive measures mentioned earlier. Most often, a thorough analysis reveals several low-hanging gains, and several percentage points of increased EBIT are usually the results of this. It is important not to assume 0% substitutability of discontinued variants (rare cases only). Assume a decent percentage in order to estimate the true incremental revenue loss, which is always a lot less that the actual revenue recorded on the variants.

Step 5: Complexity reduction program
As Step 4 is about cleaning up the product program reactively, Step 5 is about implementing the findings from Step 2-4 proactively in a complexity reduction program. A central aspect here is to identify the *drivers* causing the LCCFs to create complexity costs. We name these the LCCDs (Life Cycle Complexity Drivers) and the aim is to work with these drivers to decrease their negative impact on complexity. An example might be country specific customization of product variants that could be solved differently. The cost transparency achieved in Step 3 can serve to justify the cost of changing the country specific feature or solution.

The complexity reduction program contains the initiatives of diminishing the negative effects of the complexity cost factors (e.g. by postponing the customer order decoupling point) and initiatives of actively working with the complexity cost driver to eliminate them or delimit their negative effect on the factors.

Experiences from application

Case

The approach was applied in an action research-based study in globally leading manufacturer of mechanical consumer products. The approach was followed from Step 1 to Step 5. A number of factors supported the success of the approach leading to a large potential EBIT increase. Firstly, the company has a long history of mergers and acquisitions and is operating through a global supply chain and a global sales organization creating long "distances" and much room for cost distortion from revenue generation to cost allocation. Secondly, recent SKU-rationalization initiatives were solely based on revenue leaving a lot of improvement potential behind. Thirdly, many initiatives concerning the complexity drivers were put on hold due to the lack of quantification to support a business case narrowly focusing on standard unit cost. The application of the approach improved the cost transparency, refined the SKU-rationalization and provided a basis for continuing and scoping the initiatives of eliminating the negative effects of the complexity cost drivers.

Reflection and further work

The incorporation of hypothetical single-variant scenarios could be relevant in order to gain input for the estimation of "variant cost sensitivities". The performance of all operations depend on the number of variants, and estimating this variant cost sensitivity across functions or major fixed cost pools can be valuable input. Also, further operationalization of the identification and allocation of complexity costs is relevant. Even though complexity costs do not derive from products individually, refined allocation methods refined allocation methods for assigning these costs on component/product/product family level can help to identify the cost asymmetry of interest.

Conclusion

This paper proposes an operational method for rationalizing the product program based on the calculation of complexity costs. This is done by obtaining a "true" measure of the product cost through the identification of the largest and most asymmetric Life Cycle Complexity Factors (LCCFs) to provide a comprehensive input for quantitatively assessing the benefits of streamlining the product program. The nature and size of the LCCFs are a strong and non-negligible input for prioritizing initiatives of a complexity reduction program. The method was tested in an action-research based case study with promising results.

References

[1] Cooper, R., Kaplan, R.S., 1988. Measure Costs Right - Make the Right Decisions. Harvard Business Review. 66, 96-103
[2] Anderson, S.R., Kaplan, R.S., 2007. Time-Driven Activity-Based Costing: A Simpler and More Powerful Path to Higher Profits. Harvard Business School Press Books
[3] Lechner, A., Klingebiel, K., Wagenitz, A., 2011. Evaluation of product variant-driven complexity costs and performance impacts in the automotive logistics with variety-driven activity-based costing. IMECS - Int. MultiConference Eng. Comput. Scientists. 2, 1088-1096
[4] Zhang, M., Tseng, M.M., 2007. A Product and Process Modeling Based Approach to Study Cost Implications of Product Variety in Mass Customization. IEEE Trans. Eng. Manage. 54, 130-144
[5] Orfi, N., Terpenny, J., Sahin-Sariisik, A., 2011. Harnessing Product Complexity: Step 1-Establishing Product Complexity Dimensions and Indicators. Eng. Econ. 56
[6] Nielsen, J.B., Hvam, L., 2011. Product Complexity Impact on Quality and Delivery Performance. World Conference on Mass Customization, Personalization, and Co-Creation
[7] Sivadasan, S., Efstathiou, J., Frizelle, G., Shirazi, R., Calinescu, A., 2002. An information-theoretic methodology for measuring the operational complexity of supplier-customer systems. ijopm. 22, 80-102
[8] Wilson, S.A., Perumal, A., 2009. Waging War on Complexity Costs, 1st ed. McGraw-Hill, Europe
[9] Mahler, D., Bahulkar, A., 2009. Smart complexity. Strategy & Leadership. 37, 5-11

6.8 Paper H

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

The reduction of product and process complexity based on the quantification of product complexity costs

Lars Hvam, Christian Lindschou Hansen, Cipriano Forza, Niels Henrik Mortensen & Anders Haug

To cite this article: Lars Hvam, Christian Lindschou Hansen, Cipriano Forza, Niels Henrik Mortensen & Anders Haug (2019): The reduction of product and process complexity based on the quantification of product complexity costs, International Journal of Production Research, DOI: 10.1080/00207543.2019.1587188

To link to this article: https://doi.org/10.1080/00207543.2019.1587188

Published online: 04 Mar 2019.

International Journal of Production Research, 2019
https://doi.org/10.1080/00207543.2019.1587188

The reduction of product and process complexity based on the quantification of product complexity costs

Lars Hvam[a*], Christian Lindschou Hansen[b], Cipriano Forza[c], Niels Henrik Mortensen[d] and Anders Haug[e]

[a]*Centre for Product Modelling (CPM), Department of Management Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;* [b]*Product Architecture Group, Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;* [c]*Department of Management and Engineering, University of Padua, Vicenza, Italy* [d]*Product Architecture Group, Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark;* [e]*Institute for Entrepreneurship and Relation Management, University of Southern Denmark, Kolding, Denmark;*

(*Received 13 April 2018; accepted 18 February 2019*)

Complexity management is an increasing challenge for industrial companies. To address this issue, this paper develops a procedure to reduce the complexity of products and processes. This procedure includes five steps: (1) definition of the scope of the products and processes to be included in the analysis, (2) grouping of products into A,B, and C categories, (3) identification and quantification of the most important complexity cost factors, (4) identification of initiatives for the possible reduction of complexity costs and the quantification of possible cost savings, and (5) evaluation and prioritisation of initiatives. To test the usefulness of the suggested procedure, it was applied at a globally leading manufacturer of mechanical consumer products. The case study demonstrated the usefulness of the proposed procedure in (1) supporting the allocation of complexity costs in relation to individual product variants, (2) achieving a better understanding of the cost structure of product assortment and business processes, and (3) providing a basis for generating and evaluating initiatives aimed at reducing the complexity of products and processes. The case study also showed that the use of the procedure can produce considerable financial benefits.

Keywords: complexity management; complexity costs; quantification of complexity costs; complexity reduction; product architecture

1. Introduction

Complexity management involves identifying and reducing complexity within companies, which relates to products, business processes, and organisation, as well as the relationships between them (Wilson and Perumal 2009). Companies increasingly experience complexity due to a number of factors, including more diverse markets and groups of customers, product variants, subassemblies and components, production sites and sub-suppliers, distribution centres and customers, organisational levels, and shifts in responsibility within the organisation (Jacobs and Swink 2011).

This paper focuses on the complexity of products and processes only in order to limit the scope of the topic. Different definitions of product complexity can be found in the literature, and they relate to, among other areas, the number of functions offered by the product, the number of different core technologies incorporated into it, and the interfaces between its parts, functions, and technologies (Lakemond et al. 2013). In other words, product complexity increases the demand for resources and skills of the ones managing these. Process complexity has been defined as 'the degree to which a process is difficult to understand or carry out' (Muketha et al. 2010), which is also the definition used in this paper.

Increasing complexity is considered a major cause of the rising costs and deterioration of operational performance, leading in particular to decreased quality, long delivery times, delayed deliveries, and low process flexibility (Mariotti 2008). Therefore, companies need to gain an awareness of their levels of complexity and determine how this affects their competitiveness. However, many companies find it difficult to identify and quantify the most important costs of complexity and to prioritise possible initiatives to reduce complexity. In this vein, this paper raises the following question: *How can industrial companies reduce product and process complexity?* To address this research question, this paper proposes a procedure that aims to reduce product and process complexity. This is achieved by identifying product complexity costs, the basis on which product complexity can be reduced, as well as the processes used to handle this product complexity.

*Corresponding author. Email: lahv@dtu.dk

The costs of product and process variety can be reduced in various ways – for example, by increasing part commonality, postponing the point of product differentiation (Olhager 2003; Forza, Salvador, and Trentin 2008; Trentin et al. 2011), and lowering setup and changeover costs (Closs et al. 2008; Jacobs and Swink 2011). However, as the actual complexity costs and the potential for reducing them vary from company to company, it can be difficult to provide general recommendations. Therefore, in this paper, cost is applied as the universal metric to relate complexity cost factors and initiatives to each other. The five-step approach presented in Hansen et al. (2012) is further developed in this article.

The remainder of this paper is structured as follows. Section 2 discusses the literature on complexity identification, quantification, and reduction. Based on this discussion, Section 3 defines a procedure for product and process complexity reduction. Section 4 describes the research method, and Section 5 presents the case study to which the procedure was applied. Section 6 discusses the lessons learned from the application of the suggested procedure, and Section 7 presents the conclusions.

2. Literature review

The purposes of this literature review are (1) to define the field towards which the paper's contributions are aimed and (2) to demonstrate that there is a relevant gap in the literature. Thus, this section focuses on the literature related to product and process complexity, the quantification of complexity costs, and the strategies used for complexity reduction.

2.1. Product complexity

Product architecture is widely recognised as a crucial determinant of product complexity (ElMaraghy et al. 2013), and product architecture management enables the efficient design of new products that are targeted at individual market requirements. Furthermore, product architecture is considered a means of controlling the structure of the product assortment and the number of product variants, both of which affect the performance of sales, engineering, the production/supply chain, distribution, after-sales service, and so on (Meyer and Lehnerd 1997; Thumm and Goehlich 2015; Olivares Aguila and ElMaraghy 2018). Additionally, the literature on design for manufacturing discusses how to control product variety (e.g. Ulrich and Eppinger 2008; Arashpour et al. 2015).

A major cause of the increasing complexity in manufacturing environments is product variety (Schaffer and Schleich 2008; MacDuffie 2013). In this context, Wildemann (2001) performed an empirical study in the manufacturing industry to examine how the number of product variants affects unit costs. A comparison of a traditional manufacturing system with one that had flexible automated plants showed that with double the number of product variants in the production programme, the unit costs would increase by about 20–35% for industries with traditional manufacturing systems, while in systems with segmented and flexible automated plants, the unit costs would increase by only 10–15%.

A number of approaches and techniques have been proposed to control product architecture. For example, Meyer and Lehnerd (1997), Xie, Yang, and Tu (2008), Lindemann, Maurer, and Braun (2010), Mortensen et al. (2010), Haug, Hvam, and Mortensen (2013) and Zheng, Liu, and Xiao (2018) have suggested approaches that deal with the implementation of product architecture and the reduction of complexity in the product range.

Several techniques for modelling product assortments have also been outlined (e.g. Ericsson and Erixon 1999; Hvam 2001; Lu, Petersen, and Storch 2007; Yang et al. 2008; Haug 2010; Chiu, Chu, and Chen 2018). In this context, Haug (2013) and Hvam et al. (2018) have investigated the effects of applying such modelling techniques. Such product models help in managing the complexity induced by product variety and in many cases are implemented into product configurators (i.e. expert systems aimed at supporting sales or engineering activities) so that the knowledge they contain can be exploited in an automated way (Zhang 2014; Hvam et al. 2018; Kumar 2018).

2.2. Process complexity

ElMaraghy and Urbanic (2003) identified two factors that increase complexity: (1) the number and diversity of the features to be manufactured, assembled, and tested and (2) the number, type, and effort of the tasks required to produce the features. Samy and ElMaraghy (2012) defined complexity as 'a measure of how product variety can complicate the production process'. Similarly, Sivadasan et al. (2002) described two types of complexity in the supply chain: structural complexity (which increases with the number of elements) and operational complexity (which increases with the uncertainty of information and element flows).

Complexity in business processes is closely related to the complexity of the product assortment. Therefore, product architecture decisions can be used to control not only the complexity in product assortment but also the cost and performance of business processes (Lindemann, Maurer, and Braun 2010). Jacobs and Swink (2011) reviewed the existing research, based on which they defined a three-dimensional model that describes the nature of product portfolio complexity. In addition, ElMaraghy et al. (2012) discussed the types of complexity involved in products, engineering, and manufacturing. Several researchers in the field of supply chain management have also suggested various methods of analysing and controlling complexity from a supply chain viewpoint (Perona and Miragliotta 2004; Choi and Krause 2006; Wu, Frizelle, and Efstathiou 2007; Bozarth et al. 2009). Also, the literature on lean manufacturing (e.g. Shah and Ward 2007) discusses how to apply it to improve production efficiency, but there is little focus on how product complexity would impact production.

Blecker et al. (2004) described how to apply mass customisation to eliminate the process complexity caused by the increasing variation in the product architecture, inventory, and order-taking process, and they discussed the relationships between mass customisation and complexity. On the one hand, when applied as a pure customisation strategy, mass customisation increases product variety, which results in high planning and scheduling complexity; on the other hand, as the customer-order decoupling point moves towards the front end, mass customisation reduces product configuration and inventory complexity (Blecker et al. 2004). The degree of product customisation impact complexity in several ways since it deeply impacts the organisational design for mass customisation (Sandrin, Trentin, and Forza 2014). However, a company in choosing its degree of product customisation is influenced by several factors and in particular the demand dynamism (Sandrin 2016).

2.3. *The quantification of complexity costs*

Activity-based costing (ABC) suggests the allocation of overhead costs to individual activities. Cooper and Kaplan (1988), among other researchers, proposed ABC as a method to avoid the deficiencies of the arbitrary allocation of overhead costs. ABC first allocates indirect costs to the activities performed using shared company resources and thereafter assigns these to individual orders, customers, or even products.

Anderson and Kaplan (2007) proposed an efficient cost-modelling principle called time-driven ABC (TD-ABC) that assigns resources (e.g. the costs of a customer service department) directly to cost objects (e.g. order handling) and connects these to the unit times for performing transactional activities in order to achieve a simple cost rate measure that is based on time consumption. Park and Simpson (2008) described a method of applying ABC in the early stages of product development, which focuses on product families.

Lechner, Klingebiel, and Wagenitz (2011) proposed the use of variety-driven ABC (VD-ABC) to quantify the impact of adding or removing product variants in automotive logistics, based on the use of hypothetical zero-variant initiatives. This method is an expansion of the TD-ABC framework, which enables the calculation of incremental complexity costs associated with variants in different logistical operations. Jacobs (2013) discussed possible metrics for measuring complexity. Zhang and Thomson (2018) described how to apply knowledge-based measures of product complexity. Muketha et al. (2010) identified the metrics that are applied to measure process complexity.

The aforementioned methods of cost allocation distribute overhead costs to specific activities, but they do not focus solely on the most important costs; this means that a vast amount of data is needed and that undertaking the calculations requires considerable work. Rogozhin et al. (2010) thus suggested a method of adjusting the allocation of indirect costs in the automotive industry to obtain a more accurate estimate of the costs of adding new technology. Zhang and Tseng (2007) proposed a modelling approach to analyse the cost implications of product variety in mass customisation by bridging product variety and process variety; this was done by identifying cost drivers within the product design and including the manufacturing costs. Orfi, Terpenny, and Sahin-Sariisik (2011) proposed a set of product complexity dimensions (variety, functionality index, structural index, design index, and production index) and, along with these a set of associated indicators that consider the cost impact of the product complexity dimensions. Wan, Evers, and Dresner (2012) outlined the impact of product variety on operations and sales performance, using unit fill rate as the measure.

2.4. *Strategies for reducing complexity costs*

The calculation of complexity costs is an area that is of particular interest in this research because the aim is to rationalise a product programme with a view to allocating the true complexity costs to the product variants (Hansen et al. 2012). Several researchers that discuss the frameworks for assessing product profitability and cost behaviour have been identified in this field (Sivadasan et al. 2002; Zhang and Tseng 2007; Mariotti 2008; Wilson and Perumal 2009; Wang et al. 2011; Wan, Evers, and Dresner 2012; ElMaraghy et al. 2013). For example, Wilson and Perumal (2009) offered several top-down approaches to address interrelated product–process organisational complexity from a managerial perspective by dividing complexity costs into those that are value adding (good complexity) and those that are non-value adding (bad complexity), and Closs et al. (2008) and Jacobs and Swink (2011) provided a list of possible strategies for reducing complexity. Additionally, a

Table 1. Summary of literature.

	Contributions	Gaps
Product complexity	Product architecture as a means of controlling product variety Examples of impact of increased product variety on production costs	How to identify the most critical drivers of complexity cost between product variety and process complexity
Process complexity	Examples of how product variety impacts the costs of production processes and of how, e.g. mass customisation strategies can improve the ability to handle increased product variety in these processes	Methods of identifying the most important product variety factors that impact process complexity
Quantification of complexity costs	Examples of allocating overhead costs to products and production processes	Focus on only the most important complexity cost drivers in order to reduce the amount of data and number of resources needed for the analysis
Strategies for reducing complexity	Approaches to the top-down management of product-process and organisational complexity; bottom-up approaches to reducing product variety	Identify potential areas for improvement based on the quantification of the most important complexity costs

number of product variant rationalisation approaches focus on different ‘tail-cutting’ methods (often called stock-keeping unit [SKU] rationalisation), some of which show that no relationship exists between the number of SKUs and the market share (Mahler and Bahulkar 2009). Table 1 summarises the literature.

From the literature review, it emerges that a number of approaches have been proposed for modelling and controlling complexity in product architecture and for analysing complexity in business processes. In relation to the quantification of complexity costs, ABC provides methods of allocating overhead costs to specific activities, and several approaches are used to assess the cost of product complexity. However, to the authors’ knowledge, the existing literature fails to provide simple and operational methods of identifying the most significant complexity costs of products and processes and of quantifying this complexity using limited data and resources. Furthermore, the literature that discusses possible strategies for complexity reduction does not link this discussion to specific and simple methods of identifying and quantifying complexity costs or identifying the most relevant initiatives for reducing complexity costs in products and operations.

3. A procedure for reducing product and process complexity

The following five steps for reducing product and process complexity were developed based on the literature review:

(1) Define the scope of the products and processes to be included in the analysis.
(2) Conduct ABCC (A, B and C categorisation) analysis of products.
(3) Identify and quantify the most significant complexity cost factors.
(4) Identify and quantify possible initiatives for the reduction of complexity costs.
(5) Evaluate and prioritise initiatives to establish a complexity cost reduction programme.

These five steps are described in the following subsections.

3.1. Step 1: defining the scope of products and processes to be included in the analysis

The first step involves delimiting the analysis by determining which products to include in it and establishing whether it should focus on the finished-goods level only or should also include module levels in the product assortment (George and Wilson 2004). Furthermore, the scope is defined in terms of which parts of the process flow to include in the analysis – for example, sales, production, sub-suppliers, sales distribution, and after-sales service. When the scope is determined, the products are described using posters that show the features that are relevant to understanding the differences between the product variants – for example, capacity, power supply, dimensions, and colours (Hvam, Mortensen, and Riis 2008; Yang et al. 2008).

To restrict the amount of data and time necessary for the analysis, it is advisable to select a limited number of product families for inclusion, and they should be produced at a limited number of production sites and perhaps sold in a limited number of regions worldwide. It is possible to include product families from high- and low-end market tiers, providing there is adequate similarity in the ways in which the products are manufactured and handled internally to enable them to be analysed concurrently.

3.2. Step 2: conducting an ABCC analysis of products

The second step, ABCC analysis of products, split the products into A, B, and C categories. The purpose of applying this type of analysis is to identify (and, later, possibly eliminate) product variants that contribute only minimally to revenue but imply significant additional complexity. As a rule of thumb, the Pareto distribution can be used to separate the product variants into the three categories (Koch 2008). In practice, the statistical analysis of the revenue data could give rise to more advantageous distributions. However, if it is assumed that the Pareto distribution makes sense in a particular setting (if not, then the percentages should be adjusted), the products in category A are the variants that contribute to 80% of the contribution margin (i.e. product price minus all associated variable costs), the B products are the variants contributing the next 15% of the contribution margin, and the C products are those accounting for the remaining 5%. The contribution margins are calculated as 'contribution margin 1', which is defined as sales price minus direct production costs. When undertaking the double Pareto analysis, some products fall outside the three categories. These products are listed separately and are put into one of the three groups, based on an assessment of where they would have the strongest clustering.

A similar analysis can be undertaken for the customers of the product range to determine which ones are the most profitable. Therefore, for each customer (or group of customers), the contribution margin and the revenues are plotted in a diagram in the same way as described for products (George and Wilson 2004; Wilson and Perumal 2009).

The revenues of the included product variants are determined by collecting the realised revenues of all the variants from all the sales of the companies within a given period (e.g. 6, 12, or 18 months). For this, the revenues reported might have to be adjusted for deviations arising from customs, currencies, and discounts. The direct production costs (including those related to materials and wages) should be reported directly from the factories. Based on this, a contribution margin (named contribution margin 1) for each product variant is calculated by subtracting the cost per item from the revenue per item. The revenue, cost, and contribution margin 1 are added to the poster described in step 1, and the contribution margin 1 and revenue for each product variant are plotted in a diagram with revenue on the horizontal axis and the contribution margin on the vertical axis (Wilson and Perumal 2009). Both axes use a logarithmic scale.

3.3. Step 3: identifying and quantifying the most significant complexity cost factors

The purpose of step 3 is to identify and quantify the most significant complexity cost factors (i.e. fixed costs in the cost distributions, each of which accounts for more than, for example, 1% of the total cost) with an uneven distribution of costs between the product variants (Park and Simpson 2008; Wilson and Perumal 2009). This step is important because it directs focus towards the analysis, avoids wasting resources on analysing superfluous data, and identifies the most significant complexity drivers that should be addressed when identifying initiatives for reducing complexity.

The starting point is to brainstorm on possible complexity cost areas that have uneven cost distributions. Examples of cost areas with asymmetric cost distributions are inventories (C items may be in stock longer than A items), setup costs (C items may be produced in smaller batches than A items, leading to relatively higher setup costs for C items than for A items), or administrative costs for sales order handling (C items may be sold in smaller batches than A items, leading to relatively higher costs for the sales order handling of C items than A items).

3.3.1. Identifying the most significant complexity cost factors

The most significant cost factors can be identified through the cost distributions for the product families and the lists of possible complexity cost factors in the analysis. In this context, a list of possible cost factors to be used as the basis for a brainstorming process can be found in the literature (e.g. Closs et al. 2008; Jacobs and Swink 2011; Myrodia and Hvam 2015). The list includes the cost factors incurred in sales, the production/supply chain, product development, and distribution. Examples of these cost factors are costs of sales order administration, warranty costs, costs of setting up production, costs of inventories, and handling costs in distribution centres.

3.3.2. Carrying out analysis, quantification, and allocation

Having identified possible significant complexity cost factors, the next step involves analysing the complexity cost factors and finding quantification objects that allow for approximations of the indirect costs to allocate them directly to product variants, where applicable. By dividing all costs by the net revenue recorded for each variant, all costs are comparable as percentages. It is often necessary to settle for incomplete data extracts and be creative in applying unconventional quantification objects to develop reliable approximations (inspiration can be found in Anderson and Kaplan [2007] and Lechner, Klingebiel, and Wagenitz [2011]).

If it is possible to find data to support the quantification of the identified complexity cost factors and if the analysis shows an unambiguous uneven distribution of the costs, the complexity cost factors are used to adjust the contribution margins and contribution ratios for each product variant. The costs allocated from the analysis and quantification of the complexity cost factors can be accumulated to provide an overview of the complexity-adjusted contribution ratios (%) (as well as the complexity-adjusted contribution margins [EUR]).

3.4. Step 4: identifying and quantifying possible initiatives for reducing complexity costs

Based on the insights from steps 2–3 and on Jacobs and Swink (2011) and Mortensen et al. (2010), it is possible to generate different initiatives for reducing complexity costs by changing the product range (e.g. reducing the number of product variants, adjusting prices, adjusting the product variants offered to each market, redesigning modules, and changing the product architecture) or by making changes in the business processes (e.g. reducing setup times, changing the order-decoupling point and points of stock, and changing delivery times).

The suggested initiatives are grouped into short-term, mid-term, and long-term initiatives (Wilson and Perumal 2009). Short-term initiatives include adjusting the product assortment using the contribution ratios, as described in step 3. Often, a thorough analysis reveals several low-hanging gains and increased earnings before interest and tax (EBIT). In this context, it is important not to assume 0% substitutability of discontinued variants (rare cases only) but to apply a more realistic number to estimate the true incremental revenue loss. Other examples of short-term initiatives include minor process adjustments (e.g. reducing setup times and adjusting settings in the ERP system for planning and stock management). Examples of mid-term and long-term initiatives are redesigning products to improve variant creation or making changes to production flow or stock points. For each initiative, the quantification of possible savings is carried out based on the identified complexity cost factors and an estimate of the project costs for implementing the suggested initiative.

3.5. Step 5: evaluating and prioritising initiatives for reducing complexity costs

Step 5 involves evaluating the initiatives and making a plan for their implementation. The initiatives selected for implementation are divided into short-term, mid-term, and long-term initiatives. The insights obtained from the complexity analysis are reflected upon, and consideration is given to how they may be used, for example, in product planning, product development, and the ongoing development of business processes.

Therefore, step 5 aims to implement the findings from steps 2–4 in a complexity reduction programme. The central aspect is identifying the drivers of the complexity cost factors to create complexity costs. The aim is to work with these drivers to reduce their negative impact on complexity. An example might be the country-specific customisation of product variants that could be solved differently. The cost transparency achieved in step 3 can serve to justify the cost of changing the country-specific feature or solution.

The complexity reduction programme contains the initiatives aimed at diminishing the negative effects of the complexity cost factors (e.g. by postponing the point of product differentiation) and initiatives for working actively with the complexity cost drivers to eliminate them or reduce their negative effects.

3.6. Application of the procedure

The proposed steps of the procedure could be adapted to a specific company with regard to the level of product analysis (at the finished-goods level only or including modules and component levels in the product assortment), the product life cycle processes of the products to be included in the analysis (e.g. product development, sales, production, assembly, and distribution), the level of detail of the ABCC analysis (i.e. single products or groups of products), the inclusion of customers in the ABCC analysis, and the use of metrics other than costs to quantify the impact of complexity (e.g. flow in production, on-time delivery, lead time, and quality of products). Finally, given the need for affordable support, the proposed procedure has to employ data that are reasonably accessible and executable within a limited period and with limited use of resources.

4. Research method

To investigate the usefulness of the proposed procedure, a case study of a company using it was carried out. The study aimed to establish whether the procedure helps to identify significant areas of complexity, whether the data required for the detailed analysis and quantification are accessible, whether the analysis can be executed with a limited number of resources, and whether the results of the analysis provide an empirical basis for the generation of complexity-reduction initiatives.

The single case study can be described as having a holistic, representative design with a single unit of analysis (the case company) (Yin 2009). The case is representative because the company is typical of many major manufacturers that have had problems managing product and process complexity, which is also the main sampling criterion. As this type of case study methodology pertains to a single case, it is possible to generate only an analytical generalisation, as opposed to a statistical one (Yin 2009).

A project team was formed in an industrial company involving a controller and a product developer from the company (who spent 50% of their time working on the project), as well as two consultants. Managers from all relevant departments also participated in three workshops during the course of the project, which lasted for four months. The role of the consultants was to provide the methods for analysing and quantifying complexity costs, arrange workshops, extract and analyse data, and contribute to discussions of future initiatives to reduce complexity costs.

The entire project was followed by one researcher. The research method consisted of document studies, observations, and semi-structured interviews. These methodological steps were carried out throughout the study, as opposed to in different stages. The interviews were conducted with the managers of four departments and the participants from the product team. This method was chosen because the investigated data are relatively unstructured and their analysis involves explicit interpretation (Silverman 2005). Using semi-structured interview protocols gave the interviewer the flexibility to focus on what the company believed were the most important problems. Notes about events were taken over the course of the entire project, and the obtained information was cross-checked with the participants in the project. To minimise bias as the greatest extent possible, triangulation, in the form of a combination of interviews, direct observation, documentation, and participant observation, was carried out (Yin 2009).

5. Case study

The procedure was applied at a leading global manufacturer of mechanical consumer products with 5200 employees worldwide and an annual turnover of approximately EUR 900 million. The products are manufactured to stock and distributed via regional distribution centres. The scope and data used in the study are further described in the following sections.

At the time of the study, the company was undergoing an increase in the number of product variants. The management considered this increase to have a negative effect on company performance and, thus, focused its attention on reducing the number of product variants. However, the company did not know the specific costs of having the increased number of product variants, and it lacked a means for identifying the products that should be removed from the product assortment to reduce the complexity costs of its operations.

The company's large scale combined with the lack of systems to assess the cost implications of the complexity induced by the product variety made it a particularly interesting setting for testing the procedure. On the one hand, the managers were extremely interested in using the procedure, as the potential contribution was exactly what they were looking for. On the other hand, the scale of the company would make it difficult to find information that had not previously been stored in a structured form, such as the cost of the complexity.

5.1. Step 1: defining the scope of the analysis

The analysis focused on one of the product groups covering consumer goods sold in more than 40 countries worldwide. The group of products was chosen because it had a low total profitability and a long tail of low-selling products. To adjust for seasonal variations, the analysis was based on the sales data for the latest 12-month period. Furthermore, products with no sales and those that had been released but not sold in the period were excluded from the analysis. The final scope included approximately 350 item numbers with an annual turnover of around EUR 40 million.

The scope was decided in cooperation with the product managers, who provided insight into the product's technical features and the market/customer base. The products in the scope were manufactured in two different factories and distributed via three regional distribution centres. To distinguish between the 350 product variants, a list of descriptive characteristics (i.e. name, product family, and part number) and a list of product characteristics (i.e. capacity, type of nozzle, type of filter, cable, and voltage) were added to each item number. These characteristics were added by the R&D department and assessed by the product managers. Furthermore, data on the release date, the factory in which the item was produced, and the region in which it was sold were added to the list.

5.2. Step 2: conducting the ABCC analysis of product profitability

To calculate the contribution margin of the product variants, the realised revenues of all 350 product variants from sales companies within the 12-month period were collected. The analysis included sales numbers from the top 30 sales companies,

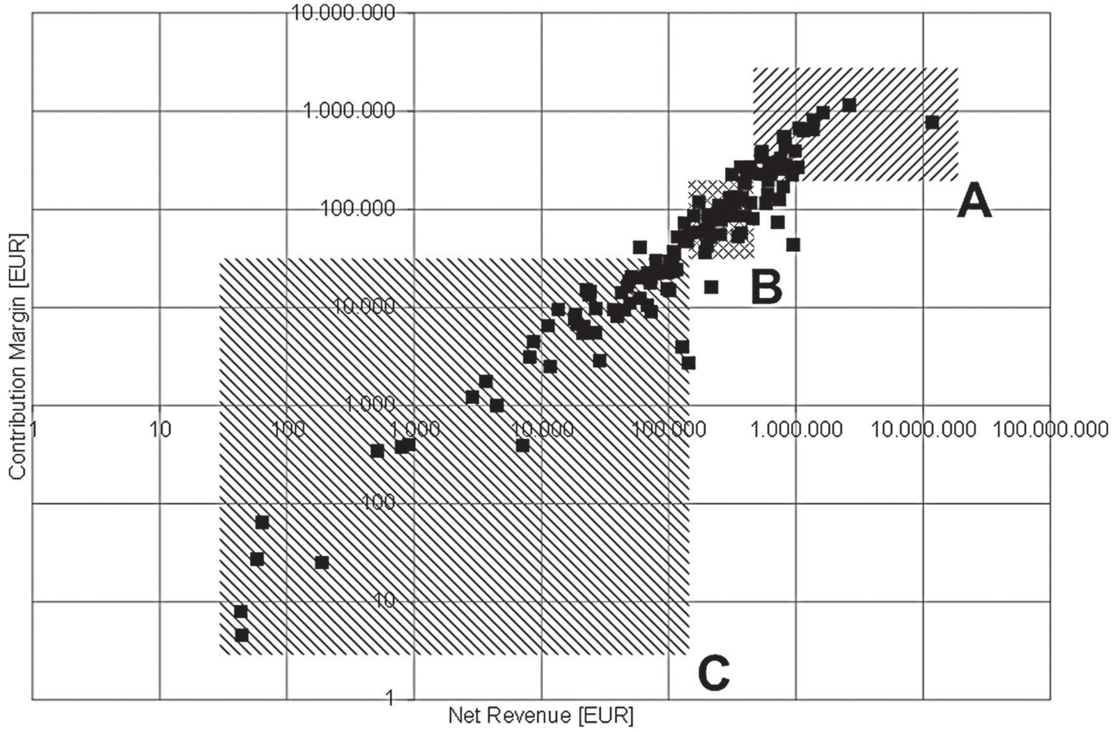


Figure 1. ABCC analysis of product profitability.

which cover approximately 98% of the sales; the remaining 12 sales companies account for less than 2% of the sales in this product group and were not included in the analysis. The sales revenues from the individual sales companies were adjusted for local bonuses, customs, and deviations in currencies. The direct production costs (including materials, wages, and other factory costs) for each of the 350 item numbers were subtracted from the sales revenue, resulting in a measure of the contribution margin. Based on this, an ABCC analysis was undertaken, as shown in Figure 1. The net revenue for each product variant is plotted on the horizontal axis and the contribution margins on the vertical axis, as explained in Section 4.

The ABCC analysis showed that 120 variants (34%) were C products, 110 were B products (31%), and 120 were A products (34%). Products that fell outside of the three categories were listed and attached to the B and C groups of products. In the analysis, the products were divided into four main categories, and each group was marked with different patterns in the diagram (see Figure 1). The ABCC analysis showed that some of the product groups had significantly lower contribution margins than the others (Figure 1). Figure 2 shows the contribution ratios (contribution margin relative to sales revenue) for the 350 items.

As can be seen in Figure 2, the contribution ratios for the 350 products in the study vary from close to 0% to more than 80% for the different items, which indicates significant potential to improve the contribution margin of the product portfolio.

5.3. Step 3: identifying and quantifying the most significant complexity cost factors

Based on the early indication of the product variants making the least contribution in step 2, the aim of step 3 was to identify and quantify the most significant complexity cost factors throughout the product life cycle and adjust the contribution margins for each item. The analysis started with a cost breakdown (top down) and a brainstorm (bottom up) of possible factors with asymmetric costs for the different product groups. The complexity cost factors were identified in cooperation with the product managers, as well as the sales, production, and distribution managers. The following possible complexity-related cost factors were identified:

- White-collar costs in factories
- Setup costs in factories
- Stocks of materials in factories
- Warranty costs
- Order-handling and administrative costs in distribution centres

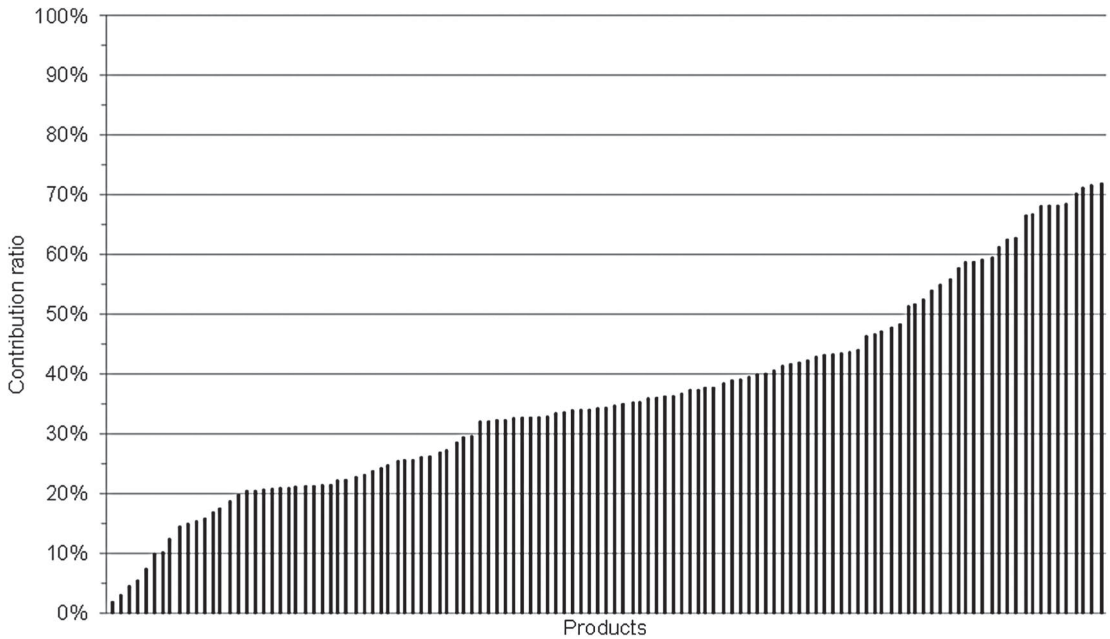


Figure 2. Contribution ratios for the products included in the analysis.

- Handling in distribution centres
- Inventory costs – finished goods
- Freight costs (inbound and outbound to distribution centres)
- Administrative costs in sales
- Advertising costs

To quantify these possible complexity-related cost factors, the necessary data were requested (e.g. the setup time and order quantities in production for different item numbers to calculate the setup costs for each item). The analysis showed a significantly asymmetric cost distribution for the following factors: 'inventory of materials in factories', 'handling in distribution centres', 'order-handling and administrative costs in distribution centres', 'finished goods inventories', 'inbound freight to distribution centres (from the factories)', 'outbound freight from distribution centres (to the customers)', and 'administrative costs in sales'. The other factors listed proved to be either insignificant (less than 0.5% of the turnover) or to have no asymmetric distributions of costs. Regarding quality costs, it was not possible to obtain complete data to analyse the costs of quality for each item number; however, the data available from three sales companies indicate an asymmetric distribution in the quality costs, which account for approximately 2% of the turnover. Based on this analysis, the company decided to implement more detailed reporting of quality costs from the sales companies.

Figure 3 shows an example of the adjustment needed because of the freight costs for the 350 products in the study. The freight costs were calculated based on an estimate of 5.2% (1.6% inbound, 3.6% outbound) of the product's sale price revenue for all 350 products handled in the distribution centres. The freight cost per item was calculated based on unit sales and actual figures for shipping quantities per container and pallet per item.

As seen in Figure 3, the freight costs vary between 1% and 9% for each item number. This variation is a result of the different filling of the pallets and containers, which is due, in part, to the volume of the boxes and partly to the order size shipped; large orders are shipped on full pallets and containers, and small orders are shipped on partly filled containers and pallets. C items tend to have higher freight costs than A items.

Another example is sales order handling, for which a time study showed that the average cost per order line is EUR 3.5. Based on this, the costs of sales order handling per product is calculated as follows:

$$\text{Sales order handling cost per product} = \begin{array}{c}\text{No. of orders per product variant} \\ * \text{ Sales order handling cost per order line}\end{array}$$

The data needed for the quantification of sales order-handling costs are as follows: number of orders per product variant and average sales order-handling costs per order line.

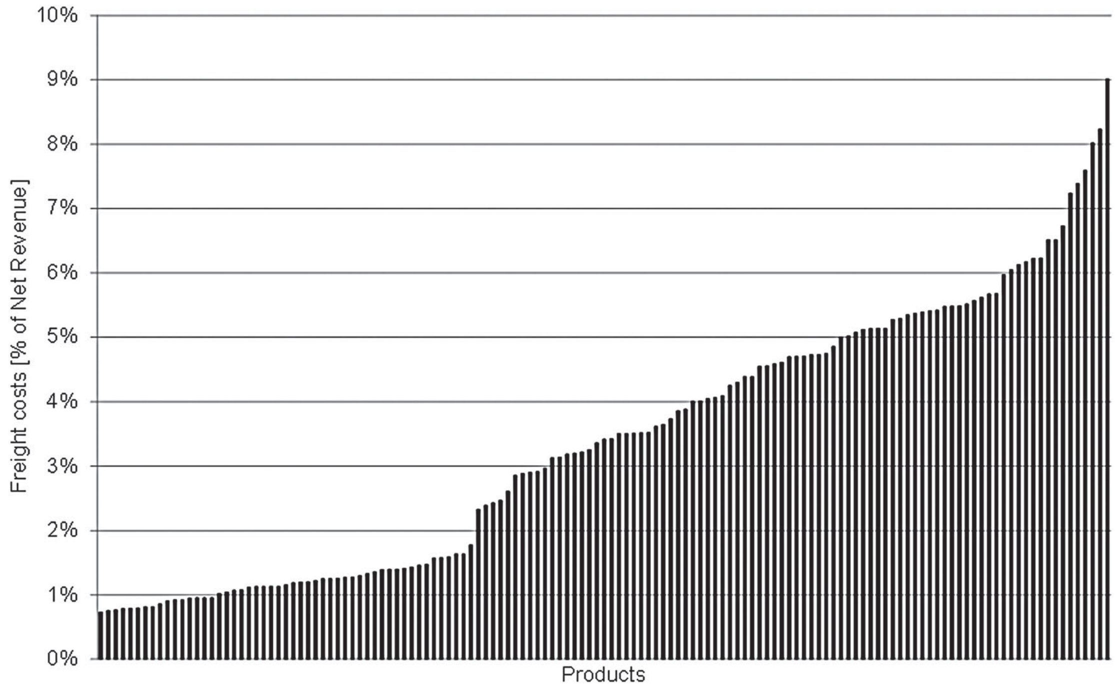


Figure 3. Contribution of freight costs.

Table 2. Example of order-handling costs.

Order frequency	[min]	Unit sale per year
High	5	> 5000 units
Medium	7	1500–5000 units
Low	12	< 1500 units

Another example is the order-handling costs in the warehouse. In this case, a study analysed the time used relative to the frequency of handling the product variant in the warehouse, as shown in Table 2.

The time per order is calculated as follows:

$$\text{Time per product} = \sum_{1}^{n} (\text{time per order} + (\text{order size} - 1) * 10 \text{ seconds})$$

In the calculation above, n is the number of orders in a year, and order size is the number of pallets in the order.

The data needed for calculating the time used per product in the distribution centre is the following: list of order lines in a one-year period, number of pallets per order and time used per order for the first pallet, and time used for the subsequent pallets (in this case, 10 s per pallet).

Based on the analysis of the complexity cost factors, the contribution margins and ratios calculated were adjusted using the exact cost of each complexity cost factor, which gives a more accurate value for the costs and contribution margins for each item number. Figure 4 shows the contribution margins and contribution ratios adjusted using the calculated complexity cost factors. Each line in Figure 4 represents a product variant. The first seven columns contain information on volume sold, revenue, production costs, and contribution margin and contribution ratio for each product. The next eight columns show the allocated complexity costs for each complexity cost factor, the sum of the complexity costs, and the allocated complexity costs relative to the revenue for each product. The last two columns show the adjusted contribution margin and the adjusted contribution ratio for each product after the allocation of the complexity costs.

As Figure 4 shows, the complexity cost factors change the contribution margins and cost ratios significantly, thus providing a more accurate calculation of the contribution margins and ratios for each item in the analysis. A further analysis of the true costs and margins of different product groups was undertaken.

Contribution margin and ration before complexity cost adjustment [EUR]							Complexity cost factors and total complexity cost adjustment [EUR]								Complexity adjusted contribution margin and ratio [EUR]	
Total Cost per unit	Avg. Unit Price	Unit Sales	Total Cost	Net Revenue	Contribution Margin	Contribution Ratio	Component inventory	Freight	EDC PO handling	EDC WH order handling	L6 Inventory	Sales order handling	Total complexity cost adjustment	Complexity cost adjustment relativ to revenue	Adj. Contribution Margin	Adj. Contribution Ratio
53	127	16.988	893.100	2.165.957	1.272.857	59%	6.288	192.823	2.997	93.902	107.883	31.867	435.760	20%	843.385	39%
69	170	10.669	740.821	1.813.329	1.072.508	59%	4.785	121.099	2.412	72.430	124.077	22.872	347.675	19%	729.618	40%
39	87	17.377	683.905	1.518.498	834.594	55%	623	65.745	1.681	26.379	16.489	8.992	119.909	8%	715.307	47%
32	85	16.434	522.066	1.404.192	882.126	63%	668	62.177	73	70.913	156.015	22.419	312.266	22%	570.529	41%
34	64	28.284	952.935	1.803.970	851.035	47%	823	107.011	1.316	89.191	89.775	30.317	318.432	18%	533.427	30%
52	164	6.436	335.813	1.054.305	718.492	68%	1.551	36.523	512	57.399	52.247	19.315	167.547	16%	552.496	52%
106	318	3.400	359.775	1.082.756	722.981	67%	4.605	51.449	1.462	44.202	70.170	12.754	184.642	17%	542.945	50%
32	112	6.412	206.750	717.181	510.431	71%	122	24.258	1.389	56.999	18.195	19.243	120.206	17%	390.347	54%
111	234	4.662	519.078	1.093.063	573.985	53%	4.691	70.553	1.023	61.484	132.528	17.490	287.770	26%	290.907	27%
126	397	1.765	222.730	701.205	478.475	68%	1.416	26.714	950	47.344	40.999	13.245	130.668	19%	349.223	50%
93	249	2.873	267.895	715.304	447.408	63%	4.943	43.476	3.582	4.024	27.260	980	84.264	12%	368.088	51%
32	83	6.982	224.697	580.231	355.534	61%	233	21.459	2.339	6.742	65.355	1.435	97.563	17%	258.204	45%
38	134	3.102	118.182	416.029	297.846	72%	690	11.736	2.632	3.360	48.943	517	67.879	16%	230.658	55%
95	231	2.342	223.144	541.753	318.608	59%	2.688	35.449	2.778	52.476	37.733	17.575	148.699	27%	172.597	32%
31	65	9.232	288.189	597.160	308.971	52%	211	28.374	2.705	68.722	29.144	23.089	152.245	25%	156.937	26%
69	216	1.063	72.935	229.514	156.579	68%	-1.964	6.034	219	29.852	9.748	7.977	51.866	23%	102.749	45%
183	398	526	96.253	209.214	112.961	54%	-1.558	7.954	744	11.417	46.741	3.944	69.242	33%	42.161	20%
184	586	134	24.786	78.705	53.919	69%	-4.347	2.034	177	6.605	14.045	1.008	19.522	25%	30.050	38%
69	122	28.586	1.965.884	3.478.146	1.512.262	43%	11.881	230.545	2.360	26.206	187.191	43.696	501.879	14%	1.022.263	29%
52	87	15.018	786.191	1.300.352	514.160	40%	2.599	60.559	1.899	39.345	70.715	49.853	224.970	17%	291.789	22%
182	352	2.304	419.045	811.166	392.120	48%	7.894	26.057	1.324	24.357	37.913	11.707	109.252	13%	290.763	36%
120	427	1.146	137.509	489.089	351.580	72%	2.004	12.323	58	1.041	40.541	593	56.560	12%	297.024	61%
41	71	13.709	566.334	966.845	400.511	41%	321	44.482	1.036	23.385	43.152	43.490	155.867	16%	244.966	25%
111	196	5.124	569.037	1.003.423	434.386	43%	3.942	55.099	403	14.983	151.556	13.912	239.896	24%	198.433	20%
177	298	3.030	536.497	903.613	367.116	41%	13.282	34.268	1.151	24.251	57.483	19.759	150.195	17%	230.203	25%
38	51	26.638	1.019.898	1.371.706	351.807	26%	777	57.622	1.094	3.547	28.457	4.816	96.312	7%	256.272	19%
117	185	5.311	621.885	983.218	361.334	37%	3.773	57.112	806	23.593	27.337	33.350	145.971	15%	219.135	22%
117	179	6.113	716.276	1.091.377	375.100	34%	9.830	65.732	748	18.198	62.522	17.215	174.245	16%	210.685	19%
38	59	14.128	537.910	839.813	301.904	36%	113	30.561	2.820	15.712	83.415	21.854	154.475	18%	147.541	18%
68	103	9.623	653.982	991.456	337.474	34%	1.283	77.606	1.842	17.340	62.841	24.641	185.553	19%	153.204	15%
104	184	3.737	386.830	686.749	299.919	44%	1.295	40.182	2.072	29.087	57.373	15.326	145.336	21%	155.878	23%
33	52	16.183	530.889	847.661	316.773	37%	2.044	43.505	2.187	49.339	39.372	48.445	184.892	22%	133.924	16%
100	188	2.818	283.097	530.869	247.772	47%	2.547	30.298	2.130	24.858	18.467	11.973	90.273	17%	160.046	30%
56	83	9.660	544.848	802.130	257.281	32%	491	47.017	173	5.100	53.379	7.360	113.520	14%	144.252	18%

Figure 4. Contribution margins and ratios adjusted using complexity cost factors.

Based on the adjusted contribution ratios, the profitability of the 16 product families was analysed. The analysis showed that an old product family, which the company had wanted to withdraw from the market for years, had a turnover of approximately EUR 5 million and an adjusted contribution ratio of 60%. Another newly launched product family, for which the company had high expectations, had a turnover of EUR 500,000 and an adjusted contribution ratio of only 8%. Based on this new information, the company considered relaunching the old product family and redesigning the newly launched product family.

5.4. Step 4: identifying and quantifying initiatives for complexity reduction

Based on the analysis of complexity cost factors, in this step, initiatives for reducing the complexity of the products and processes were developed, and the potential savings were quantified. The following are possible initiatives for reducing complexity:

- Adjust the product line based on an analysis of product variants, price points, and contribution margins per country.
- Optimise variance creation (products and accessories are packed in distribution centres rather than in factories).
- Reduce the number of components kept in stock in factories.

- Introduce direct shipments from the factory to the customers for low-selling products.
- Adjust bonus agreements per country/customer/product group.

The following ideas were also identified for further analysis:

- Conduct complexity analysis of spare parts and accessories across product categories.
- Analyse the cost of certificates and the possible relationships to product platforms and OEM customers.
- Conduct OEM analysis (improving profitability for OEM customers).
- Investigate optimising the R&D process to include freight costs to a greater extent.
- Implement a phasing-out strategy – alignment of product management, product development, and production when phasing out products to avoid producing obsolete components and products.
- Analyse the trade-off between lower stock cost vs. high purchase order-handling cost and changeover cost.

The potential savings from each initiative were calculated based on the quantified complexity cost factors. An example of this is the first initiative in which the product line is adjusted based on an analysis of the variants, price points, and contribution margins per country. For this, eight initiatives for each brand in the product group were evaluated, which cover high or low levels of product pruning, full or no product substitution, and high or low levels of positive price adjustment. Recognising the different challenges faced by each brand, decisions were made for each brand separately with regard to the following issues: which items to exclude, which ones could be substituted with others, and the extent to which the prices were to be adjusted. The initiatives yielded new overall net revenues, contribution ratios, and contribution margins for each brand. The quantified scenarios indicate an increase of between 2% and 4% in portfolio profitability (increased contribution margin), corresponding to an increased EBIT of between EUR 800,000 and EUR 1.6 million. Where applicable, the costs that were dependent on the number of variants were subtracted in the scenarios (e.g. freight costs, component inventory in factories, and finished goods inventory in the distribution centres). The cost of undertaking the analysis and implementing the revised product assortment was estimated to be EUR 100,000. Similar quantifications of possible savings and project costs were made for each initiative.

5.5. *Step 5: evaluating the initiatives and the insights gained from the procedure*

In step 5, the suggested initiatives were evaluated and prioritised based on the quantification derived from the complexity analysis. This included an assessment of the strategic impact of the suggested initiatives. Based on the complexity analysis, the company decided to implement the following three projects to reduce complexity:

- Adjust the product line, as described in the previous section
- Reduce complexity costs in the factory/supply chain
- Change the order-decoupling point by shipping accessories to the distribution centres and undertaking the final configuration of products and accessories in the distribution centres rather than at the factory.

The third project led to a significant reduction in finished goods inventories in the distribution centres. A number of minor changes, such as adjusting settings in the ERP systems to manage inventory levels and procedures for handling orders in the distribution centres, were also made.

The application of the procedure had implications for several functions and directors of functions. More specifically, the analyses were utilised in the following areas:

- For the product managers, the project provided detailed insight into the profitability of each finished goods item. This insight was used to trim the product portfolio in the first implementation project mentioned and is currently used in negotiations between the product manager and the sales representatives in each country to decide which variants to promote in the portfolio for each country and price setting.
- For R&D, the project provided new and more detailed insight into the cost structure of product variants and the costs of sales, production, and distribution. This insight is of significant value when R&D department develops new products and makes decisions about which product variants to include in the new product portfolio and which modules to include in the product architecture.
- For the managers of sales, production, and distribution, the complexity analysis provided insights into how costs are allocated for different groups of processes in business areas. This information has led to an increased focus on complexity costs and to the initiation of projects, leading to lower costs of handling complexity in sales, production, and distribution.

Furthermore, to ensure that the findings would be implemented in the daily business, the company decided that the ABCC analysis should be conducted every six months and submitted to the product manager. The ABCC analysis should be adjusted using the complexity cost factors found in the first analysis.

5.6. Use of the procedure after the test period

The project was implemented within four months, using approximately 700 man-hours, excluding consultant time. Since the completion of the project, the company has initiated and completed three other projects on complexity analysis, using the procedure on other groups of products. These projects have been carried out by internal employees using the competences they gained from the first project, but with some supervision from the researchers. Based on the experiences gained from these projects, the employees can undertake the data extraction and the calculations, but they still need support to identify the complexity cost factors, interpret the data and findings, and determine possible initiatives for reducing complexity. The employees also need support when checking and validating the data and findings – that is, when asking critical questions regarding the accuracy of data, calculations, and findings, as well as double-checking all data and findings.

6. Discussion

The aim of the proposed procedure was to identify the most significant complexity costs in the system of products and processes, quantify these costs, and allocate them to individual products and process steps. On this basis, it is possible to conclude whether the potential for complexity reduction lies in the product domain or in the supply/production/process/delivery domain or whether a coordinated redesign of the two domains is needed.

Table 3 lists the major difficulties that the case company encountered and its requests for assistance during the project, as well as solutions that were found to work and can, thus, be recommended for future application of the procedure.

Table 3. Major difficulties/requests and recommended solutions.

Difficulties encountered and specific requests	Solutions
The accuracy of the data and findings were questioned by employees at all levels of the company.	It is crucial to have trustworthy data and findings. Thus, experienced employees should be involved in undertaking a critical assessment of data and calculations. Furthermore, it should be ensured that data and calculations are transparent and easy to understand; and all data and calculations for possible errors should be checked (possibly double-checked) and validated.
Obtaining data to quantify complexity cost factors was a challenge with regard to certain factors.	Project managers should identify sources of data with production, distribution, and sales managers. If a complexity cost factor cannot be quantified due to a lack of data, begin temporary registration of these data, and if the results are interesting, implement permanent registration of these data.
Finding potential initiatives to reduce complexity was crucial to ensuring business impact from the analysis.	Workshops generate fruitful discussions and numerous ideas from area managers and others. It is important to use experiences from other companies to enable the consideration of as many possible solutions as possible.
The company requested to use the findings in ongoing projects and in its daily business.	There was significant interest in the findings from managers at all levels. The analysis needs to be updated constantly to include findings from the current product-planning process. The company has decided to update the ABCC analysis every six months for the product managers and for use in R&D and ongoing process improvement projects.
The company demanded that it should be able to apply the procedure after the project period.	The researchers supervised the working team during the procedure. The team members were then able to undertake data extraction and calculations. Future projects will benefit from having a person on the team who has been trained in the procedure and has experience with similar projects in other companies.
The company required that the analysis be completed within the scheduled four-month period and without exceeding the resources that were assigned for the project.	These requirements were met. It is crucial to have a realistic scope for the analysis and to constantly limit data gathering and calculations to ensure that only what is needed is included. Experienced employees should be involved in discussing the scope and focus of the analysis.

The case study revealed a series of difficulties and further requirements related to the suggested procedure, which led to a list of recommendations for applying the procedure to future projects (Table 3). Further applications will lead to additional insights that may give rise to improvements in the procedure and may generate a more detailed description of how to adapt it to individual company settings. An important aspect of learning from the project is that it is crucial to use significant resources to check and validate all data and calculations, thereby ensuring that the data are correct and unquestionable. Furthermore, it is essential for the interpretation of the results to be focused on the analysis of the most significant areas and for the findings to be presented and discussed with the managers of the company.

The results of the complexity analysis were presented and discussed with the group of production, distribution, R&D, and sales managers. During the first three phases of the project, the project team identified possible initiatives for reducing complexity when interviewing and discussing the findings with individuals in the organisation. These initiatives were discussed and further elaborated with the managers during workshops in which the complexity analysis and the findings were discussed. The knowledge and experience of the company's managers and employees were used to focus the analysis of complexity costs on the most significant cost elements and on the identification of initiatives to reduce complexity. This approach enabled the delimitation of not only the data but also the resources needed for the data analysis and calculation.

A vital aspect of the procedure is the identification and quantification of the company's most significant complexity cost factors. In that respect this procedure contributes to the Suzić, Forza, et al. (2018; Suzić, Sandrin, et al. 2018) call for as-is analysis tools to help practitioners to implement mass customisation. Such insights contribute to the literature on process complexity (Jacobs and Swink 2011; ElMaraghy et al. 2012) by providing empirical evidence (albeit limited to one case) identifying the product and process correlations that contribute to the most significant complexity cost factors. The use of these complexity cost factors enriches the literature in that it delimits the data and resources needed to calculate these costs.

Determining initiatives based on the identified complexity cost factors contributes to the literature on strategies for reducing complexity (Closs et al. 2008; Jacobs and Swink 2011), in that these factors help focus on the most significant complexity cost drivers. Thus, the paper provides a basis for identifying initiatives that have significant potential to reduce complexity in the company. Furthermore, involving experienced managers and employees in determining the project scope, as well as in the analysis and synthesis, makes it possible to use the company's internal knowledge and experience in the analysis and identification of the most relevant initiatives, thus contributing to defining strategies for reducing complexity in individual companies (Closs et al. 2008; Jacobs and Swink 2011).

7. Conclusion

Based on a literature review, this paper proposed a five-step procedure that provides a structured method of identifying and quantifying the most significant complexity cost factors to then enable the identification and quantification of possible initiatives to reduce these complexity costs. The five steps are as follows: (1) define the scope of the products and processes to be included in the analysis, (2) conduct ABCC analysis of the products, (3) identify and quantify the most significant complexity cost factors, (4) identify and quantify possible initiatives for the reduction of complexity costs, and (5) evaluate and prioritise initiatives to establish a complexity cost reduction programme. To test the usefulness and efficiency of the procedure, a case study was carried out.

The case study showed that the procedure is applicable to projects of a relatively short duration, and that it demands relative limited resources to apply. The study also showed that the use of the procedure in a project made it possible to obtain the necessary data, identify and quantify significant complexity cost factors, and identify and quantify initiatives to reduce the complexity. Furthermore, the analysis provided the case company with new and more detailed insight into the costs of complexity for each variant in the product portfolio.

Because the study pertains to a single case, it is possible to make only analytical generalisations concerning justification based on similarities to other cases. In this context, it should be considered that the challenges in the investigated company, as well as the ways in which the products and processes were managed, are largely typical in manufacturing companies. Thus, it seems that the procedure would also be applicable to many other companies. More specifically, the suggested procedure should be applicable to other companies that manufacture consumer products, where the number of finished items could be considered for reduction, the costs of having numerous variants in the product portfolio could be investigated, and possibilities for reducing complexity costs in the business processes could be examined. However, detailed cost registration is required for the application of the procedure. If the cost information is too limited, the procedure could encounter too many difficulties, thereby preventing its application. In such cases, a more qualitative approach is advisable. Finally, it should be mentioned that subsequent applications of the proposed procedure have been made by the case company. The experiences from these subsequent projects have demonstrated that while company employees are capable of carrying out the data analysis and calculations, they need support in regard to checking and validating the data and findings, identifying potential complexity cost factors and initiatives, and interpreting the results. In other words, it has been shown that the procedure can be applied without facilitation by consultants and researchers – that is, support for the validity of the findings.

The proposed procedure contributes to the literature on product complexity (Yang et al. 2008; ElMaraghy et al. 2013) by suggesting an operational method of grouping products into A, B, and C categories and characterising them using features that are identified by the company's R&D employees. Furthermore, the adjusted contribution margin for each product variant contributes to the literature on product architecture (Meyer and Lehnerd 1997) by suggesting the use of adjusted contribution margins based on the quantified complexity cost factors. The proposed procedure also supplements the literature on process complexity (Sivadasan et al. 2002; ElMaraghy et al. 2012) by suggesting complexity cost factors as a means of analysing the most significant correlated complexity between product variants and processes and by using experienced practitioners in the company to identify the most significant complexity cost factors. The proposed procedure contributes to the literature on the quantification of complexity costs (Zhang and Tseng 2007; Orfi, Terpenny, and Sahin-Sariisik 2011) by describing the most significant complexity cost factors based on an analysis of the cost distribution of the products identified and providing a list of possible complexity cost factors. The use of these insights reduces the data and calculations needed to quantify the complexity costs and to calculate the expected impact of the initiatives identified to reduce complexity. Finally, this procedure contributes to the theory on strategies for reducing complexity (Wilson and Perumal 2009; Jacobs and Swink 2011) by making use of the company employees' knowledge and experience to identify the most relevant initiatives for reducing complexity and by using the complexity cost factors to quantify the expected benefits from each initiative.

Scheiter, Scheel, and Klink (2007) claimed that reducing complexity could improve EBIT by 3–5%; however, according to Wilson and Perumal (2009), Mariotti (2008), and Jacobs and Swink (2011), many companies are unaware of this potential and lack operational and easy-to-use procedures for analysing and reducing complexity. The proposed procedure assists with the analysis and quantification of the most significant complexity cost factors in companies with limited resources and data. This is important to enable awareness and insight into the potential gains from reducing complexity in products and operations. Furthermore, the proposed procedure suggests how to develop initiatives to reduce complexity based on the analysis. Ultimately, as shown by the first applications, the use of the procedure may lead to a significant reduction in complexity costs and improved EBIT for the company.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

Anderson, S. R., and R. S. Kaplan. 2007. *Time-driven Activity-based Costing: A Simpler and More Powerful Path to Higher Profits*. Boston, MA: Harvard Business School Press Books.

Arashpour, M., R. Wakefield, N. Blismas, and B. Abbasi. 2015. "Quantitative Analysis of Rate-driven and Due Date-driven Construction: Production Efficiency, Supervision, and Controllability in Residential Projects." *Journal of Construction Engineering and Management* 142 (1): 1–11.

Blecker, T., N. Abdelkafi, B. Kaluza, and G. Kreutler. 2004. "Mass Customization vs. Complexity: A Gordian Knot?" In *Proceedings of 2nd International Conference "An Enterprise Odyssey: Building Competitive Advantage"*, edited by L. Galetic, 890–903. Zagreb, Croatia: University of Zagreb, 17–19.06. 2004.

Bozarth, C., D. Warsing, B. Flynn, and E. J. Flynn. 2009. "The Impact of Supply Chain Complexity on Manufacturing Plant Performance." *Journal of Operations Management* 27 (1): 78–93.

Chiu, M.-C., C.-Y. Chu, and C.-C. Chen. 2018. "An Integrated Product Service System Modelling Methodology with a Case Study of Clothing Industry." *International Journal of Production Research* 56 (6): 2388–2409.

Choi, T. Y., and D. R. Krause. 2006. "The Supply Base and Its Complexity: Implications for Transaction Costs, Risks, Responsiveness, and Innovation." *Journal of Operations Management* 24 (5): 637–652.

Closs, J., M. A. Jacobs, M. Swink, and G. S. Webb. 2008. "Toward a Theory of Competencies for the Management of Product Complexity: Six Case Studies." *Journal of Operations Management* 26 (5): 590–610.

Cooper, R., and R. S. Kaplan. 1988. "Measure Costs Right: Make the Right Decisions." *Harvard Business Review* 66 (5): 96–103.

ElMaraghy, W., H. ElMaraghy, T. Tomiyama, and L. Monostori. 2012. "Complexity in Engineering Design and Manufacturing." *CIRP Annals: Manufacturing Technology* 61 (2): 793–814.

ElMaraghy, H., G. Schuh, W. ElMaraghy, F. Piller, P. Schönsleben, M. Tseng, and A. Bernard. 2013. "Product Variety Management." *CIRP Annals: Manufacturing Technology* 62 (2): 629–652.

ElMaraghy, W. H., and R. J. Urbanic. 2003. "Modelling of Manufacturing Systems Complexity." *CIRP Annals: Manufacturing Technology* 52 (1): 363–366.

Ericsson, A., and G. Erixon. 1999. *Controlling Design Variants: Modular Product Platforms*. Dearborn, MI: Society of Manufacturing Engineers.

Forza, C., F. Salvador, and A. Trentin. 2008. “Form Postponement Effects on Operational Performance: A Typological Theory.” *International Journal of Operations & Production Management* 28 (11): 1067–1094.

George, M. L., and S. A. Wilson. 2004. *Conquering Complexity in your Business*. New York: McGraw-Hill.

Hansen, C. L., N. H. Mortensen, L. Hvam, and U. Harlou. 2012. “Calculation of Complexity Costs: An Approach for Rationalizing a Product Program.” In *DS 71: Proceedings of NordDesign 2012*, the 9th NordDesign Conference. Denmark: Aalborg University, 22–24.08. 2012.

Haug, A. 2010. “A Software System to Support the Development and Maintenance of Complex Product Configurators.” *The International Journal of Advanced Manufacturing Technology* 49 (1–4): 393–406.

Haug, A. 2013. “Improving the Design Phase Through Interorganisational Product Knowledge Models.” *International Journal of Production Research* 51 (2): 626–639.

Haug, A., L. Hvam, and N. H. Mortensen. 2013. “Reducing Variety in Product Solution Spaces of Engineer-to-Order Companies: The Case of Novenco A/S.” *International Journal of Product Development* 18 (6): 531–547.

Hvam, L. 2001. “A Procedure for the Application of Product Modelling.” *International Journal of Production Research* 39 (5): 873–885.

Hvam, L., K. Kristjansdottir, S. Shafiee, N. H. Mortensen, and Z. N. L. Herbert-Hansen. 2018. “The Impact of Applying Product-Modelling Techniques in Configurator Projects.” *International Journal of Production Research*. doi:10.1080/00207543.2018.1436783.

Hvam, L., N. H. Mortensen, and J. Riis. 2008. *Product Customization*. Berlin: Springer.

Jacobs, M. A. 2013. “Complexity: Toward an Empirical Measure.” *Technovation* 33 (4–5): 111–118.

Jacobs, M. A., and M. Swink. 2011. “Product Portfolio Architectural Complexity and Operational Performance: Incorporating the Roles of Learning and Fixed Assets.” *Journal of Operations Management* 29: 677–691.

Koch, R. 2008. *The 80/20 Principle: The Secret to Achieving More with Less*. New York: Doubleday.

Kumar, L. 2018. “Knowledge-based Expert System in Manufacturing Planning: State-of-the-Art Review.” *International Journal of Production Research*. doi:10.1080/00207543.2018.1424372.

Lakemond, N., T. Magnusson, G. Johansson, and K. Säfsten. 2013. “Assessing Interface Challenges in Product Development Projects.” *Research-Technology Management* 56 (1): 40–48.

Lechner, A., K. Klingebiel, and A. Wagenitz. 2011. “Evaluation of Product Variant-Driven Complexity Costs and Performance Impacts in the Automotive Logistics with Variety-Driven Activity-Based Costing.” *Proceedings of the International MultiConference of Engineers and Computer Scientists* 2: 1088–1096. Hong Kong, 16–18.03. 2011.

Lindemann, U., M. Maurer, and T. Braun. 2010. *Structural Complexity Management*. Berlin: Springer.

Lu, R. F., T. D. Petersen, and R. L. Storch. 2007. “Modeling Customized Product Configuration in Large Assembly Manufacturing with Supply-Chain Considerations.” *International Journal of Flexible Manufacturing Systems* 19 (4): 685–712.

MacDuffie, J. P. 2013. “Modularity-as-Property, Modularization-as-Process, and ‘Modularity’-as-Frame: Lessons from Product Architecture Initiatives in the Global Automotive Industry.” *Global Strategy Journal* 3 (1): 8–40.

Mahler, D., and A. Bahulkar. 2009. “Smart Complexity.” *Strategy & Leadership* 37 (5): 5–11.

Mariotti, J. L. 2008. *The Complexity Crisis: Why Too Many Products, Markets and Customers Are Crippling Your Company – And What to Do About It*. Avon, MA: Platinum Press.

Meyer, H. M., and A. P. Lehnerd. 1997. *The Power of Product Platforms*. New York: Free Press.

Mortensen, N. H., L. Hvam, A. Haug, P. Boelskifte, C. Lindschou, and S. Frobenius. 2010. “Making Product Customization Profitable.” *International Journal of Industrial Engineering* 17 (1): 25–35.

Muketha, G. M., A. A. A. Ghani, M. H. Selamat, and R. Atan. 2010. “A Survey of Business Process Complexity Metrics.” *Information Technology Journal* 9 (7): 1336–1344.

Myrodia, A., and L. Hvam. 2015. “Identification of Complexity Cost Factors in Manufacturing Companies.” In *Proceedings of the 22nd EurOMA Conference*. Neuchâtel, Switzerland, 28.06-01.07 2015.

Olhager, J. 2003. “Strategic Positioning of the Order Penetration Point.” *International Journal of Production Economics* 85 (3): 319–329.

Olivares Aguila, J., and W. ElMaraghy. 2018. “Structural Complexity and Robustness of Supply Chain Networks Based on Product Architecture.” *International Journal of Production Research* 56 (20): 6701–6718.

Orfi, N., J. Terpenny, and A. Sahin-Sariisik. 2011. “Harnessing Product Complexity: Step 1 –Establishing Product Complexity Dimensions and Indicators.” *The Engineering Economist* 56 (1): 59–79.

Park, J., and T. W. Simpson. 2008. “Toward an Activity-based Costing System for Product Families and Product Platforms in the Early Stages of Product Development.” *International Journal of Production Research* 46 (1): 99–130.

Perona, M., and G. Miragliotta. 2004. “Complexity Management and Supply Chain Performance Assessment: A Field Study and a Conceptual Framework.” *International Journal of Production Economics* 90: 103–115.

Rogozhin, A., M. Gallaher, G. Helfand, and W. McManus. 2010. “Using Indirect Cost Multipliers to Estimate the Total Cost of Adding New Technology in the Automobile Industry.” *International Journal of Production Economics* 124 (2010): 360–368.

Samy, S. N., and H. A. ElMaraghy. 2012. “Complexity Mapping of the Product and Assembly System.” *Assembly Automation* 32 (2): 135–151.

Sandrin, E. 2016. “An Empirical Study of the External Environmental Factors Influencing the Degree of Product Customization.” *International Journal of Industrial Engineering and Management* 7 (4): 135–142.

Sandrin, E., A. Trentin, and C. Forza. 2014. “Organizing for Mass Customization: Literature Review and Research Agenda.” *International Journal of Industrial Engineering and Management* 5 (4): 159–167.

Schaffer, J., and H. Schleich. 2008. “Complexity Cost Management.” In *Built to Order: The Road to the 5-Day Car*, edited by Glenn Parry, and Andrew Graves, 155–174. London: Springer.

Scheiter, S., O. Scheel, and G. Klink. 2007. *How Much Does Complexity Really Cost?* Düsseldorf: A. T. Kearney.

Shah, R., and P. T. Ward. 2007. “Defining and Developing Measures of Lean Production.” *Journal of Operations Management* 25 (4): 785–805.

Silverman, D. 2005. *Doing Qualitative Research*. London: Sage.

Sivadasan, S., J. Efstathiou, G. Frizelle, R. Shirazi, and A. Calinescu. 2002. “An Information–Theoretic Methodology for Measuring the Operational Complexity of Supplier–Customer Systems.” *International Journal of Operations and Production Management* 22: 80–102.

Suzić, N., C. Forza, A. Trentin, and Z. Anišić. 2018. “Implementation Guidelines for Mass Customization: Current Characteristics and Suggestions for Improvement.” *Production Planning & Control* 29 (10): 856–871.

Suzić, N., E. Sandrin, S. Suzić, C. Forza, A. Trentin, and Z. Anišić. 2018. “Implementation Guidelines for Mass Customization: A Researcher-Oriented View.” *International Journal of Industrial Engineering and Management* 9 (4): 229–243.

Thumm, B. R., and D. Goehlich. 2015. “Product Architecture Design Methodology for Developing Standardized Modules.” In *DS 80-7 Proceedings of the 20th International Conference on Engineering Design (ICED 15) Vol 7: Product Modularisation, Product Architecture, Systems Engineering, Product Service Systems.* Milan, Italy, 27–30.07. 15.

Trentin, A., F. Salvador, C. Forza, and M. J. Rungtusanatham. 2011. “Operationalising Form Postponement from a Decision-making Perspective.” *International Journal of Production Research* 49 (7): 1977–1999.

Ulrich, K. T., and S. D. Eppinger. 2008. *Product Design and Development*. New York: McGraw Hill.

Wan, X., P. T. Evers, and M. E. Dresner. 2012. “Too Much of a Good Thing: The Impact of Product Variety on Operations and Sales Performance.” *Journal of Operations Management* 30 (4): 316–324.

Wang, H., X. Zhu, H. Wang, S. Jack Hu, Z. Lin, and G. Chen. 2011. “Multi-objective Optimization of Product Variety and Manufacturing Complexity in Mixed-Model Assembly Systems.” *Journal of Manufacturing Systems* 30 (1): 16–27.

Wildemann, H. 2001. *Das Just-in-Time Konzept: Produktion und Zulieferung auf Abruf*. München: TCW Unternehmensberatung.

Wilson, S. A., and A. Perumal. 2009. *Waging War on Complexity Costs*. New York: McGraw-Hill.

Wu, Y., G. Frizelle, and J. Efstathiou. 2007. “A Study on the Cost of Operational Complexity in Customer–Supplier Systems.” *International Journal of Production Economics* 106: 217–229.

Xie, S. Q., W. Z. Yang, and Y. L. Tu. 2008. “Towards a Generic Product Modelling Framework.” *International Journal of Production Research* 46 (8): 2229–2254.

Yang, W. Z., S. Q. Xie, Q. S. Ai, and Z. D. Zhou. 2008. “Recent Development on Product Modelling: A Review.” *International Journal of Production Research* 46 (21): 6055–6085.

Yin, R. K. 2009. *Case Study Research: Design and Methods*. Los Angeles: Sage.

Zhang, L. L. 2014. “Product Configuration: A Review of the State-of-the-Art and Future Research.” *International Journal of Production Research* 52 (21): 6381–6398.

Zhang, X., and V. Thomson. 2018. “A Knowledge-based Measure of Product Complexity.” *Computers & Industrial Engineering* 115: 80–87.

Zhang, M., and M. M. Tseng. 2007. “A Product and Process Modeling Based Approach to Study Cost Implications of Product Variety in Mass Customization.” *IEEE Transactions on Engineering Management* 54 (1): 130–144.

Zheng, H., W. Liu, and C. Xiao. 2018. “Structural Relationship Model for Design Defect and Influencing Factors in the Concurrent Design Process.” *International Journal of Production Research* 56 (14): 4897–4924.

6.9 Paper I

Standard Article

Concurrent Engineering: Research and Applications
2016, Vol. 24(2) 153–163

DOI: 10.1177/1063293X15624133
cer.sagepub.com
SAGE

Assessing the cost saving potential of shared product architectures

Niels Henrik Mortensen[1], Christian Lindschou Hansen[1], Martin Løkkegaard[1] and Lars Hvam[2]

Abstract

This article presents a method for calculating cost savings of shared architectures in industrial companies called Architecture Mapping and Evaluation. The main contribution is an operational method to evaluate the cost potential and evaluate the number of product architectures in an industrial company. Experiences from the case company show it is possible to reduce the number of architectures with 60% which leads to significant reduction in direct material and labor costs. This can be achieved without compromising the market offerings of products. Experiences from the case study indicate cost reductions between 0.5% and 2% of turnover. The main implication is that the method provides a quantitative basis for the discussion on whether or not to implement shared product architectures. This means a more fact-based approach is introduced.

Keywords

product architecture, manufacturing architecture, modularization, concurrent engineering

Introduction

Many industrial companies (developing, producing, and selling physical products) have developed the product portfolio sequentially product by product over a number of years. There can be many good reasons to continue doing so, for example, ability to develop specific products for specific markets and targeting specific low-cost needs and high-end needs. The consequences are, however, often that there exist a large portfolio of products, where there is very limited sharing between the product families, leading to increased complexity cost, several "inventing the wheel" projects and thereby increased time to market and profit for new products (Andreasen, 1980; Hansen, 2015; Harlou, 2006; Levandowsky et al., 2014). A warning signal is often that costs are increasing faster than turnover.

Companies typically have challenges such as the need to reduce cost, increase quality, reduce delivery time, and launch more new innovative products faster. One of the means to address this challenge that is often discussed in both academia and industry is application of modular architectures (Ericsson and Erixon, 1999; Guðlaugsson et al., 2014; Herrmann et al., 2004). The basic idea of modular product architectures is to build up product lines based on a limited well-defined module having well-defined performance steps with clear definition of interfaces (Ericsson and Erixon, 1999). This should lead to the reduction in the number of components, cost reduction in general, and more focused effort on key modules leading to more cost-effective products. The rationalization benefits may be utilized to develop more new innovative products.

In principle, everyone, from board of directors, board of management, and down in an industrial organization, agree on this. But, in practice, there are many uncertainties and many opinions. In our research, we have often come across viewpoints such as "we have exactly the products that we need," "all products are profitable and needed for our customers and markets," "we are already modular," and "if we increase the level of modularity, we will compromise key customer

[1]Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby, Denmark
[2]Department of Management Engineering, Technical University of Denmark, Kongens Lyngby, Denmark

Corresponding author:
Niels Henrik Mortensen, Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark, Building 426, DK-2800 Kongens Lyngby, Denmark.
Email: nhmo@mek.dtu.dk

requirements and increase cost." All the above viewpoints can often not be proved right or wrong.

Organizational wise, this is a very sensitive topic. Asking the question "could we do better with modular architectures." From research and development (R&D), the reaction is often that they are accused of not having done the perfect job. From sales, the viewpoint will often be that due to the competitive situation and so on, all products are required. In manufacturing, the reaction is very positive—but often more short-term initiatives are in focus such as day-to-day process improvements.

The basic question asked in this article is as follows: how to find out what the financial potential is of shared modular architectures? The target audience is board of management. The intention has been to identify a quantitative method in such a way that discussions are based on facts and not opinions of individuals.

Concerning the link between the number of architectures and concurrent engineering (CE), it is the assumption that when the number of architectures in product and manufacturing is reduced, it will lead to increased efficiency and increased possibilities of achieving concurrent development of product and manufacturing.

The structure of the article is as follows: in section "Research approach," the research method is explained, and section "What is a product architecture and which evaluation parameters to include?" will go through the benefit dimensions and the reasons for including them. Section "State of the art" describes state-of-the-art literature. In section "AME method," the method for calculation of the benefits of shared architectures is presented and section "Application of the AME method" contains experience from application of the method in a large-scale organization. Section "Discussion" contains the conclusion.

Research approach

The suggested method for assessing the potential of shared product architectures has been developed by taking from both the existing literature and some experiences of practitioners. More specifically, the method is based on classical systems' thinking (Andreasen, 1980; Skyttner, 2005), theory of technical systems (Hubka, 1973), and Product Family Master Plan (PFMP) (Harlou, 2006). The work is carried out by three MSc projects and two PhD projects (Bruun, 2015; Hansen, 2015) at the Technical University of Denmark, Department of Mechanical Engineering, Section of Engineering Design and Product Development. The basic assumption is that the competitiveness of a product program cannot be evaluated in itself—only when mapped relative to the market and the internal functions of the company, conclusions can be made. In this study, three aspects are considered: market, product, and manufacturing. The links between these three aspects are the foundations for evaluating the potential of shared architectures. Most approaches in the literature on shared product architectures are concerned with the product aspects in terms of, for example, shared parts. Even though this can be of relevance, it is not sufficient.

A method named Architecture Mapping and Evaluation (AME) method is proposed. The method has been tested in a global company that has approximately 75,000 commercial variants in the market. The company has divided the product portfolio into six product lines. The AME has been tested on all six product lines. This means that in total, six global data sets have been collected and evaluations have been carried out. The main aim of this case study is to test the suggested operational method and receive feedback from the managers in the company.

With regard to internal validity, the research team has full access to detailed data from the company. In order to gather accurate qualitative data, un- and semi-structured interviews are performed with the "key" informants. The research group had semi-structured interviews with the managers, involved in this project, in order to assess the results and receive feedback. The received feedback is valuable for the verification of the results from the analysis and for assessing the proposed method. The studies have been carried out over a time span of 2 years from 2012 to 2014. The next section will discuss the meaning of product architecture and what evaluation parameters to include.

What is a product architecture and which evaluation parameters to include?

Like most phenomena in engineering design, there does not exist a common and agreed way of defining architecture. In this article, a distinction between product structure and product architecture is made (Hansen et al., 2012). Product structure means the way a single product is built up from systems and components. Product architecture means the way a product family or portfolio of products is built up. Traditionally, companies have good control of product structure in, for example, computer-aided design (CAD), enterprise resource planning (ERP), and product data management (PDM) systems. Product architecture is normally very weakly taken care of. Traditionally, responsibility for product structure is well defined, but responsibility of product architecture is ill defined. It is the main assumption in this article that the number of product architectures is a very important fundamental aspect, and that top management and other key persons should consider very carefully. Having too many architectures will lead to high complexity cost and long time to market for product development. The implication of too few architectures can be too high cost for product in, for example, the lower performance areas of the portfolio or simply that the company cannot serve the variety of needs among customers.

In this work, product architecture has the following characteristics:

- Shared core interfaces.
- Core modules/systems exist in balanced performance steps.
- The architecture is explicitly prepared for derivative products and related properties in terms of cost and performances are known.

The above phenomena will briefly be explained.

Shared core interfaces

Only a small fraction of interfaces play an important role, but a few are extremely important for, for example, quality and time to market. An example of a core interface of a truck might be the interface between the cab and the rest of the chassis. If this interface is stable, the cab can be developed without changing the rest of the chassis. The whole product family can be upgraded in one step with one development project.

Core modules exist in well-balanced performance steps

An example of a core module could be the wash-group of a washing machine; some of the performance steps could be 6, 8, 10, and 12 kg. Balanced means that there the number of modules is consciously determined according to market needs and internal complexity within the company, for example, production, service, stock level and development capability. One "ideal" way of balanced performance thinking is "one need—one solution."

Architecture is prepared for future launches

An example could be boggies of a truck. There might exist 21 and 30 ton, but modules are prepared for a 26-ton variant with adding only a few new parts. Another consequence of this is that interfaces have to be stable over time. This is one of the weak parts of architecture work in most companies that we have studied (Bruun et al., 2014; Hvam et al., 2008).

In this article, a product architecture is considered shared when more than 90% of the core interfaces are shared. Then one can ask what a core interface is. This is pragmatic defined among senior market, product, and manufacturing persons. For a car, an example of a core interface is between engine and transmission. For a drilling tool, a core interface will exist between battery and chassis. The basic assumption is that the number of architectures is driving complexity cost; it is driving CAPital Expenditures (CAPEX) in manufacturing and it is often constraining the ability to launch new products and product variants. Then why put the requirement on 90% sharing of core interfaces? This is a pragmatic decision, but due to the size of the test company, the criteria had to be explicitly defined in such a way that each division of the case company could not have individual perceptions.

Having clarified the meaning of architecture, the next question is what evaluation parameters to include. There is in principle an infinite number of evaluation parameters that could be studied. In the literature, roughly two types of evaluations parameters are reported in the literature (Fixson, 2005; Hultink et al., 1997; Krause et al., 2013; Ulrich, 1995) from application of shared architectures. They can be divided into growth parameters and rationalization parameters. Examples of growth parameters could be time to market and ability to make new innovative products. Examples of rationalization parameters could be direct material cost and labor cost. In this article, it has been decided not to include growth parameters, not because it is irrelevant, but because it is difficult to obtain quantifiable data. There are often many opinions but very few facts. On the rationalization side, it has been decided to include data that are available in modern companies with modern information and technology (IT) systems, mainly ERP systems. Again, there are many possibilities, but included are four parameters: direct material cost, direct labor cost in manufacturing and CAPEX on tooling, and number of architectures. The basic assumption is that if benefits can be justified in these dimensions, the rest such as the growth parameters will be additional benefits.

State of the art

The review of the state-of-the-art includes a review of five different groups of supporting methods for the identification of shared architecture benefits for a product program including product lines. The five groups identified are function-based models, matrix-based models, CE, design for manufacture (DFM), and mathematical models.

Function-based models

Methods describing the development of modular product architectures often choose to start with the conscious mapping of functional structures into physical modules (Levandowsky et al., 2014). Functions can be represented in function-based models, for example, functions-and-means trees (Andreasen, 1980), or by schematics of the product including physical elements to a meaningful extent (Stone et al., 2000).

The understanding product functions can be used in different ways to identify possible modules. To improve the identification of modules and make sure that the modular architecture will serve its objectives (Fixson, 2005), define a set of module drivers. The module drivers can support the reasoning behind the module identification by elaborating the justification of the modules' existence, for example, "planned product changes" module, "process" module, "different specification" module, and "technology evolution" module. The module drivers are a part of a comprehensive framework called modular function deployment (MFD), which in analog to the quality function deployment (QFD) method provides support for the linking of relationship between the module drivers and technical solutions.

Matrix-based models

Another approach to identify modules is the application of design structure matrices (DSMs). This approach takes its point of departure in the decomposition of a product into parts and/or subsystems while identifying the relations (and possible future interfaces) among these (Gonzalez-Zugasti et al., 2000; Otto and Wood, 1998). By applying different algorithms and clustering techniques, it is possible to encapsulate functional "chunks" that have the potential of becoming physical modules, due to their functional interrelations. DSM techniques are the subject of many research initiatives and serve as the basis for an array of derived methodologies. An example of this is the multi-domain-matrix (Ulrich, 1995). Alternatively, other design tools focus more on the specific task of examining different functional flows with the aim of identifying modules (Otto and Wood, 1998; Pimmler and Eppinger, 1994). These methods are heuristically based.

Other more general methods focus on the identification of common features in the existing product program in order to point out the basis of the product architecture. By formulating the design task as a quantitative problem, which can be subject to optimization, this method is balancing inputs from requirements and product variants design with data models of performance and costs. By iteration, the optimal product variants are designed and evaluated through quantitative performance metrics.

CE

From the associated area of CE, one can also find research into the concurrent development of product and production architectures, with phrasings such as "methods supporting the development of product platforms." Nevertheless, interesting contributions are submitted within this area. Otto and Wood (1998) introduced a three-dimensional (3D) methodology superimposing the traditional domains of CE, by suggesting the linking of technology, architecture, and focus relations in the process, product, and supply chain domains. Olesen (1992) proposed an important step of operationalization of this 3D-CE approach by developing a multi-dimensional framework that enables comprehensive assessment of alternative product architectures.

The concept of architecture for product family (APF) is introduced as a conceptual structure, proposing logics for the generation of product families (Hultink et al., 1997). The generic product structure (GPS) is then proposed as the platform for tailoring products to individual customer needs. In Andreasen and Olesen (1990), another systematic method for concurrent development of product families is presented, by combining QFD-based methods with quantified DSM techniques and morphology analysis to visualize concepts.

DFM

Original contributions from Olesen (1992) proposed a framework for the concurrent development of manufacturing supported by the theory of dispositions (Andreasen and Olesen, 1990). This is done by proposing a set of models aligning the product design and the product life system phase of manufacturing to create a fit. However, for the case with Design for Assembly (DFA) and DFM methodologies, the main focus is single product development. Herrmann et al. (2004) comment that an extension of the DFM tools to comprise multi-product development will hold the key to achievement of competitiveness.

Mathematical models

Some researchers have undergone the task of developing methods based on mathematical models. Some methods are based on measures of modularity, which act as subjects of optimization using different techniques (Hultink et al., 1997). Others seek to integrate product platform, manufacturing process, and supply chain decisions through the application of mathematical models, thus extending the concept of the generic bills of materials (GBOM) by quantifying relations between decisions from the different domains.

Conclusion

It is evident that the contributions mentioned above can play a role in the identification of program architectures. Situated in this cross-functional research field, it is clear how research centered within either the product or production domain tends to leave out important aspects of the adjacent fields, and considering the identification of program architecture this is a deficiency considering the contributions listed above. Extensive research is also found within the reengineering of business processes and different means of optimization of operations, but these areas exclude necessary details within the field of architectures. They are simply not concrete enough, or deal with sub-optimization of operations and processes leaving out the product domain. The methods do not explain how the modeling and evaluation is carried out for very large product programs with, for example, 70,000 commercial products and 300,000 parts. There is very little support for supporting the very fundamental question: "how many product architectures are right for our company?"

AME method

This section presents a seven-step method to evaluate the benefits of shared modular architectures. These seven steps are as follows:

Step 1: map the market globally and main required properties;
Step 2: map cost/performance for core module areas;
Step 3: map each as-is product architectures;
Step 4: map each as-is manufacturing architectures;
Step 5: identify to-be product architectures and manufacturing architectures;
Step 6: map cycle plan;
Step 7: calculate financial impact.

In the following, each step will be explained.

Step 1: map the market globally and main required properties

In this step, the market and required properties are mapped according to Fixson (2005), Levandowsky et al. (2014), and Meyer and Lehnerd (1997). It means that the market for a product line is grouped into approximately 4–12 categories. There are normally two axes in the mapping (segment and performance levels such as high end and basic), see Figure 1.

For a pump manufacturer, it might segment wise be geographical area (e.g. North America, Europe, Asia pacific) and performance wise, media pressure (up to 2 bar and above). For each group, key properties, for example, energy efficiency and lifting height, are identified. The product line properties are then mapped and competitor product (best in class) is mapped. The result is a number of "spider charts" as shown in Figure 1. Finally trend indicators are identified. It means in

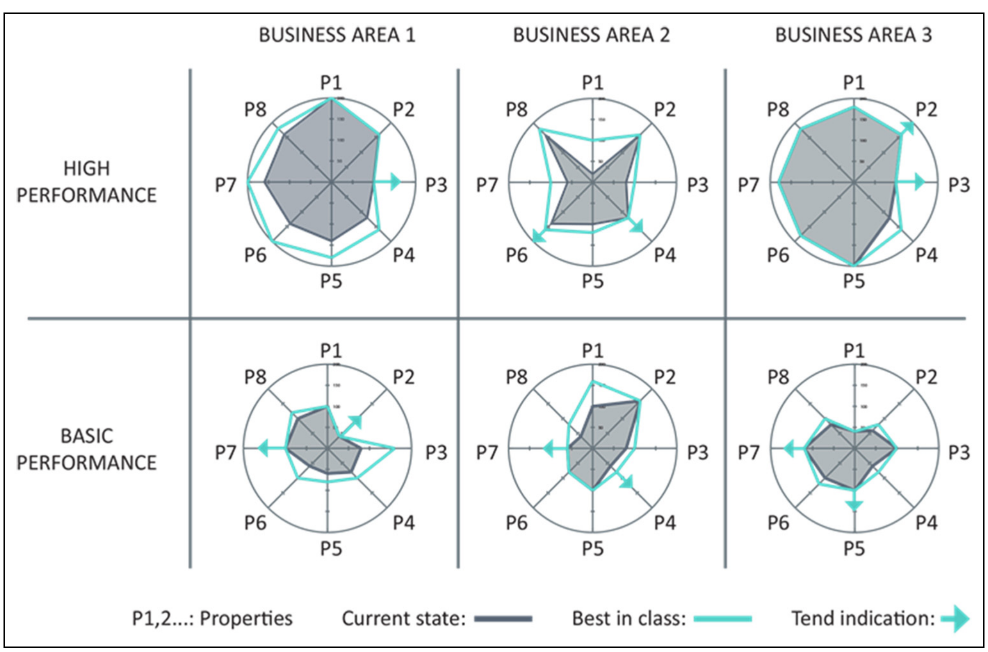


Figure 1. Mapping of market segments and required properties.

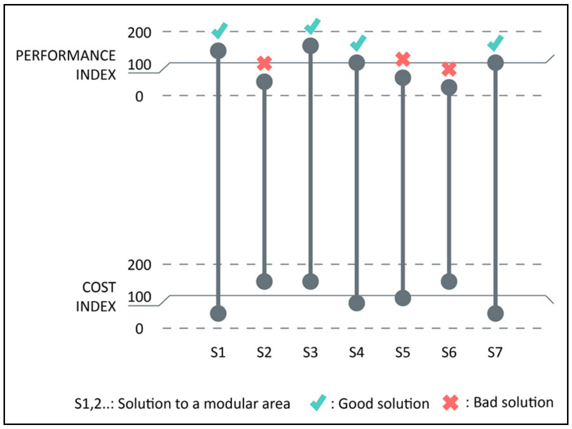


Figure 2. Description of cost/performance for a key module is mapped. Each vertical line represents a module area. The lower dot is the cost and the upper dot is a certain performance of a property.

which directions do the company expect that requirements will change. Concerning the energy efficiency, it is very likely to be reduced in next generation of pumps. Trend indicators are utilized in Step 5. It is very important that the architectures are prepared to deliver the right properties. One CEO explained, "It is important to be prepared for the next war and not the previous one."

Step 2: map cost/performance for core module areas

In this step, the products in a product line are pragmatically divided into a number of module areas that are the carrier of key properties for a product. For a pump, it might be motor, hydraulics, controls, and so on. Then a few key module areas are identified, which are the carrier of major cost and major properties (Huang et al., 2005; Otto and Wood, 1998). The purpose of this step is to map key module areas in a direct material cost and a relevant performance dimension. For a pump manufacturer, it could be the motor and the controls. Often, a few module areas cover the majority of the cost and performance in a product. Then each module area variant is mapped in a cost performance diagram as shown in Figure 2.

This overview is quite important in the method (Guo and Gershenson, 2007). Often, there will be different module areas with very different cost levels but is delivering same performance. In other cases, there will be module areas that have low performance and high cost. In other words, the module area is expensive and can do very little. What should be the immediate reaction to such a module area "is there any good reason for having this module area in the product line." What has been observed in the case project is that there is up to a factor 3 in direct cost differences between module areas that have similar performance. So there are significant direct material cost reduction possibilities by consequently utilizing the most cost-effective modules.

Step 3: map each as-is product architecture

In this step, the number of architectures within a certain product line is identified (Olesen, 1992). The process is that key interfaces are identified. This number has in this research project been 10 or below. Examples on key interfaces in a pump might be between housing and impeller. The interfaces play a crucial role in development for an industrial company. If and only if interfaces are shared, the modules can be shared. Figure 3 shows an example of how the number of architectures is identified. There is a very important link between Step 1 and Step 3. Reasoning from Step 1 to Step 3 should bring forward the question: how many architectures are right for our company in order to deliver good products in the different segments? In the case company, there has been a clear tendency that the companies have more product architectures than can be justified from a market point of view.

Step 4: map each as-is manufacturing architecture

The main purpose of this step is to identify differences in manufacturing properties, that is, labor cost (Andreasen and Olesen, 1990; Stone et al., 2000). Figure 4 shows an example where a product with different architectures is manufactured in different factories in Europe, United States, and China. What is compared are the differences in labor assembly time on subassembly lines and main assembly lines. In the case projects there has been a factor 2 deviation in labor time between the best and worst performing product architecture. This means that the product architecture plays a major role for efficiency in production. In principle, the productivity in the studied factories can be improved with a factor 2 by conscious selection of the best product architectures.

Step 5: identify to-be product architectures and manufacturing architectures

In this step, experienced persons from sales, product development, and manufacturing are taking a top–down look from a market point of view and identify how many architectures and module variants are needed in order to serve the market (Lindemann et al., 2009; Meyer and Lehnerd, 1997). This is really an expert judgment, where the most senior people in the organization have to be involved. In the case project, the reduction possibilities in terms of product

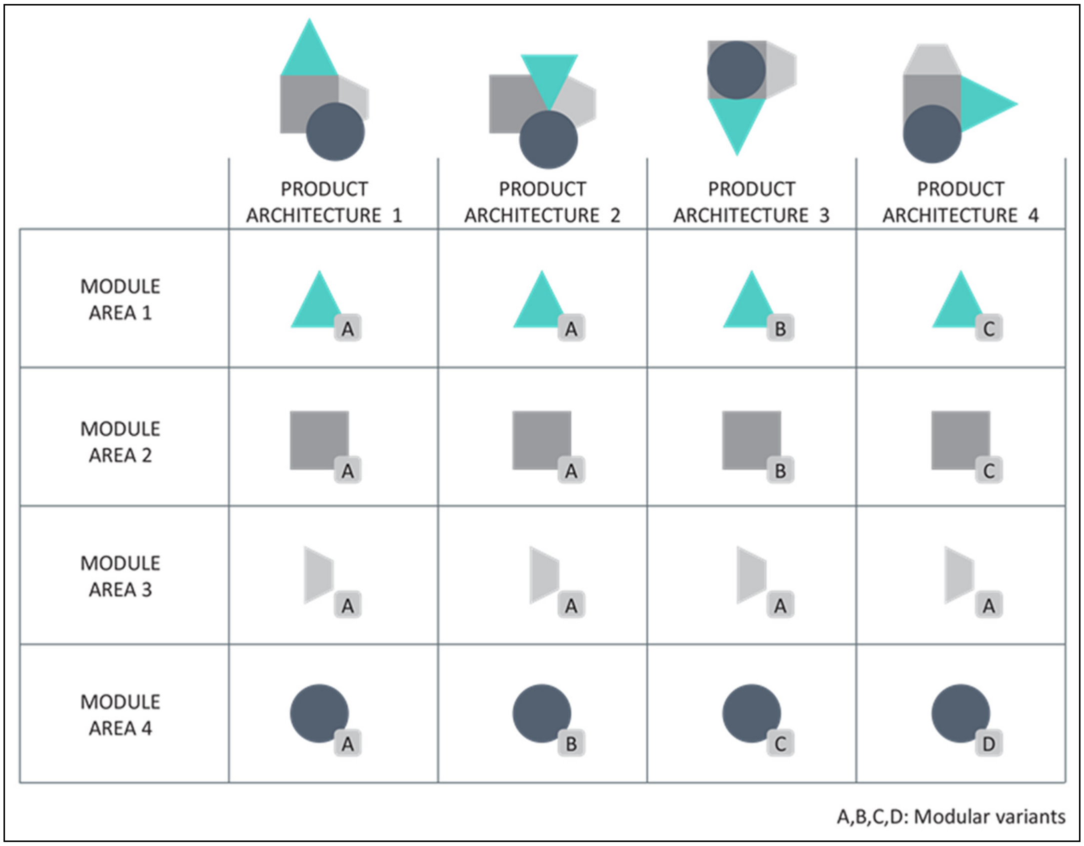


Figure 3. Mapping of current architectures.

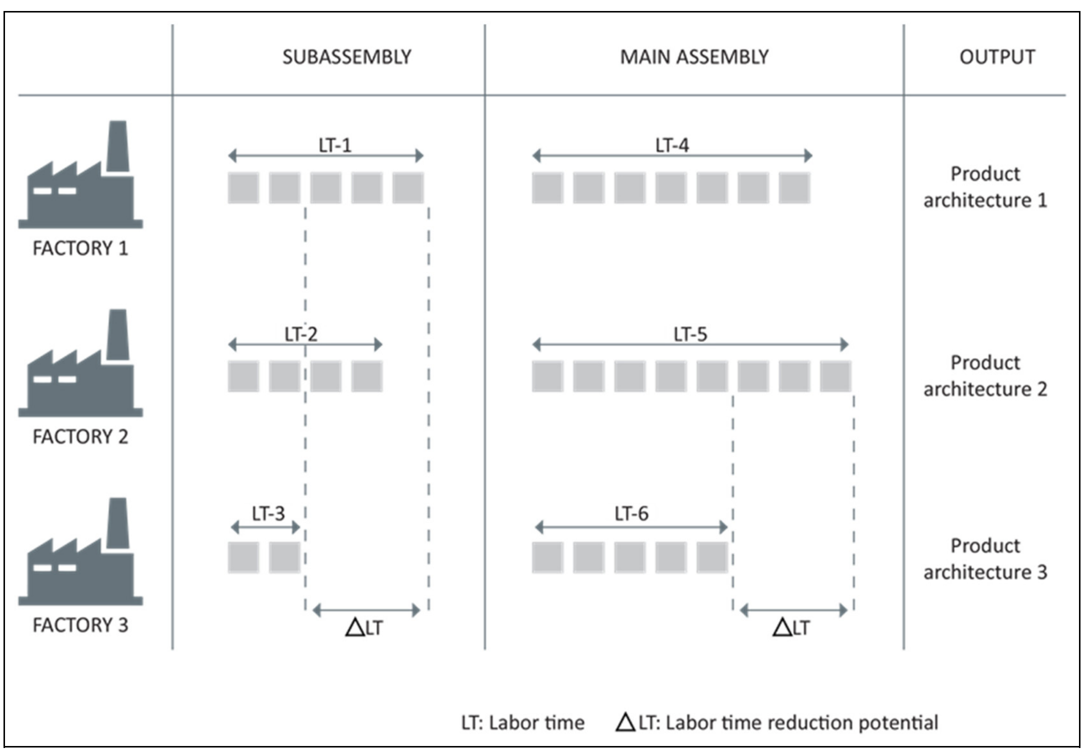


Figure 4. Manufacturing architectures.

architectures, manufacturing architectures, and module variants have been between 5% and 50%. In other words, the company is much more complex than needed.

Step 6: map cycle plan

The starting point in this step is a 5- or 10-year cycle plan, showing when products and product line are expected to be upgraded or relaunched (Krause et al., 2013; Pimmler and Eppinger, 1994). Next, phase out and phase in of architectures are added. It is further assumed that the best modules are consequently utilized across the product lines. Based on the reduction in product and manufacturing architectures, utilizing the most cost-effective module areas, it is possible to estimate direct material savings and direct labor savings.

Step 7: calculate financial impact

In this step, the benefits in terms of direct material cost, direct labor cost, and CAPEX avoidance are summed up (Du et al., 2001; Kester et al., 2013; MacDuffie, 2013). The results are three numbers explaining the financial potential of shared architectures. In the case company, this has been a very important step in order to put shared architectures on the top management agenda. One of the main advantages is that now such an initiative can be compared to other big initiatives such as automation, low-cost country sourcing, and manufacturing footprint location.

Application of the AME method

The method has been tested in a global business to consumer company. The case company has sales companies in 100 + countries and 35 factories in Europe, Asia, North America, and South America. There are six R&D centers that develop six product lines. The company has approximately 75,000 commercial product variants in the market and around 400,000 part numbers. The company has over a longer period been part of several mergers and acquisitions.

For several years, there had been a discussion in the board of management concerning the complexity of the product lines. It has among certain members been the assumption that it should be possible to serve the markets with fewer architectures and parts, but no definitive conclusions could be made. Therefore, the company wanted to test the AME method.

A team of three full-time persons (called the core team) for each product line was appointed. The core team consisted of a researcher, a senior R&D person, and a financial controller. This core team has ad hoc access to senior experts in sales/marketing, R&D, manufacturing, purchase, and financial control. In total, approximately 30 persons for each product line have been active in the work.

The AME work has been carried out during 20 weeks for each product line. The assessment work has been divided into three phases.

Phase 1

This included Steps 1, 2, 3, and 4. The main way of working has been interviews with key persons, site visits to key factories, and data extract from the ERP systems.

Phase 2

In Steps 5 and 6, three workshops with senior market, product, and manufacturing experts were carried out. The work was fundamentally anchored around the number of architectures. The main question asked was as follows: how many architectures do the company need in future? It is an illusion that there will be consensus concerning this. What happened in the workshops is that there were structured discussions and viewpoints were delivered from the experts. After the workshop, the core team made a conclusion concerning the needed number of product architectures, manufacturing architectures, and module area performance steps. This is a very crucial step—and much further detailed work has to be carried out later on in implementation. Table 1 shows a possible reduction in product architectures from 60 to 25. Perhaps, detailed studies will later show that, for example, 30 or 15 product architectures are better. This will, however, not change the main conclusion—that significant cost reductions are possible.

Phase 3

This is calculating (Step 7) the benefits in terms of direct material, direct labor, and CAPEX avoidance

Table 1. List of as-is architectures, to-be architectures, and financial impact.

Product line	As-is product architectures	To-be product architectures	Financial impact (% of turnover)
1	8	4	2.0
2	9	4	0.5
3	12	6	1.2
4	5	3	0.9
5	10	4	2.1
6	16	4	1.0
Total	60	25	

concerning tooling. The main inputs are the cycle plan, number of new architectures, and number of key module areas with "best of breed" cost/performance levels obtained in Step 3. This means that impact calculations are very conservative, that is, it is based on solutions and principles that are already available in the company today. The main results are summarized in Table 1. The work has lead to significant conclusions and discussions in the board of management.

Reduction in the number of product architectures. It is possible to reduce the number of product architectures significantly from 60 to 25 without comprising the number of commercial variants in the market. No one can for sure know whether this is completely true, but it seems that a significant reduction is possible. It has become clear to the board of management that the number of product architectures is strategic decision in the company that has to be anchored on senior vice president level. One vice president explained, "One architecture is very wrong—there will be bad cost/performance compromises. On the other hand 20 architectures is also wrong—this will lead to high complexity on and unfocused R&D effort."

Additional benefits of fewer architectures. It is the assumption that the benefits in Table 1 are only the top of the iceberg. There are additional savings in terms of reduced ware house cost, due to fewer module areas and part number. The efficiency in factories should increase due to fewer change-overs on the assembly lines. It should also be possible to introduce later customer order decoupling points, which should reduce delivery time. Furthermore, it should be possible to increase utilization level in factories, due to fewer parts, modules, and architectures. Furthermore, additional savings can be expected in purchase due to higher purchasing volume.

From an R&D perspective, fewer architectures means that the R&D effort on each architecture could be increased. This should again lead to increased quality, higher level of innovation, and reduction in time to market for new variants.

Product line design principles. During the work, it has been clear that some product lines are fundamentally wrongly designed. The engineering design approach has been wrong. It means that high-end products have been designed first and then the approach has been to "strip" them to reach mid- and low-end markets. The results have been that the costs for mid- and low-end products are too high. One R&D manager explained, "Stripping a Rolls Royce will not lead to a cost effective Polo car." The conclusion is that every module area design should in the future be based on scale up thinking rather than scale down thinking and part of one or more well-defined architectures.

Implementation. Two fundamental implementation alternatives are being considered. The traditional organization could drive implementation according to the approved cycle plans. The implementation time would then be approximately 7 years for all product lines. Another alternative is to establish a separate product and manufacturing architecture organization that has the full responsibility for all product lines and manufacturing. This would reduce implementation time but increase CAPEX. So far, no conclusions have been made. Another concern is the coordination between shared product architectures and increased automation in assembly. These two initiatives naturally have to be coordinated. It would be waste of resources to automate product architectures that will be phased out. Implementing shared architectures and then afterward increased atomization might take too long time.

Discussion

In the state-of-art literature on platforms and engineering design in general, it is often the assumption that concepts for the future product program have to be developed in order to evaluate cost reduction potentials. For practical reasons, this will not be possible in large global companies, so another approach is necessary to evaluate financial impact of shared architectures. The main contribution in this article is a top–down reasoning approach. This means reasoning from what is required in the market and relating this to the number of as-is product architectures. Hereby, the mismatch between market requirement and the current number of product architectures should be recognized and the ideal future number of architectures is identified. Compared to a real conceptualization project, the AME method will not provide financial benefits with the same level of confidence, but still good enough to evaluate whether it is relevant to continue working toward shared and fewer product architectures.

The AME method is very dependent on senior people in an organization, the top–down reasoning from the current state to future state is often difficult and there might be conflicting opinions. The viewpoint of the authors is that even though the "ideal" number of architectures is slightly higher or lower, it will not change the main conclusions. This means that a significant reduction in the number of product architectures is possible without compromising the market coverage.

Concerning application to the AME, the ideal company is mass producing with a history of mergers and acquisitions, distributed R&D, and manufacturing. Due to mergers and acquisitions, there will often be product lines with overlapping products. Due to distributed R&D and manufacturing, there will often be misalignment, that is, reinventing the "wheel" examples.

Conclusion

The article has presented a relatively simple method for calculating the benefits of shared architectures, the so-called AME method. There are three major contributions. The first one is an operational way to describe and count the number of product architectures. Second, the cost performance mapping shows in simple way how the performance steps of modules are realized. Number 3 contribution is top–down reasoning concerning the number of product and production architectures. From a practical point of view, the main contribution is the increased ability to have strategic discussion on the right number of architectures in a company based on facts.

There are many improvement areas in the AME method. One of them is finding out how to reason from requirements in the market, to the number of product architectures, and to the number of manufactured architectures. There must be sound principles for obtaining the right balance between the product and manufacturing architectures. Second, it should also be possible to include other quantifiable benefit dimensions such as time to market, R&D efficiency, and complexity reduction in manufacturing.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Andreasen MM (1980) *Machine design methods based on a systematic approach—contribution to a design theory*. PhD Thesis, Department of Machine Design, Lund University, Lund (in Danish).

Andreasen MM and Olesen J (1990) The concept of dispositions. *Journal of Engineering Design* 1: 17–36.

Bruun HPL (2015) *PLM support to architecture based development: contribution to computer-supported architecture modelling*. PhD Thesis, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby, 188 pp. (DCAMM Special report no. S174).

Bruun HPL, Mortensen NH and Harlou U (2014) Interface diagram: design tool for supporting the development of modularity in complex product. *Concurrent Engineering: Research and Applications* 22: 62–76.

Du X, Jiao J and Tseng MM (2001) Architecture of product family: fundamentals and methodology. *Concurrent Engineering: Research and Applications* 9: 309–325.

Ericsson A and Erixon G (eds) (1999) *Controlling Design Variants—Modular Product Platforms*. Dearborn, MI: Society of Manufacturing Engineers.

Fixson SK (2005) Product architecture assessment: a tool to link product, process, and supply chain design decisions. *Journal of Operations Management* 23: 345–369.

Gonzalez-Zugasti J, Otto KN and Baker JD (2000) A method for architecting product platforms. *Research in Engineering Design* 12: 61–72.

Guo F and Gershenson JK (2007) Discovering relationships between modularity and cost. *Journal of Intelligent Manufacturing* 18: 143–157.

Guðlaugsson TV, Ravn PM, Mortensen NH, et al. (2014) Front-end conceptual platform modeling. *Concurrent Engineering: Research and Applications* 22(4): 267–276.

Hansen CL (2015) *On the identification of architectures for product programs In a complexity cost perspective*. PhD Thesis, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby.

Hansen CL, Mortensen NH and Hvam L (2012) On the market aspect of product program design: towards a definition of an architecture of the market. In: *12th international design conference—design*, Dubrovnik, 21–24 May.

Harlou U (2006) *Developing Product Families Based on Architectures—Contribution to a Theory of Product Families*. Kongens Lyngby: Department of Mechanical Engineering, Technical University of Denmark.

Herrmann JW, Cooper J, Gupta SK, et al. (2004) New directions in design for manufacturing. IDETC/CIE, pp. 1–9. Available at: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid = 1651563

Huang GQ, Zhang XY and Liang L (2005) Towards integrated optimal configuration of platform products, manufacturing processes, and supply chains. *Journal of Operations Management* 23: 267–290.

Hubka V (1973) *Theorie Technischer Systeme—Grundlagen einer wissenschaftlichen Konstruktionslehre*. Berlin, Heidelberg: Springer-Verlag (in German).

Hultink EJ, Griffin A, Hart S, et al. (1997) Industrial new product launch strategies and product development performance. *Journal of Product Innovation Management* 14: 243–257.

Hvam L, Mortensen NH and Riis J (2008) *Product Customization*. Berlin, Heidelberg: Springer-Verlag.

Kester L, Hultink EJ and Griffin A (2013) An empirical exploration of the antecedents and outcomes of NPD portfolio success. Available at: http://repository.tudelft.nl/view/ir/uuid%3A276695e8-1a3e-4914-a6d9-db8b8937cc3e/

Krause D, Eilmus S and Jonas H (2013) Developing modular product families with perspectives for the product program. In: Abramovici M and Stark R (eds) *Smart Product Engineering*. Berlin, Heidelberg: Springer-Verlag, pp. 543–552.

Levandowsky C, Michaelis MT and Johannesson H (2014) Set-based development using an integrated product and manufacturing system platform. *Journal of Concurrent Engineering: Research and Applications* 22(3): 40–58.

Lindemann U, Maurer M and Braun T (eds) (2009) *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin: Springer.

MacDuffie JP (2013) Modularity-as-property, modularization-as-process, and "Modularity"-as-frame: lessons from product architecture initiatives in the global automotive industry. *Global Strategy Journal* 3: 8–40.

Meyer MH and Lehnerd AP (eds) (1997) *The Power of Product Platforms—Building Value and Cost Leadership*. New York: The Free Press.

Olesen J (1992) *Concurrent development in manufacturing—based on dispositional mechanisms*. PhD Thesis, Department of Engineering Design, Technical University of Denmark, Kongens Lyngby.

Otto KN and Wood KL (1998) Product evolution: a reverse engineering and redesign methodology. *Research in Engineering Design* 10: 226–243.

Pimmler TU and Eppinger SD (1994) Integration analysis of product decompositions. Design Engineering Division, American Society of Mechanical Engineers. Available at: http://web.mit.edu/eppinger/www/pdf/Pimmler_DTM1994.pdf

Skyttner L (2005) *General Systems Theory*. Singapore: World Scientific Publishing Co.

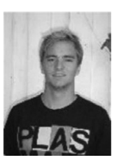
Stone RB, Wood KL and Crawford RH (2000) Using quantitative functional models to develop product architectures. *Design Studies* 21: 239–260.

Ulrich K (1995) The role of product architecture in the manufacturing firm. *Research Policy* 24: 419–440.

Author biographies

Niels Henrik Mortensen is professor of product development and do research on methods for developing product families.

Christian Lindschou Hansen is a PhD student focusing on complexity cost calcation.



Martin Løkkegaard is a PhD student focusing on development of system architectures.

Lars Hvam is professor of configuration management and do research on configuration system development and complexity cost calculation.

References

The references included here are used throughout the thesis, but do not include references only appearing in the appended papers.

Anderson, S.R., Kaplan, R.S., (2007). Time-Driven Activity-Based Costing: A Simpler and More Powerful Path to Higher Profits. Harvard Business School Press Books

Andreasen, M.M., (1994). Modelling - the language of the designer. Journal of Engineering Design. 5, 103-115

Andreasen, M.M., 1980. Machine Design Methods based on a Systematic Approach - contribution to a design theory, Ph.d. thesis. (Danish) Department of Machine Design, Lund University, Sweden

Andreasen, M.M., Hansen, C.T., Mortensen, N.H., (1996). The Structuring of Products and Product Programmes. Proceedings of WDK2, 4, 28

Andreasen, M., Mortensen, N., Harlou, U., (2004). Multi Product Development–New Models and Concepts, 75-86

Andreasen, M.M., (2011). 45 years with design methodology. J. Eng. Des. 22, 293-332

Andreasen, M.M., Howard, T.J., Bruun, H.P.L., (2014). Domain Theory, its models and concepts, in: An Anthology of Theories and Models of Design. Springer, pp. 173-195

Andreasen, M M, Hein, L. (Eds.), (1987). Integrated Product Development. Springer, Berlin

Andreasen, M.M., Olesen, J., (1990). The Concept of Dispositions. Journal of Engineering Design. 1, 17-36

Andreasen, M.M., (2009). Complexity of industrial practice and design research contributions -we need consolidation. DFX: Proc. Symp. Des. X, 1-9

Ashby, W., (1956). Ross: An introduction to cybernetics. London: Champman & Hall

Baldwin, C Y, Clark, K.B. (Eds.), (2000). Design rules: The power of modularity. The MIT Press

Blessing, L T M, Chakrabarti, A. (Eds.), (2009). DRM, a design research methodology. Springer, Dordrecht

Boothroyd, G., Dewhurst, P., (1983). Design for Assembly: Selecting the Right Method. Machine Design. 55, 94-98

Büchler, H., (1990). Bedeutung der Produktgestaltung und Logistik für die Wettbewerbfaehigkeit von Industrieunternehmen.

Buiga, A., (2012). Investigating the Role of MQB Platform in Volkswagen Group's Strategy and Automobile Industry. International Journal of Academic Research in Business & Social Sciences. 2

Buur, J (Ed.), (1990). A theoretical approach to mechatronics design. Institute for Engineering Design, Technical University of Denmark

Calinescu, A., Efstathiou, J., Sivadasan, S., Schirn, J., Huaccho Huatuco, L., (2000). Complexity in manufacturing: an information theoretic approach, 30-44

Campagnolo, D., Camuffo, A., (2010). The Concept of Modularity in Management Studies: A Literature Review. International Journal of Management Reviews. 12, 259-283

Checkland, P., Holwell, S., (1998). Action Research: Its Nature and Validity. Systemic Practice and Action Research. 11, 9-21

Closs, D.J., Jacobs, M.A., Swink, M., Webb, G.S., (2008). Toward a theory of competencies for the management of product complexity: six case studies. J. Oper. Manage. 26, 590-610

Cook, T D, Campbell, D.T., Day, A. (Eds.), (1979). Quasi-experimentation: Design & analysis issues for field settings. Houghton Mifflin Boston

Cooper, R., (1992). Formal organization as representation: remote control, displacement and abbreviation. Rethinking Organization: new directions in organization theory and analysis, 254–72

Cooper, R., Kaplan, R.S., (1988). Measure Costs Right - Make the Right Decisions. Harvard Business Review. 66, 96-103

Coughlan, P., Coghlan, D., (2002). Action Research for Operations Management. International Journal of Operations & Production Management. 22, 220-240

Czarniawska, B, Mouritsen, J. (Eds.), (2009). What is the object of management? How management technologies help to create manageable objects. Oxford University Press, USA

Du, X., Jiao, J., Tseng, M.M., (2001). Architecture of Product Family: Fundamentals and Methodology. Concurrent Engineering. 9, 309-325

Duffy, A.H.B., Andreasen, M.M., (1995). Enhancing the Evolution of Design Science. WDK WORKSHOP DESIGN KONSTRUKTION. 1, 29-35

Ericsson, A, Erixon, G. (Eds.), (1999). Controlling Design Variants - Modular Product Platforms. Society of Manufacturing Engineers, Dearborn, Michigan

Feitzinger, E., Lee, H.L., (1997). Mass Customization at Hewlett-Packard: The Power of Postponement. Harvard Business Review. 75, 116-121

Fine, C (Ed.), (1998). Clockspeed - Winning Industry Control in the Age of Temporary Advantage. Perseus Books, Reading, MA

Finger, S., Dixon, J.R., (1989). A review of research in mechanical engineering design. Part I: Descriptive, prescriptive, and computer-based models of design processes. Research in engineering design. 1, 51-67

Fixson, S.K., (2004). Assessing product architecture costing: Product life cycles, allocation rules, and cost models. Proceedings of the ASME Design Engineering Technical Conference. 1, 857-868

Frizelle, G (Ed.), (1998). The management of complexity in manufacturing: A strategic route map to competitive advantage through the control and measurement of complexity. Business Intelligence, London

Frizelle, G., Woodcock, E., (1995). Measuring complexity as an aid to developing operational strategy. International Journal of Operations & Production Management. 15, 26-39

Galbraith, J R (Ed.), (1973). Designing complex organizations. Addison-Wesley Longman Publishing Co., Inc.

Gershenson, Prasad, Zhang, (2003). Product modularity: definitions and benefits. J. Eng. Des. (UK). 14, 295-313

Gonzalez-Zugasti, J., Otto, K.N., Baker, J.D., (2000). A Method for Architecting Product Platforms. Research in Engineering Design. 12, 61-72

Hansen, C.L., Mortensen, N.H., (2014). Proactive identification of scalable program architectures: How to achieve a quantum-leap in time-to-market. International Journal of Industrial Engineering 2014, Vol. 21(2)

Hansen, C.L., Mortensen, N.H., Hvam, L., (2012). On the Market Aspect of Product Program Design: Towards a Definition of an Architecture of the Market. 12th International Design Conference - Design 2012

Hansen, C.T., Andreasen, M.M., (2002). Two approaches to synthesis based on the domain theory, in: Engineering design synthesis. Springer, pp. 93-108

Harlou, U., 2006. Developing Product Families Based on Architectures - Contribution to a Theory of Product Families. Department of Mechanical Engineering, Technical University of Denmark, Kgs. Lyngby

Hubka, V., Eder, W.E., (1988). Theory of technical systems: a total concept theory for engineering design. Berlin and New York, Springer-Verlag, 1988, 291 p. 1

Jacobs, M.A., Swink, M., (2011). Product portfolio architectural complexity and operational performance: Incorporating the roles of learning and fixed assets. J. Oper. Manage. 29, 677-691

Jacobs, M.A., (2013). Complexity: Toward an empirical measure. Technovation. 33, 111-118

Jiao, J., Simpson, T.W., Siddique, Z., (2007). Product Family Design and Platform-based Product Development: A State-of-the-Art Review. Journal of Intelligent Manufacturing. 18, 5-29

Joergensen, K., (1992). Videnskabelige arbejdsparadigmer (In Danish). Institute for Production, Aalborg University of Technology, Denmark

Johannesson, H., (2013). Emphasizing Reuse of Generic Assets Through Integrated Product and Production System Development Platforms, in: Simpson, T.W., Jiao, J.(., Siddique, Z., Höltta-Otto, K. (Eds.), . Springer New York, pp. 119-146

Karlsson, C (Ed.), (2008). Researching Operations Management. Routledge, New York, N.Y.

KPMG, (2011). Confronting Complexity - how business globally are taking on the challenges and opportunities

Krause, D., Eilmus, S., Jonas, H., (2013). Developing Modular Product Families with Perspectives for the Product Program, in: Smart Product Engineering. Springer, pp. 543-552

Kvist, M., 2009. Product Family Assessment, Ph.d. thesis. DTU Management, The Technical University of Denmark, Kgs. Lyngby

Langenberg, K.U., Seifert, R.W., Tancrez, J., (2012). Aligning supply chain portfolios with product portfolios. Int J Prod Econ. 135, 500-513

Latour, B., (1986). Visualization and cognition. Knowledge and society. 6, 1-40

Lawrence, P R, Lorsch, J.W., Garrison, J.S. (Eds.), (1967). Organization and environment: Managing differentiation and integration. Division of Research, Graduate School of Business Administration, Harvard University Boston, MA

Lechner, A., Klingebiel, K., Wagenitz, A., (2011). Evaluation of product variant-driven complexity costs and performance impacts in the automotive logistics with variety-driven activity-based costing. IMECS - Int. MultiConference Eng. Comput. Scientists. 2, 1088-1096

Leigh Star, S., (2010). This is Not a Boundary Object: Reflections on the Origin of a Concept. Science, Technology, & Human Values. 35, 601-617

Lindemann, U, Maurer, M., Braun, T. (Eds.), (2009). Structural Complexity Management: An Approach for the Field of Product Design. Springer, Berlin

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

Michelsen, A.U., Pagh, J., (2002). Mass Customization og Postponement - Strategies and Models for Integrated Production and Distribution of Customized Products (in Danish). En mosaik af dansk logistikforskning, 107-132

Miles, B.L., (1989). Design for Assembly - a Key Element within Design for Manufacture. Proceedings of the Institution of Mechanical Engineers. 203, 29-38

Miller, T., (2001). Modular engineering. An approach to structuring business with coherent, modular architectures of artifacts, activities, and knowledge, Technical University of Denmark, Department of Mechanical Engineering, DTU, Lyngby

Mortensen, N.H., 2000. Design modelling in a designer's workbench, PhD thesis. Department of Control and Engineering Design, Technical University of Denmark, Kgs. Lyngby

Mortensen, N.H., Hvam, L., Haug, A., Boelskifte, P., Hansen, C.L., Frobenius, S., (2010). Making Product Customization Profitable. International journal of industrial engineering. 17, 25-35

Mortensen, N.H., Pedersen, R., Kvist, M., (2008). Modeling and Visualizing Modular Product Architectures for Mass Customization. International Journal of Mass Customisation. 2, 216-239

Nielsen, O.F., 2010. Continuous Platform Development, Ph.d. thesis. DTU Management, The Technical University of Denmark, Kgs. Lyngby

Olesen, J., 1992. Concurrent Development in Manufacturing - based on dispositional mechanisms, Ph.d. thesis. Department of Engineering Design, Technical University of Denmark, Kgs. Lyngby

Pagh, J.D., Cooper, M., (1998). Supply Chain Postponement and Speculation Strategies: How to choose the right strategy. Journal of Business Logistics. 19, 13-33

Park, J., Simpson, T.W., (2008). Toward an activity-based costing system for product families and product platforms in the early stages of development. Int J Prod Res. 46, 99-130

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J.K., Mistree, F., (2000). Validating design methods and research: the validation square

Pedersen, R., 2009. Product Platform Modeling, Ph.d. thesis. DTU Management, The Technical University of Denmark, Kgs. Lyngby

Perona, M., Miragliotta, G., (2004). Complexity management and supply chain performance assessment. A field study and a conceptual framework. Int J Prod Econ. 90, 103-115

Ramdas, K., (2003). Managing Product Variety: An integrative review and research directions. Production and Operations Management. 12, 79-101

Rathnow, P J (Ed.), (1993). Integriertes Variantenmanagement: Bestimmung, Realisierung und Sicherung der optimalen Produktvielfalt. Vandenhoeck & Ruprecht

Rebovich, G, White, B.E. (Eds.), (2011). Enterprise systems engineering : advances in the theory and practice. CRC Press/Taylor & Francis, Boca Raton

Rode, J, Sant, K.L. (Eds.), (1983). UPS Fremgangsmåden. Institut for Produktudvikling, Kgs. Lyngby, Denmark

Salvador, F., De Holan, P.M., Piller, F., (2009). Cracking the Code of Mass Customization. MIT Sloan Management Review. 50, pp. 71

Sanchez, R., (2000). Modular Architectures, Knowledge Assets and Organizational Learning: New management processes for product creation. International Journal of Technology Management. 19, 610-629

Schuh, G (Ed.), (2001). Produktkomplexität managen: Strategien-Methoden-Tools. Hanser Verlag, München

Simon, H.A., (1957). Models of man: Social and national. Wiley, New York

Simon, H.A., (1962). The Architecture of Complexity. Proc. Am. Philos. Soc. 106, 467-482

Simpson, (2004). Product platform design and customization: status and promise. (AI EDAM) Artif. Intell. Eng. Des. Anal. Manuf. (UK) Artificial intelligence for engineering design, analysis and manufacturing. 18, 3-20

Sivadasan, S., Efstathiou, J., Frizelle, G., Shirazi, R., Calinescu, A., (2002). An information-theoretic methodology for measuring the operational complexity of supplier-customer systems. ijopm. 22, 80-102

Skinner, W., (1974). The Focused Factory. Harvard Business Review. 52, 113-121

Star, S.L., Griesemer, J.R., (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. Social studies of science. 19

Suh, N.P., (2005). Complexity in engineering. CIRP Annals-Manufacturing Technology. 54, 46-63

Ulrich, K., (1995). The Role of Product Architecture in the Manufacturing Firm. Research Policy. 24, 419-440

Wiener, N (Ed.), (1948). Cybernetics. Hermann Paris

Wilson, S A, Perumal, A. (Eds.), (2009). Waging War on Complexity Costs, 1st ed. McGraw-Hill, Europe

Wu, Y., Frizelle, G., Efstathiou, J., (2007). A study on the cost of operational complexity in customer–supplier systems. Int J Prod Econ. 106, 217-229

On the identification of architectures for product programs

In a complexity cost perspective

Christian Lindschou Hansen

On the identification of architectures for product programs
Christian Lindschou Hansen