THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

On the Design of Functionally Integrated Aero-engine Structures

Modeling and Evaluation Methods for Architecture and Complexity

VISAKHA RAJA

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

On the Design of Functionally Integrated Aero-engine Structures:
Modeling and Evaluation Methods for Architecture and Complexity

ISBN: 978-91-7905-117-4
Series number: 4584
in the series Doktorsavhandlingar vid Chalmers tekniska högskola. Ny serie (ISSN 0346-718X)

Department Industrial and Materials Science
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone + 46 (0)31-772 10

Printed by
Chalmers Reproservice
Gothenburg, Sweden 2019

Abstract

The drive for airplanes with radically reduced fuel consumption and emissions motivates engine manufacturers to explore innovative engine designs. The novelty of such engines results in changed operating conditions, such as newly introduced constraints, increased loads or rearranged interfaces. To be competitive, component developers and manufacturers must understand and predict the consequences of such changes on their sub-systems. Presently, such assessments are based on detailed geometrical models (CAD or finite element) and consume significant amounts of time. The preparation of such models is resource intensive unless parametrization is employed. Even with parametrization, alternative geometrical layouts for designs are difficult to achieve. In contrast to geometrical model-based estimations, a component architecture representation and evaluation scheme can quickly identify the functional implications for a system-level change and likely consequences on the component. The schemes can, in turn, point to the type and location of needed evaluations with detailed geometry. This will benefit the development of new engine designs and facilitate improvements upon existing designs. The availability of architecture representation schemes for functionally integrated (all functions being satisfied by one monolithic structure) aero-engine structural components is limited.

The research in this thesis focuses on supporting the design of aero-engine structural components by representing their architecture as well as by developing means for the quantitative evaluation and comparison of different component designs. The research has been conducted in collaboration with GKN Aerospace Sweden AB, and the components are aero-engine structures developed and manufactured at GKN. Architectural information is generated and described based on concepts from set theory, graph theory and enhanced function–means trees. In addition, the complexities of the components are evaluated using a new complexity metric. Specifically, the developed modeling and evaluation methods facilitate the following activities:

- identification and representation of function–means information for the component
- representation and evaluation of component architecture
- product complexity evaluation
- early selection of load path architecture
- impact assessment for the component's functioning in the system

By means of the methods developed in this thesis, the design rationale for a component is made explicit, and the storing, communicating and retrieving of information about the component in the future is enabled. Through their application to real-life engine structures, the usability of the methods in identifying early load carrying configurations and selecting a manufacturing segmenting option is demonstrated. Together, the methods provide development engineers the ability to compare alternative architectures. Further research could focus on exploring the system (engine) effects of changes in component architecture and improvements to the complexity metric by incorporating manufacturing information.

Keywords: Product Development, Aero-engine Structures, Function–Means Modeling, Configurable Components, Product Architecture, Functionally Integrated Product Architecture, Load Paths, Structural Complexity, Design Product Complexity

Acknowledgements

This research project was carried out at the Systems Engineering Design (SED) research group in the Product Development division of the Industrial and Materials Science Department at Chalmers University of Technology, Göteborg, Sweden. The financial support by NFFP, the National Aeronautical Research Programme, jointly funded by the Swedish Armed Forces, Swedish Defense Materiel Administration (FMV) and Swedish Governmental Agency for Innovation Systems (VINNOVA) is acknowledged. The support has been provided through VINNOVA projects with reference numbers 2013-01185 and 2017-04862.

Although this thesis has only one person's name on it, a number of people have been instrumental in forming works behind it. First, I thank my employer GKN Aerospace Sweden AB for allowing me to pursue this research project. Then, I thank my supervisor Prof. Ola Isaksson (Chalmers University of Technology) and assistant supervisors, Prof. Emeritus. Hans L Johannesson (Chalmers University of Technology), and Prof. Michael Kokkolaras (McGill University, Quebec, Canada) for all the advice, motivation, and support. I am especially grateful for their immense, endless patience with me throughout this project.

I thank my industrial supervisor Dr. Andreas Borg at the Design Engineering department of GKN Aerospace for the discussions and support in all project and even non project related matters. Thanks to Dr. Jonas Landahl and Jakob Muller at the SED research group for the very interesting discussions. I also thank Dr. Petter Andersson (GKN), Prof. Tomas Grönstedt and Sebastian Samuelsson (both at the Fluid Dynamics division of the Mechanics and Maritime Sciences department at Chalmers) for all the discussions and their kind participation in the initial stages of this research.

There are a number of colleagues at GKN who have extended their help and advice at different phases of this research. Many such gestures have been instrumental in how this research is formed and I gratefully acknowledge these actions. Although most of the research has been performed at GKN, visits and discussions at the university have always been enlightening. I have always felt welcomed at the Industrial and Materials Science department. I consider myself very privileged to be associated with Chalmers University of Technology.

I call to mind all those individuals, my teachers as well as others, who helped me learn at various stages of my life.

I would like to thank my family for their unconditional well wishes. Heartfelt thanks for my wife Gayathri for her constant reminder that even without the research, there will still be air in the world for me to breathe. You keep me sane.

03 May 2019,
Trollhättan.

Visakha Raja

Appended Publications

Paper A Raja, V. and Isaksson, O. (2015) ‘Generic Functional Decomposition of an Integrated Jet Engine Mechanical Sub System Using a Configurable Component Approach’, *22nd ISPE Inc. International Conference on Concurrent Engineering, CE 2015*. TU Delft, Netherlands, July 20-23.

Paper B Raja, V., Samuelsson, S., Isaksson, O. and Grönstedt, T. (2015) ’Exploring Influence of Static Engine Component Design Variables on System Level Performance’, *22nd International Symposium on Air Breathing Engines, ISABE2015*. Phoenix, Arizona, October 25-30.

Paper C Raja, V., Johannesson, H. and Isaksson, O. (2018). ‘Describing and evaluating functionally integrated and manufacturing restricted products’, *Research in Engineering Design*, 29(3), pp. 367-391.

Paper D Raja, V., Isaksson, O. and Kokkolaras, M. (2017) ’An optimization-based approach for supporting early product architecture decisions’, *21st International Conference on Engineering Design, ICED 2017.* Vancouver, Canada, August 21-25.

Paper E Raja, V. and Isaksson, O. (2019) ‘Modelling Integrated Product Architectures: An Aero Engine Component Example’, *7th International Conference on Research Into Design*. Indian Institute of Science, Bangalore, India, January 9 - 11

Paper F Raja, V., Isaksson, O. and Kokkolaras, M. (2019) ‘A Simulation-assisted Complexity Metric for Design Optimization of Integrated Architecture Aero-engine Structures’, *Structural and Multidisciplinary Optimization* (Accepted for publication 03 May 2019)

Work Distribution

Paper A	**Raja** conceptualized, performed the necessary work, and wrote the paper. Isaksson contributed with comments and reviews.
Paper B	**Raja** and Samuelsson conceptualized the paper. Samuelsson performed the specific fuel consumption (SFC) calculations using pressure drop data supplied by **Raja**. **Raja** wrote the paper with assistance from Samuelsson. Isaksson and Grönstedt contributed with comments and reviews.
Paper C	**Raja** conceptualized, performed the necessary work, and wrote the paper. Johannesson contributed to the CC theory, created the CC models, and functioned as a reviewer. Isaksson contributed with comments and reviews.
Paper D	**Raja** conceptualized, performed the necessary work, and wrote the paper. Isaksson and Kokkolaras contributed with comments and reviews.
Paper E	**Raja** conceptualized, performed the necessary work, and wrote the paper. Isaksson contributed with comments and reviews.
Paper F	**Raja** conceptualized, performed the necessary work, and wrote the paper. Isaksson and Kokkolaras contributed with comments and reviews.

Table of Contents

1 Introduction ... 1
1.1 Background ... 1
1.2 Aero-engine Structures and GKN Aerospace ... 2
1.3 Research Motivation ... 4
1.4 Research Questions ... 5
1.5 Delineation of the Research and Terminology ... 6
1.6 Outline of the Thesis ... 7
2 Frame of Reference ... 9
2.1 Product Development and Engineering Design ... 9
2.1.1 Design Theories ... 10
2.1.2 Design Research ... 10
2.1.3 Summary ... 10
2.2 Product Architecture ... 10
2.2.1 Classification ... 11
2.2.2 Significance ... 12
2.2.3 The Functionally Integrated Architecture ... 13
2.3 Architecture Modeling Approaches ... 13
2.3.1 Function–Means Modeling ... 13
2.3.2 Platform Modeling with Configurable Components ... 15
2.3.3 Design Structure Matrix ... 16
2.3.4 Axiomatic Design Matrix ... 17
2.3.5 Graphs and Graph Centralities ... 18
2.3.6 Singular Value Decomposition ... 19
2.4 Complexity in Engineering Design ... 20
2.4.1 Definitions ... 20
2.4.2 Classification ... 21
2.4.3 Complexity Metrics ... 21
2.5 Engine Structure Development Concepts ... 22
2.5.1 Sections and Manufacturing Options ... 22
2.5.2 Load Paths ... 23
2.5.3 Flow Paths ... 25
2.6 System and Component Interactions ... 25
3 Research Approach ... 29
3.1 Methodologies for Design Research ... 29

3.1.1 Jørgensen's Model ... 29
3.1.2 The Design Research Methodology ... 30
3.2 Research Quality ... 31
3.2.1 Scientific Confidence: The Validation Square ... 32
3.2.2 Industrial Confidence: Technology Readiness Levels ... 33
3.3 Research Approach in the Present Thesis ... 34
3.3.1 Research Data ... 34
3.3.2 Research Progression ... 35
4 Results and Appended Paper Summaries ... 37
4.1 Paper A: Function–Means for Engine Structures ... 37
4.1.1 Objective ... 37
4.1.2 Approach ... 37
4.1.3 Results and Findings ... 38
4.2 Paper B: System Influence of Engine Structures ... 39
4.2.1 Objective ... 39
4.2.2 Approach ... 39
4.2.3 Results and Findings ... 40
4.3 Paper C: A Product Architecture Description Method ... 40
4.3.1 Objective ... 40
4.3.2 Approach ... 40
4.3.3 Results and Findings ... 41
4.4 Paper D: Optimization-Based Architecture Decisions ... 43
4.4.1 Objective ... 43
4.4.2 Approach ... 43
4.4.3 Results and Findings ... 44
4.5 Paper E: Demonstration of Architecture Description ... 44
4.5.1 Objective ... 44
4.5.2 Approach ... 44
4.5.3 Results and Findings ... 45
4.6 Paper F: A Complexity Metric for Engine Structures ... 46
4.6.1 Objective ... 46
4.6.2 Approach ... 46
4.6.3 Results and Findings ... 47
4.7 Summary of Major Results ... 48
5 Discussion ... 49
5.1 Answering the Research Questions ... 49
5.1.1 RQ1 ... 49

5.1.2 RQ2 49
5.1.3 RQ3 50
5.1.4 RQ4 51
5.2 Research Quality 52
5.2.1 Scientific Confidence: Applying the Validation Square 52
5.2.2 Industrial Confidence: Technology Readiness Levels Assessment 54
6 Conclusion and Further Work 57
6.1 Scientific Contribution 57
6.2 Industrial Contribution 58
6.3 Further Research 59
REFERENCES 61

Appendix
Errata to Paper C
Appended Papers
PAPER A
PAPER B
PAPER C
PAPER D
PAPER E
PAPER F

List of Abbreviations

Abbreviation	**Expansion**
ADM	Axiomatic Design Matrix
C	Constraint
CC	Configurable Component
CFD	Computational Fluid Dynamics
CO	Component
DP	Design Parameter
DR	Design Rationale
DS	Design Solution
DSM	Design Structure Matrix
EF-M	Enhanced Function-Means
EU	European Union
FE	Finite Element
F-M	Function-Means
FMEA	Failure Mode and Effects Analysis
FR	Functional Requirement
GE	General Electric
HP	High Pressure
icb	is constrained by
iib	is influenced by
isb	is solved by
LCC	Load Carrying Configuration

LP	Low Pressure
MDO	Multidisciplinary Optimization
OEM	Original Equipment Manufacturer
PAG	Parametric Associativity Graph
RQ	Research Question
RRSP	Risk and Revenue Sharing Partnership
SOP	Subtract and Operate Procedure
TRS	Turbine Rear Structure

1 Introduction

Today, the systems in an aero-engine are highly optimized and complex, which is a result of long-standing, incremental improvements over previous versions. Designs for various engine modules, such as compressors or turbines and associated components, are proven, and designers are aware of where to look for improvements in later versions of an engine. As requirements for operating costs, emissions and environmental effectiveness become more demanding, engine manufacturers explore and demonstrate novel engine architectures (SAFRAN, 2017) that lead to new component designs. To remain competitive, component developers will need to understand the conditions that engine architectures impose and perform the design accordingly. This does not suggest that previously performed designs can be neglected, as existing development and manufacturing methods are tailored for such designs. Effective component design, whether incremental or new, depends on the practical appraisal of requirements for the component and the existing and proposed means to satisfy these requirements.

This thesis aims to enable design assessments on engine components that significantly deviate from existing designs due to newer engine architectures as well as to improve upon prevailing designs. Components are incrementally developed with a high degree of functional integration. Functions such as channeling fluid flows or transferring mechanical loads are satisfied by a single, monolithic structure. This makes it difficult to obtain an overall understanding of the components based on the functions they satisfy, the manner in which the functions are satisfied, the various implications of the functions (structural or fluid flow related) and manufacturing options (e.g. cast or fabricated). Presently, methods that can facilitate such understanding, considering both the functional and solution (physical) aspects of the design, are lacking. Most methods consider aspects in isolation (e.g. creating a function structure or performing finite element [FE] simulations), and a need exists for methods that combine all aspects applicable to aero-engine components. This thesis focuses on developing methods for modeling and evaluating aero-engine components, considering the functional and physical aspects of their design. The methods are developed based primarily on concepts from product architecture and complexity theory. The research was performed at GKN Aerospace Sweden AB, Trollhättan, and the components considered are engine structures developed and manufactured by GKN.

1.1 Background

It is very unlikely that anyone who reads this thesis has not taken a flight to a destination. The number of people who fly and flight departures have been rising steadily. Between 2016 and 2017, the International Civil Aviation Organization (ICAO) reported a passenger increase of 300 million and a flight departure increase of one million (ICAO, 2017). This growth in commercial aviation increases an airline's revenue, but to maintain profits, operating costs must be lowered. The costs of fuel and oil and maintenance and overhaul alone account for more than a third of the expenses incurred by airlines (Ferjan, 2016). In addition to managing costs, commercial aviation should meet stringent demands on environmental friendliness and safety. The EU aviation research and innovation agenda aims to reduce CO_2 and NO_x emissions and noise levels by 75%, 90% and 65%, respectively, by 2050, compared to 2000 levels (ACARE, 2017). Accident levels are to be reduced to less than one per one million departures, from the current level of 2.3 per one million departures (ICAO, 2017). Aero-engines directly affect the operating costs, safety and environmental impacts of commercial aviation. For example, reducing engine fuel consumption significantly reduces operating costs. The specific fuel consumption (SFC) of jet engines consistently decreased between 1950 and 2000, and the trend is projected to continue with the aid of new technologies (Avellan, 2011). However, reducing fuel consumption has the effect of increasing NOx emissions (Saravanamuttoo, 2009, p. 35), which must be addressed by developing the appropriate technology. It is evident that the effective design and development of engines and their components are important to meet aviation goals.

The nature of development is affected by the nature of the design: whether the design is new or incremental. For the same aircraft, two engine original equipment manufacturers (OEMs) can offer the same performance improvements when following different design approaches. The availability of two engine options for the Airbus A320NEO aircraft, from Pratt & Whitney Commercial Engines and CFM International Aero Engines, is a case in point. Pratt & Whitney focused on a new engine design (improving efficiency through slower intake fan rotations by introducing a gear box, a so-called geared turbofan engine), while CFM focused on incremental design (improving the efficiency of engine modules, such as combustors and compressors) to offer the same performance enhancement (Martin, 2016). From the perspective of an aero-engine component developer, demands arising from both new and incremental designs must be met.

For commercial aircraft engines, product development resembles the spiral framework common in the software industry (Hague, 2001). After each development phase, following integration and testing, new conditions may arise that must be met. For instance, subsequent to an engine test, the loads on an engine component can differ greatly from the initial set of loads used to decide its geometry. The updated set of loads can make the geometry fail to meet the structural requirements for the component. If the component satisfies multiple functions, consequent redesign will be time consuming, as design analyses can span multiple disciplines (structural, aero-thermo and fluid flow analyses, for example). On the other hand, if the component design accounts for possible load variations, it is likely to be undesirably heavy. A change of geometry will also impose consequences for manufacturing. A difficult geometry might lead to improper cooling in the cast structure, which can form crack-prone brittle regions (known as alpha-cases in structures made from titanium alloys) that limit component life. Thus, demands can arise both due to the type of design and the nature of development of the aero-engine that causes far-reaching effects on engine components.

To effectively meet demands on a component, a developer must understand the functions of the component in the system in which it operates: what system parameters are influenced by the component and how. Developers also must understand ways in which the component satisfies the functional requirements (FRs) from the system. In other words, the architecture of the component must be better understood. Failure to understand the architecture will cause slower responses to system requirements, which may lead to cost overruns due to increased developmental efforts and propagation effects. A lack of understanding may also lead to missing sustainability targets and losing the developer's competitive position.

The foregoing paragraphs make clear that knowledge about system–component interaction and component architecture will enable developers to remain competitive. This research focuses on creating such an understanding regarding aero-engine structures developed and manufactured by GKN Aerospace.

1.2 Aero-engine Structures and GKN Aerospace

The primary function of an aero-engine is to generate the thrust necessary to propel an aircraft. Thrust is produced by the reaction of accelerating streams (a jet) of combustion gases, air or both. The type of engine that generates propulsive power from both air and combustion gas streams is known as a turbofan engine. Figure 1 displays the GE9X commercial turbofan aero-engine, which will power the Boeing 777x aircraft (GE Aviation, 2018). The fan creates most of the thrust by drawing in air from the atmosphere and accelerating it. A part of the air that the fan draws in is compressed in a series of low- and high-pressure compressors, mixed with fuel and burned in a combustor. The combustion gases are then expanded through a series of high- and low-pressure turbines to produce power, and the fan receives its necessary power from the turbine (such engines are also called gas turbine engines). The combustion gases exiting the final turbine stage are also accelerated (in addition to the fan accelerating the inlet air) to generate a smaller portion of the engine's thrust. Compared to other types of aero-engines, turbofans are superior in converting the energy contained in the fuel into useful thrust (Saravanamuttoo, 2009) and are the most common type of engines used in large commercial aircraft (National Academies of Sciences, 2016).

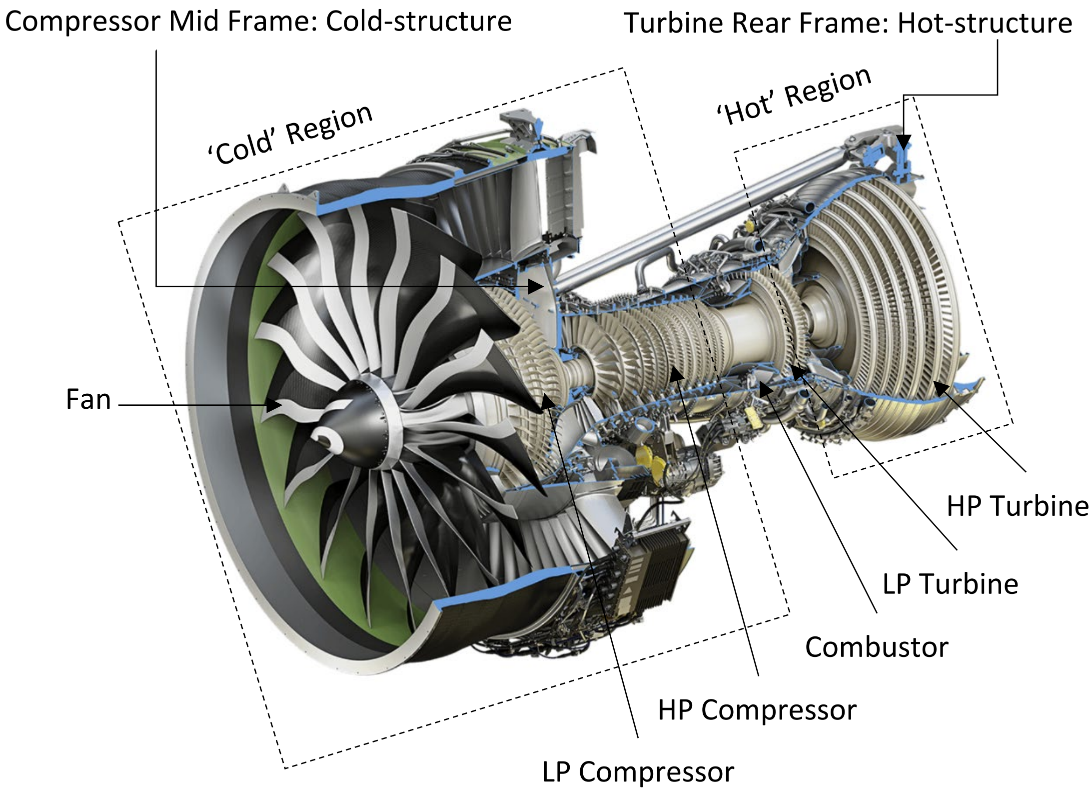


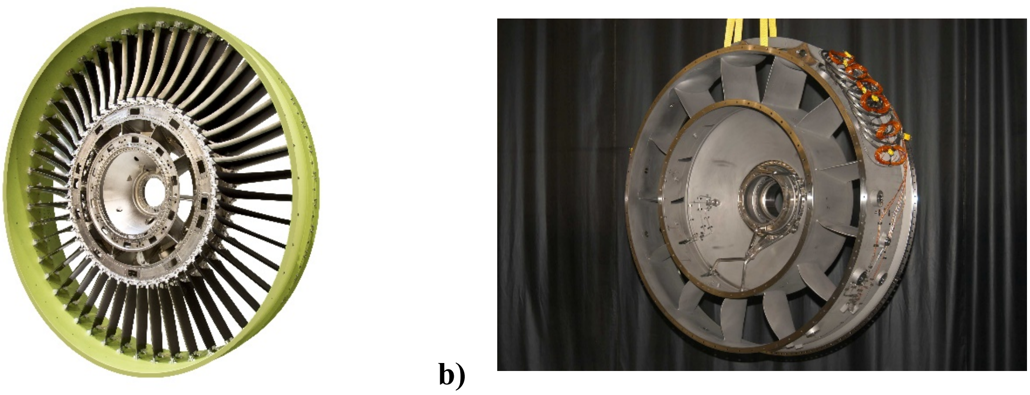
Figure 1 The GE9X aero-engine with regions and components marked. The acronyms LP and HP refer to low and high pressure, respectively. Locations of cold- and hot-structures are also marked in the figure. Image is from GE Aviation, GE9X Commercial Engine web-page (GE Aviation, 2018).

The components in an engine, particularly non-rotating structures, are often said to be hot or cold, depending on their location with respect to the combustor. Structures located upstream of the combustor that are not exposed to the hot combustion gases are called cold-structures, while those located downstream of the combustor exposed to the hot combustion gases are called hot-structures. Figure 2 displays typical cold- and hot-structures.

For obtaining higher component efficiencies, the compression and expansion in an engine are achieved in multiple stages, which are grouped in one or more compressors or turbines (Saravanamuttoo, 2009). For example, in the engine shown in Figure 1, compression is achieved in three and 11 stages in low- and high-pressure compressors, respectively (GE

Aviation, 2018). Similarly, the expansion is achieved in two turbines: the high- and low-pressure turbines. This necessitates that the flow be directed between the low- and high-pressure rotating components. Non-rotating structures perform this function. The flow is channeled along an annulus formed in the structure, with the inner and outer walls of the annulus being connected by means of guide vanes. The structures also house bearings for engine shafts and carry loads from the shafts to the engine's outer frame. In addition to the flow and load transfer functions, the structures provide mounting points for various measurement devices, allow bleed air tapping for different uses in the engine and transfer thrust to the engine by providing attachment points for airframe, among other functions. They are often manufactured as single-piece castings or are weld fabricated from a number of cast, forged or sheet metal segments.

It can be observed that in case of aero-engine static structures, a number of functions are satisfied by one single, monolithic component. Such products are called functionally integrated products, and the product architecture is termed functionally integrated product architecture. Subsection 2.2.3 further discusses this subject.



a) b)

Figure 2 Generic cold- and hot-structures; a compressor structures is shown in a), and a turbine structure is shown in b). Figures do not correspond to structures marked in Figure 1. Figures are courtesy of GKN Aerospace Sweden AB, Trollhättan.

Within GKN Aerospace, GKN Aerospace Engine Systems in Trollhättan develops, manufactures and maintains static aero-engine components for a range of engine architectures (engine types). GKN acts as both a build-to-print supplier as well as a risk and revenue sharing partner (RRSP) for various commercial engine manufacturers. In its RRSP role, GKN is responsible for developing technologies and design solutions so that OEMs can directly integrate the components into their engines. The engine structures that the title of this thesis refers to are the non-rotating structures (sometimes referred to as structural frames), which are part of the compressor or turbine modules of commercial turbofan engines.

1.3 Research Motivation

It is of mutual benefit to both engine OEMs and component developers to understand component behavior in the engine. Developers can closely meet the demands on their component while OEMs become aware of the performance limits of individual components. From a developer's perspective, the increased design responsibility through RRSP contracts puts additional pressure to comprehend the consequences of system design changes on their components. This requires that a developer is well aware of their component's production methods and architecture (how the product satisfies its function) so that development efforts can be directed to relevant areas. Traditionally, component developers possess production-related knowledge about their structures although design knowledge is largely experiential and is not formalized. Consequence evaluations of system level changes involve detailed simulations using advanced analyzes tools (such as FE or computational fluid dynamics [CFD] simulations). As new engine architectures are explored, the value of previous experience is reduced, that puts enhanced emphasis on detailed simulations, increasing the time to respond to system level changes. Presently, methods that enable understanding the effects of system level changes on the component, considering its architecture, production and influence from system parameters such as operational loads, are lacking. It is necessary to create methods and metrics to create such understanding and improve the product design so that a developer can quickly respond to system requirements.

In the scientific literature, modeling and evaluation methods for product architecture and complexity are often targeted to multi-part products or multi-component systems. Examples of such methods include design structure matrices (Eppinger and Browning, 2012) and directed, weighted graph-based complexity measures (Gokpinar *et al.*, 2010; Tamaskar *et al.*, 2014). Techniques to represent the architecture, especially its functional aspects using various means of functional decomposition, are well developed in the literature. van Eck *et al.* (2008) and (Eisenbart *et al.*, 2017) provide rigorous review of such techniques. Due to functional integration, applying such techniques to individual engine structures is not straightforward. Existing methods must be adapted and demonstrated on functionally integrated products such as aero-engine structures for practical utilization.

An industrial opportunity exists to develop methods supporting the design of functionally integrated products. Through this research, the validation of academically developed design support methods on real-life components is facilitated, while for industry, theoretically rigorous methods will be made available for design improvements. The design for performance research projects (Chalmers Research, 2013; Chalmers Research, 2017) have been formed to utilize this research opportunity.

1.4 Research Questions

The research questions are formed based on the research motivation detailed in Section 1.3. Since the products developed are intended to operate in a system, this research work can be summarized as answering two general questions:

A**.** *How does the product behave in the system?*

B. *How does the product adapt to the system?*

In both A and B, *product* refers to the aero-engine structures, while *system* refers to the complete aero-engine.

Referring to A, a first step is to obtain quantitative measures of the product's performance in the system. Here, the product is considered as a whole. To obtain a measure of the product's performance in the system (behavior), coupled disciplinary studies must be performed, and novel frameworks must be specified to perform the studies. It may also be necessary to perform simplifications of the component so that only the most sensitive information is coupled back and forth. Keeping Question A as a guide, a new RQ is formed:

RQ1*: What are the effects of engine structures on the performance of the engine?*

Referring to B, adaptation to the system is directly related to the internal organization or architecture of the product in response to system requirements. An understanding of how the product satisfies its functions is needed. Keeping Question B as a guide, a new RQ is formed:

RQ2*: What characterizes the architecture of the product and how to represent the architecture?*

Insights created from architecture studies must be implemented to benefit industrial practice. To benefit industrial usage, the following RQs are proposed (RQ3 and RQ4):

RQ3*: How can the architectural insights derived for the product be utilized in initial design stages?*

RQ4*: How can a quantitative metric be created for the architecture so that comparison among products of the same class is made easier?*

Concerning RQ3, a demonstration of the utility of the developed methods and understanding is intended.

With reference to RQ4, a class of products refers to a product family, wherein each member possesses common functional features (Jiao *et al.*, 2007). In case of this thesis, the products that are termed cold-structures form a class, and products that belong to hot-structures form another class.

1.5 Delineation of the Research and Terminology

Throughout the thesis, the terms "structures," "components" or "products" can refer to both cold and hot engine structures. The structure is a component in an engine, while it is a product of the developer. Since the product is functionally integrated, where only one monolithic component satisfies all functions required, the terms "product," "structure," "component" and "part" all have the same meaning. This thesis has the developer's view, and therefore the term "product" is frequently used to refer to the structures.

A design support in this thesis implies the *means, aids and measures that can be used to improve design* (Blessing and Chakrabarti, 2009). A method in this thesis is the specification of how a task must be performed (Estefan, 2007). The tasks are related to designing the aero-engine structure. A design contextually refers to both to the act of designing and the outcome of the act (both as verb and noun). In the action sense, design refers to engineering design, defined as the application of scientific and engineering knowledge to solve technical problems (Pahl *et al.*, 2007). It is the final geometry of the structure that is signified in the outcome sense of design.

The research is performed exclusively at GKN Aerospace in Trollhättan, which manufactures aero-engine structures. The generality of the research is thus limited to structures in aero-engines (and gas turbines upon which the engines are based). Even though the structures are part of a compressor or turbine module, design effects from the modules are not considered in detail.

1.6 Outline of the Thesis

In Chapter 1, the necessity of this research in light of aero-engine structures development is discussed and the research questions are formulated.

In Chapter 2, the specific research areas covered in this thesis with related developments are discussed.

In Chapter 3, the approach for performing and validating the research is detailed.

In Chapter 4, summaries of the publications based on which this thesis is formed are given.

In Chapter 5, the research questions are answered, and the research quality is discussed.

In Chapter 6, the concluding remarks are made.

The core of the research work is in the form of academic papers. The papers are appended to the end of this thesis.

2 Frame of Reference

This chapter positions the topics addressed in the thesis within a larger, foundational research area. It also offers a review of existing literature within the topics addressed. After a general introduction to product development, the specific topics are reviewed.

2.1 Product Development and Engineering Design

Since this thesis aims to assist the development of aero-engine structures, product development and engineering design concepts are briefly introduced before a discussion of relevant literature.

The activities involved, from market exploration through the design, production and delivery of a product for the market, are defined as product development (Ulrich and Eppinger, 1995). Approaches to development that consider the design and production in parallel, such as concurrent engineering (Prasad, 1996) and integrated product development (Andreasen and Hein, 2000), are popular. In such approaches, only one product concept is chosen to develop to manufacture, and they are often termed point-based development. A set-based development, wherein (e.g. set-based concurrent engineering by Sobek *et al.* (1999)) a number of design and associated manufacturing alternatives are simultaneously considered, is also prevalent. The alternatives are systematically eliminated to converge to a few, followed by the selection of only one alternative as the development progresses.

For managing development activities, a phased model with reviews at the end of each phase is adopted. The stage-gate model (Cooper, 1983; Cooper, 2001), shown in Figure 3, is one such approach that is widely used in the industry (Högman and Johannesson, 2013). The periodic gate reviews in the model are perhaps more familiar to development engineers than the specific approaches to development, such as concurrent engineering. In the "development" stage (Stage 3 in Figure 3), the final product is created based on business requirements. This stage is distinguished by engineering design, when engineers apply their scientific and engineering knowledge to the solution of technical problems (Pahl *et al.*, 2007). Engineering design is characterized by several methods intended to support various activities, such as concept generation and evaluation. The modeling and evaluation methods for architecture and complexity developed in this thesis support activities in engineering design.

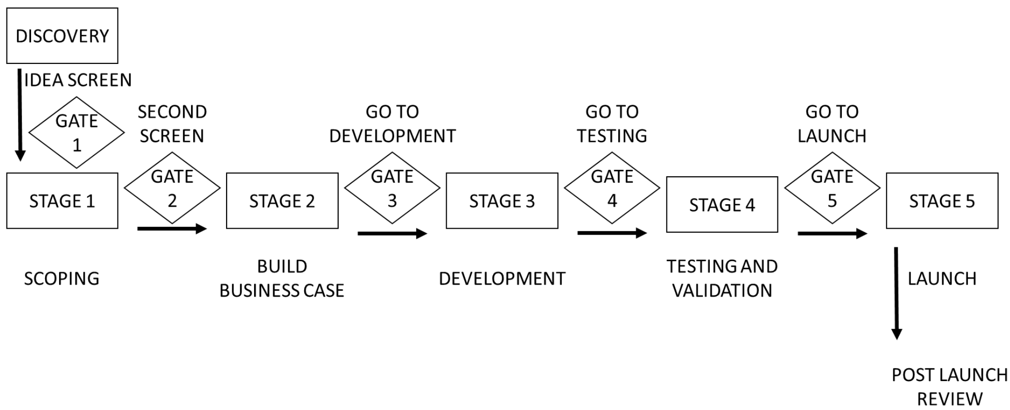


Figure 3 The stage-gate model of managing product development, adapted from Cooper (2001). Engineering design is predominant in the developmental stage (Stage 3).

2.1.1 Design Theories

Design theories help to understand the activities in engineering design. A number of design theories have been developed in recent years; Chakrabarti and Blessing (2015) provide a comprehensive review. The foundational ideas for engineering design methods are often grounded upon various design theories. Two such theories, upon which the architecture description methods in this thesis are founded on, are described below.

The domain theory (Andreasen, 1991; Hansen and Andreasen, 2002) describes design as occurring in three domains: a "transformation domain" that describes the effects of using a product, an "organ" domain that describes the means that create the effects and a "part" domain that describe the parts that make up the design solutions. Each domain is successively better defined than the preceding one. Designing occurs by moving back and forth across the domains, connecting elements in each domain. The connections are achieved using "a function–means" (F-M) law that establishes a hierarchical relation between two domains.

The axiomatic design theory (Suh, 1990) considers design as the mapping between a functional and a physical space. The functional and physical spaces are occupied by the specific requirements (called functional requirements, FRs) for the design and the characteristics of the physical embodiment of the design (called design parameters, DPs). Good designs can be described by two axioms (self-evident truths): an independence axiom that states that each FR for a design must be unaffected by other FRs and an information minimization axiom that states that the information content of the design should be minimized.

2.1.2 Design Research

Research into engineering design assists in developing knowledge, both for and about design (Horvath, 2001). Thus, design research builds knowledge about designing (the process of creating an artifact) and design (the object resulting after designing) (Reich and Subrahmanian, 2013). According to Blessing and Chakrabarti (2009), the two objectives of design research are (i) the formulation and validation of theories of design and (ii) the development and validation of support based on the theories. In essence, research performed to support engineering design is engineering design research. This thesis addresses the second objective of design research, the formulation and validation of support for design.

2.1.3 Summary

Many firms have different approaches to developing their products and managing the associated activities. Engineering design and the methods therein are inseparably associated with product development. In the academic literature, a number of theories have been formed to understand and describe the phenomena of engineering design, and the corresponding research area is known as engineering design research. This thesis performs engineering design research by developing methods that support engineering design.

2.2 Product Architecture

The word "architecture" is commonly used in connection with buildings or monuments to refer to the way they are constructed. For a product, architecture refers to the way it is constructed from its functional and physical elements (Ulrich, 1995; Ulrich and Eppinger, 1995). Functional elements are the individual operations and transformations that contribute to the overall performance of a product. For example, for a static aero-engine structure, located between the low-pressure (LP) and high-pressure (HP) compressors of a two-shaft engine, one of the functional elements will be to "transfer core flow between LP compressor and HP compressor."

The physical elements of a product are the parts, components and sub-assemblies that ultimately implement the product's functions. The physical elements are organized into several major building blocks called "chunks." Chunks are formed by the components of the product. Product architecture then, according to Ulrich and Eppinger (1995), is defined in terms of three particulars:

- The arrangement of functional elements
- The mapping from functional elements to physical components
- The specification of interfaces among physical components

Fujimoto (Fujimoto, 2007) defines a product-process architecture as the overall mapping that links together the core components of a product, such as its functions, physical elements, development process and interfaces among the physical elements. Product architecture is often used synonymously with system architecture when considering technical systems (technical system: a human implemented process or manufactured object, following the definition by Hubka and Eder (1996)). Jankovic and Eckert (2016) view system architecture as the structural arrangements of the parts of a product or components of a system so as to satisfy the functions required of it.

The various definitions of architecture converge to the manner of organizing and realizing different aspects associated with a product. The aspects can be physical, such as the parts of the product, or non-physical, such as the functions required of the product. In this sense, the architecture definition provided by Ulrich and Eppinger captures all the definitional characteristics: perhaps the reason for the longevity of the definition as it is frequently used in recent publications.

This thesis views product architecture as a means to create increased understanding about the product. In this respect, the mapping between the functional elements and physical components (or the specification of solutions to functions) is important. Therefore, the second particular from of Ulrich's definition is followed closely in this thesis.

2.2.1 Classification

Ulrich's definition classifies product architecture into modular and integral. In general, when one or a few of the functional elements are satisfied by a single chunk, the architecture is modular. The concept of modularity is widely used in the development of consumer goods, such as personal electronic devices or passenger cars. Modularity offers a number of benefits across the product development stages, from easily accommodating changed functionalities in the development phase to facilitating simpler testing in the final phase of development (Gershenson *et al.*, 2003). An aero-engine has modular architecture. For example, the GEnX engine by GE Aviation has two primary modules: the fan and the propulsor. The propulsor (the module that contains the compressor, combustor and turbine assemblies) can be detached from the fan module for independent maintenance (Donner, 2010). The architecture of the product is integral when one chunk satisfies more than one function or when a function is implemented by more than one chunk. Interactions among chunks are not precisely defined and can be incidental to the primary functions of the product (Ulrich and Eppinger, 1995). Integral architecture products are preferred when performance of the product is important.

Based on empirical studies in automotive and electronic industries, Fujimoto (Fujimoto, 2007; Park *et al.*, 2012) provides an additional dimension for architecture classification: as either open or closed. If the product interface specifications are made public, the architecture is open, and it is closed otherwise. When placed in an open–closed versus an integral–modular axis, aero-engines occupy the closed-modular axis, while engine structures occupy the closed-integral axis. Thus, two different architectures can exist within the same system, a fact also remarked by Jankovic and Eckert (2016). The dimensions for architecture classification according to Fujimoto are presented in Figure 4.

	integral	modular
open		• Personal computers • Trucks • Bicycles
closed	• Luxury cars • Aero-engine structures	• Aero-engines

Figure 4 Examples of product architecture classification along the open–closed versus integral–modular axis, based on the classification by Fujimoto (2007). Aero-engines are placed as closed-modular, as development is often performed within a closed consortium of companies.

The architecture classification by Jankovic and Eckert (2016) is based on system characteristics, such as degree of innovation on system level, degree of reuse on component or subsystem level, degree of integration with other products or degree of modification over the life cycle. An aero-engine and its structure will both belong to the two classifications of incremental design (the product is an improvement over previous versions) and the reuse of solution principles (products that use the same concepts of a previous product version). Incremental design is particularly applicable to functionally integrated products, since the development seldom begins from scratch. Otto *et al.* (2016) consider inputs to product architecture definition as function based and component based. For function-based inputs, an intermediary step is necessary to identify the correct components. For a component-based input, past experience is sufficient to go directly to the selection of components.

Product architecture need not be static. Citing the example of development of numerical controllers, Shibata *et al.* (2005) opines that product architecture may evolve from being integral first, to modular and then to open. For aero-engine components, the trend is to move toward integrality enabled by additive manufacturing technologies. Weight reduction and reduced inspection times due the elimination multiple manufacturing steps and, in some cases, savings in costs and lead time, are some of the driving factors behind the trend (Waller *et al.*, 2015). It is important for an aero-engine component developer to evaluate product architecture in light of various manufacturing possibilities so that no functions are adversely affected due to a newly adopted production alternative.

2.2.2 Significance

A number of decisions downstream of the product development cycle, such as the selection of suppliers, manufacturing operations, and delivery and service plans, are affected by the architecture chosen. Architectural knowledge is the means to understand, design and manage complex systems (Crawley *et al.*, 2004). Yassine and Wissmann (2007) hypothesize that a company's ability to efficiently and effectively manage a product portfolio is directly dependent on the product architecture. Product architecture plays a key role in how design organizations are structured, as described by Sosa *et al.* (2004). They propose a method to compare different architectures for the same product so that managers are aware of the changes in interaction among development departments due to the architecture changes. Architecture affects how changes are propagated through a system. Jung *et al.* (2018) provide a review of various means to assess change propagation and propose a metric to quantitatively assess inter-component dependencies within a product.

For modular architecture, the flexibility to meet changed requirements by recombining modules is the most important benefit articulated (Gershenson *et al.*, 2003). For integral architecture products, such recombination is not possible, and the entire product might need to change upon changes in requirements, which can be costly. Therefore, for manufacturers of integral architecture products, a thorough knowledge of product architecture is imperative, so that accurate decisions are made early in the development stage.

2.2.3 The Functionally Integrated Architecture

For a product with functionally integrated architecture, all functions are satisfied by one single monolithic component. The component may be fabricated from a number of segments with their own part numbers at the manufacturing firm, but when used by the system integrator, it is designated by one part number. Aero-engine structures, hulls of ships and automotive engine cylinder blocks are all typical examples of such products. Reasons for designing a product as integrated can be varied, from resource conservation (Whitney, 2002) to inheritance from the system in which it operates. Since the mapping from functions to physical components is not straightforward, Levandowski *et al.* (2014) term such products as functionally integrated. Functionally integrated architecture is unique, in that only a few firms in the world has such structures as their primary or only product offering.

The simultaneous implementation of all functions by a single structure is designated as function sharing by Ulrich and Seering (1990). Chakrabarti considers the structure as the entity being shared and calls the simultaneous implementation of all functions at the same time "structure sharing" (Chakrabarti, 2001; Chakrabarti and Singh, 2007). Functionally integrated products exhibit both function and structure sharing.

For integrated architecture products, no physical modularity exists, as there are no separable parts. During development, independent disciplinary considerations may be possible after suitable assumptions. For instance, it is possible to analyze deformations in the component independently of the effects from gas flow through the component after assuming negligible deformation effects due to gas flow pressure. Thus, even though a physical separation is not possible, separate disciplinary evaluations are possible for the structures, similar to products of any other architecture.

2.3 Architecture Modeling Approaches

From the definition of architecture in Section 2.2, a representation of product architecture can be made in the functional or physical domain, or as a mapping between the functional and physical domains. To represent architecture, a visualization of the mapping between functions and means is desired. In the following subsections, techniques for F-M modeling are discussed, followed by techniques that model and analyze dependencies between the functional and physical domain.

2.3.1 Function–Means Modeling

Several approaches to modeling functions exist, although building and using a functional model is not straightforward (Eckert, 2013). In engineering design, a lack of consensus exists regarding the definition of function (van Eck *et al.*, 2008). Ambiguity regarding the meaning of functions leads to difficulties in describing and modeling them (Crilly, 2013). Function can be used to mean the intended behavior of an entity, the desired effects of the behavior of an entity or the purpose for which the entity is designed (Eisenbart *et al.*, 2017). In general, a function can be specified in terms of a verb–noun combination, for example, "provide lubrication." For structures of integrated architecture, the functions do not change significantly over consecutive design iterations or across different designs. For example, irrespective of engine architecture, turbofan compressor structures must fulfill a certain set of functions, such as "transfer core flow." The regions that satisfy the functions (such as "vanes" or "flow walls") will also be present in all structures, although their geometries will be dependent on the engine architecture. The "means" are the ways in which the functions are satisfied. They are the organs in domain theory (Subsection 2.1.1). For an engine structure, the function "transfer core flow" can be satisfied by the means "flow path."

The subtract and operate procedure (SOP) (D. Lefever and Wood, 1996; Ullman, 2010) works by the successive removal and replacement of parts in a product. The effects of removing the part are documented, and from the effects, the function of the removed part is deduced. A second part is removed after replacing the first, and the procedure is repeated until all parts are removed and their functions determined. This technique associates functions to physical form. According to the domain theory (Hansen and Andreasen, 2002), designers think back and forth among and within the three domains (transformation, organ and parts), and the SOP technique facilitates this movement by concretely associating physical parts to functions.

In a function analysis diagram (Aurisicchio *et al.*, 2013), the functions of different parts, the type of connections among the parts, the flows through the parts and the transformation that occurs to the flows are directly shown above a layout of the actual product. A software implementation of the modeling scheme called DRed (Design Rationale Editor) is reported to be in use at Rolls-Royce plc.

An F-M tree (Hansen and Andreasen, 2002) models the function and the means (ways) to satisfy the function. A product is decomposed into multiple F-M levels. The F-M for a static engine structure can be represented as shown in Figure 5. The top-level function "transfer core flow" is satisfied by means of the a "flow path," which in turn has two subsidiary functions, "retain shape" and "offer least flow resistance," which are in turn satisfied by having a "rigid structure" and "streamlined shape," respectively.

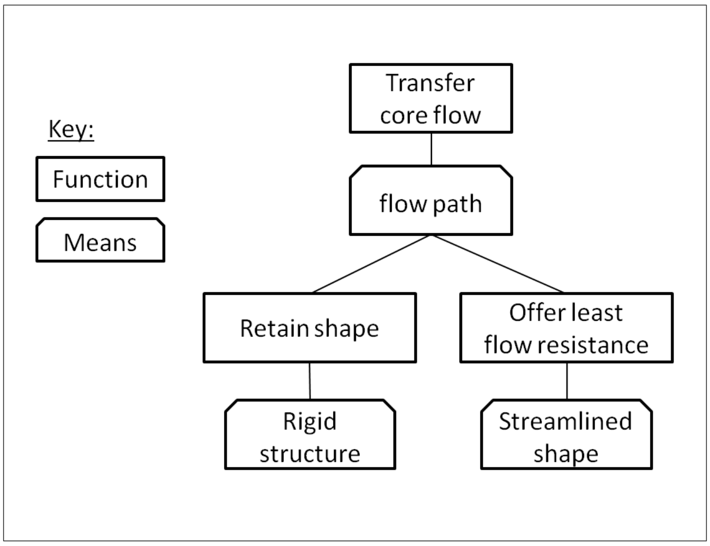


Figure 5 A Function–Means (F-M tree) for an engine structure

The extended F-M tree (Malmqvist, 1997) and the enhanced function–means (EF-M)

(Andersson *et al.*, 2000; Johannesson and Claesson, 2005) trees are improvements over the original F-M tree. The models incorporate constraints attached to the means as well as the relationships that exist among functions, means and constraints at different levels, using specific labels. This enables detailed information about the product's design be included in the F-M trees. Subsection 2.3.2 discusses the EF-M trees in relation to product platforms and configurable components.

2.3.2 Platform Modeling with Configurable Components

The definition of product architecture does not address how variants are created. Platforms address this issue (Kreimeyer, 2014). Robertson and Ulrich (1998) define platforms as a set of products sharing common assets. These assets are components (parts of the product), processes (manufacture or assembly equipment), knowledge (design know-how) and people (teams and relationships among teams). Platforms help a firm to meet the demands for product variety by the reuse of knowledge and technologies. The coordinated development for military engines, civil engines and industrial gas turbines by large engine manufacturers (Hongxia *et al.*, 2015) can be considered an example of a platform approach in aero-engine development. A shared development is possible because of the similarities in product architecture.

Approaches for platform modeling can be used for architecture representation. The configurable component (CC) method is a platform modeling approach developed by Claesson (2006). It is based on systems theory principles (Hitchins, 2003) and design theory (Andreasen, 1991; Hubka, 2013) and aims to model reusable platform elements. A CC is a model of a multi-functional product satisfying an arbitrary number of primary FRs. Each FR is the root to an EF-M tree (Johannesson and Claesson, 2005) branch in the product's design rationale (DR). The EF-M tree forms the basis of a CC description of a product platform by providing its DR. A DR with its EF-M trees can be seen as a formalized description of a specification of a technical system. In axiomatic design terminology, this description exists in the functional and physical domains. Architecture for a product can be modeled as its DR, using the EF-M tree.

Functional requirements in an EF-M tree are defined as what a product, or an element of a product, actively or passively shall do to contribute to a certain purpose, by creating internal or external effects. In this sense, the FRs motivate the downright existence of a specific solution. The means, organs or design solutions (DSs), are the to-be physical (for example, components or features) or non-physical (for example, service or software) entities that can possibly fulfill a specific FR (means are renamed from "design parameter" to "design solution," in part to make the word "parameter" free, usable in, for example, parameterized designs). The role of the non-functional requirements (referred to here as constraints, Cs) is to delimit the allowed design space for the FR-driven DSs. The relationships among FRs, DSs and Cs at various levels are denoted in Figure 6. The DR can be modeled in a software, called a configurable component modeler (COPE Sweden AB, 2014).

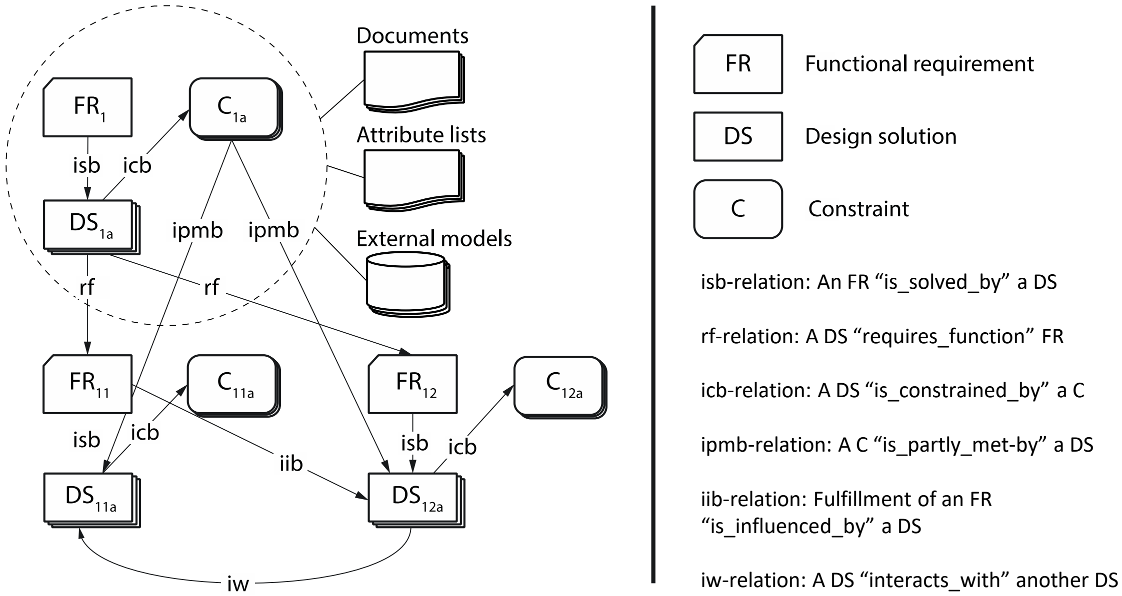


Figure 6 Enhanced function–means tree adapted from Johannesson and Claesson (2005)

The physical implementation of a DS (the various parts or components) is termed as a CO (component) in CC terminology. A CO can provide multiple DSs, unlike an FR, which is satisfied by only one DS. For example, if the means in Figure 5 (rigid structure and streamlined shape) are considered as DSs, both can be supplied by a static structure; that is, both DSs are provided for by a single CO.

EF-M trees can help to delineate a product's DR in detail, which makes them a potential candidate for architecture modeling.

2.3.3 Design Structure Matrix

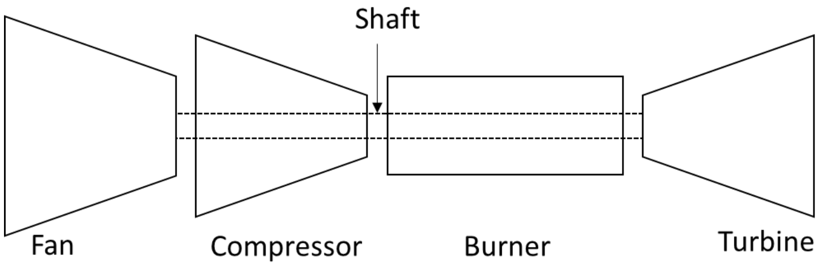
The design structure matrix (DSM) is a network modeling approach used to represent the elements of a system and their interaction (Eppinger and Browning, 2012). A DSM is a square matrix that shows the composition of different parts in a product (or components in a system) based on the type of interaction among the parts. The rows and columns of the matrix are formed by the parts of the product. In its simplest form, a DSM provides a binary matrix indicating the connectivity of parts in a product. If there are n parts in a product, the elements of a DSM, D is given as

$$D := (D_{ij})_{nxn}$$

$$\text{where } D_{ij} = \begin{cases} 1, & if\ i \neq j\ and\ parts\ i\ and\ j\ are\ connected \\ 0, & otherwise \end{cases} \tag{1}$$

For example, consider the schematic diagram of a single shaft aero-engine, shown in Figure 7(a). The DSM for the schematic engine can be constructed as shown in Figure 7(b).

a)



b)

	Fan	Compressor	Burner	Turbine	Shaft
Fan		0	0	0	1
Compressor	0		0	0	1
Burner	0	0		0	0
Turbine	0	0	0		1
Shaft	1	1	0	1	

Figure 7 Illustration of a Design Structure Matrix. a) Schematic of the single shaft aero-engine b) Mechanical-motion connectivity DSM.

It can be read from the DSM that the shaft engages the fan, compressor and turbine in motion by being mechanically connected to the components. The DSM captures the structural elements in a system and the scheme of the connection among the elements; that is, the architecture of the system.

The type of connection between different components might be standardized. This enables the simpler replacement of components and facilitates a modular system. DSMs are not only used on products but are also used to analyze processes and organizations. Browning (2016) prsents an extensive survey of the various applications of DSMs.

When it comes to functional integration, as discussed in Subsection 2.2.3, it is difficult to use a DSM, as there is only one component in the system. Literature that discusses usage of DSMs for monolithic components is not common. A sectional division of the component can facilitate the usage of DSM. The sectional division concept is used in this thesis.

2.3.4 Axiomatic Design Matrix

In axiomatic design (Suh, 1990), the mapping between the FR and DP (see Subsection 2.1.1) can be expressed in a matrix form. If **FR** is the vector of FRs with m elements, and **DP** is the vector of DPs with n elements, a design equation can be written as

$$\{\mathbf{FR}\} = [\mathbf{A}]\{\mathbf{DP}\} \tag{2}$$

In Equation (2), $[\mathbf{A}] = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$ is the design matrix (here, termed the ADM). Each element of **A** relates an FR to a DP. Taking $m = 3$, $n = 3$, and setting A_{ij} 0 or 1 based on the absence or existence respectively of FR – DP relations, three distinct types of design matrices can be exemplified, as in Table 1. The matrices indicate uncoupled, decoupled and coupled designs.

Table 1 the axiomatic deign matrix and the different design possibilities for axiomatic design based on the matrix

	DP1	DP2	DP2
FR1	1	0	0
FR2	0	1	0
FR3	0	0	1

a) Uncoupled design

	DP1	DP2	DP2
FR1	1	0	0
FR2	1	1	0
FR3	1	1	1

b) Decoupled design

	DP1	DP2	DP2
FR1	1	1	0
FR2	1	1	0
FR3	1	1	1

c) Coupled design

For an uncoupled design, each FR is satisfied by one and only one DP. No FR is influenced by any other DP. A change in FR can be met by changing only the concerned DP, and the design satisfies Axiom 1. For decoupled design (Table 1[b]), FRs are influenced by DPs other than their own. The functional independence can be maintained if DPs are perturbed from right to left. First, FR1 can be addressed by DP1. This will cause a change in FR2 that can be corrected by adjusting DP2, which in turn affects FR3, which is corrected by adjusting DP3. For a coupled design, even by managing the perturbing order, DPs cannot maintain functional independence. Such a design must be changed to a decoupled design by suitably changing FRs and DPs.

The simultaneous consideration of FRs and DPs is an advantage of an axiomatic design matrix (ADM) over a DSM (Dong, 2002). An ADM addresses the second particular in the definition of product architecture by Ulrich and Eppinger, while a DSM addresses the third (see Section 2.2). An ADM can be generated from an EF-M tree. Considering only the FRs and DSs at one level, the "isb" and "iib" relations ("is solved by" and "is influenced by" relations; see Subsection 2.3.2 and Figure 6) provide an ADM. The ADM generated from an EF-M tree is a potential candidate for architecture description once the FRs and DPs are identified.

2.3.5 Graphs and Graph Centralities

Another technique to model dependencies in a product structure is to use graphs. A graph G is a finite, non-empty set of objects called "vertices", together with a (possibly empty) set of unordered pairs of distinct vertices of G called "edges" (Chartrand and Lesniak, 1996).

The set of vertices for G shown in Figure 8 is

$$V = \{v1, v2, v3, v4, v5\},$$

and the set of edges is

$$E = \{(v1, v3), (v2, v4), (v2, v5), (v3, v5), (v4, v5)\} = \{e1, e2, e3, e4, e5\}.$$

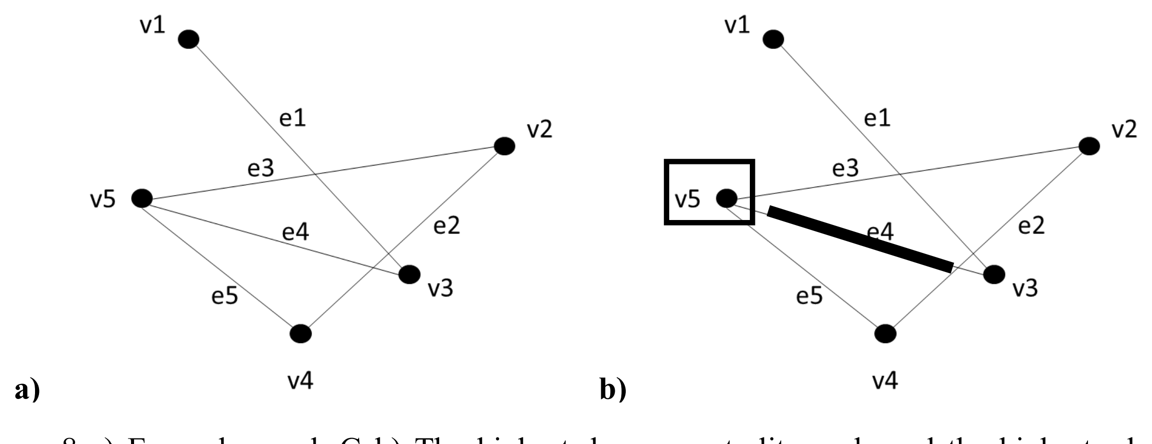


Figure 8 a) Example graph G b) The highest degree centrality node and the highest edge-betweenness centrality edge is highlighted in G

The vertices are also termed nodes. The matrix that illustrates the connectivity of nodes in a graph (whether the nodes are connected through an edge) is called an adjacency matrix. The adjacency matrix for a graph with n nodes is written as

$$A := (A_{ij})_{nxn}$$

$$\text{where } A_{ij} = \begin{cases} 1, & if\ node\ i\ is\ connected\ to\ node\ j\ through\ an\ edge \\ 0, & otherwise \end{cases} \quad (3)$$

When the nodes in a graph are of only one type (a so-called unipartite graph) and there are no edges that connect a node to itself, the adjacency matrix resembles a DSM. In this way, a graph can show the same information contained in a DSM. Compared to DSMs, Keller *et al.* (2006) note that graphs are better suited for reading information when they are small and sparse.

Once a graph is constructed, the underlying structure can be analyzed using its centralities. A central node and central edge of a graph are an important node and relationship, according to some criteria (Newman, 2010). The most important nodes can be identified based on the number of incident edges on that node. This is known as the degree centrality of that node. In Figure 8(a), the degree centrality for Node v5 is 3. For edges, a measure known as edge-betweenness centrality can be used to identify the most important relationships. A high edge-betweenness centrality indicates that the concerned edge must be traversed a number of times when tracing the shortest paths between any two pairs of nodes. A formal definition for the edge-betweenness centrality is given by Equation (4):

$$EB(e) = \sum_{v_i} \sum_{v_j} \frac{\sigma_{v_i v_j}(e)}{\sigma_{v_i v_j}} \quad (4)$$

In the equation, $EB(e)$ is the betweenness centrality of an edge e. $\sigma_{v_i v_j}$ is the number of shortest paths between edges v_i and v_j, and $\sigma_{v_i v_j}(e)$ is the number of shortest paths that passes through edge e (Wolfram Research Inc, 2015).

In Figure 8(a), the Edge $v5 - v3$ has a high edge-betweenness centrality, indicating that the relationship $v5 - v3$ is important to the structure of the graph. The high centrality node and edge is highlighted separately in Figure 8(b). Centralities, when found in a product structure graph, help to isolate the important parts and inter-part relations (e.g. connectivity) that influence the structure.

Graphs have been widely used in design contexts. An early use of graph theory was in design synthesis. Alexander (1964) uses graph theoretical concepts in the design of a rural village. In engineering design, the infused design approach (Shai and Reich, 2004) uses graphs as the means to share context-free information about the design problem among engineers of different disciplines. Several variants of graphs exist, such as signal flow graphs, which have directed edges, used to assess process effectiveness (Isaksson *et al.*, 2000). Graphs with directed edges are also termed networks. Wyatt *et al.* (2011) use a network structure, composed of the type of components and the type of connections among them in a product to model its architecture. In this thesis, graphs are used for architecture representation and the evaluation of functionally integrated structures.

2.3.6 Singular Value Decomposition

Singular value decomposition (SVD) is a technique to analyze the underlying structure of a matrix. The SVD for any matrix M is expressed as

$$M = U \Sigma V^T \quad (5)$$

In Equation (5), U and V are orthogonal matrices, and Σ is a diagonal matrix, the elements of which represent the singular values of M (Gershenfeld, 1999). Software tools such as MATLAB can directly create the decomposition and provide the singular values for a given matrix input. The singular values (elements of Σ) are sorted naturally, in descending order. When the singular values are plotted against their index, a pattern (called the singular value decay pattern) is revealed that is characteristic of the underlying matrix M, for which the values were generated. Since graphs can be expressed in terms of their adjacency matrices, the singular value decay patterns for a given graph can be determined, which can later be used for characterizing architecture. For example, Hölttä-Otto and de Weck (2007) and Sarkar *et al.* (2013) use SVD decay patterns to classify product architecture into integral, modular or bus-modular types.

2.4 Complexity in Engineering Design

Aero-engines are highly complex machines that require a multitude of technologies to design and manufacture. The components necessary for an engine are sourced from a range of suppliers, who are experts in a certain manufacturing technology who build parts according to specifications. From the traditional build-to-specification roles, partner firms participate in an RRSP, wherein they design, develop and produce key components for the engine (Johnston, 2017). The spread of development over several firms across national boundaries introduces several types of complexity, in domains as varied as design and public relations (Altfeld, 2016).

For a product, complexity can arise from internal (increased product variety) or external sources (Lindemann *et al.*, 2009). For engine structures, this can be interpreted as complexity due to the structure's design (internal) and requirements from the engine (external). In large-scale systems such as air traffic control and transmission grids, the complexity arises due to the connected nature of system. In contrast, the complexity of an aero-engine structure arises from its functional integration and manufacturing choices. Even though there is only a monolithic structure, understanding the complete structure and its functions is difficult.

The architecture of a product affects how complexity is viewed and managed. Architecture representations, such as function models or DSMs, are visual indication of a product's complexity.

2.4.1 Definitions

The concept of complexity is used in many fields, but it lacks a generally accepted definition (Chu *et al.*, 2003). According to Chambers 21st Century Dictionary, the primary characteristic of complexity is the quality of being complex, where a complex is *that which is composed of many inter-related parts*. A complex problem is one that is difficult to subdivide (Maurer *et al.*, 2014). Consequently, the problem cannot be solved easily by partitioning and allocating additional resources. Early definitions of complexity from computer science were based on the quantitative specification of information content (Kolmogorov, 1968). Complexity is taken as the minimal length of a program that will reveal information about object y with a given x, by taking x as its input. Complexity is often described in terms of various metrics based on the numbers and interactions of distinct components or functions of a system (Shah and Runger, 2013). Suh (2005) defines complexity as the uncertainty of achieving the FRs of a product. The more information is needed to achieve the requirement, the more the uncertainty in realizing it, and the more the complexity.

From the definitions, the difficulty to subdivide, having many parts and inter-relations, and information heaviness are all identified as characteristics of complexity. In this thesis, the complexity of aero-engine structures is viewed based on their physical arrangement and manner of satisfying the primary functions required of them.

2.4.2 Classification

In product development and engineering design, Weber (2005) considers complexity to exist in five dimensions: numerical (related to the number of components), relational/structural (relations and interdependencies among components), variational (number of variants of the product), disciplinary (number of disciplines involved in creating the product) and organizational (related to how the development organization is structured). The first three dimensions (numerical, relational, variational) may be readily associated with the physical characteristics of a product. The development process impacts the remaining two dimensions (disciplinary and organizational). ElMaraghy *et al.* (2012) classify complexity as existing in the functional and physical domains. Summers and Shah (2010) classify complexity in engineering design in terms of three metric categories: metrics developed for design problems (statement of objectives and requirements), design processes (steps taken to achieve the design problem objectives) and design products (the final result of carrying out the design processes). This can be viewed as an assessment of complexity during the different stages of product development. According to Bashir and Thomson (1999), complexity metrics can be inductive and deductive. An inductive metric is based on a large number of observations, while a deductive metric is developed to satisfy predefined criteria. A summary of the classification is provided in Table 2.

Table 2 Complexity classification

<table>
<tr><th colspan="5">Classification</th><th>Authors</th></tr>
<tr><td>Numerical</td><td>Variational</td><td>Relational</td><td>Disciplinary</td><td>Organizational</td><td>(Weber, 2005)</td></tr>
<tr><td colspan="2">Physical Domain</td><td colspan="3">Functional Domain</td><td>(ElMaraghy et al., 2012)</td></tr>
<tr><td colspan="2">Inductive (metrics)</td><td colspan="3">Deductive (metrics)</td><td>(Bashir and Thomson, 2001)</td></tr>
<tr><td>Design Problem (metrics)</td><td colspan="2">Design Process (metrics)</td><td colspan="2">Design Product (metrics)</td><td>(Summers and Shah, 2010)</td></tr>
</table>

2.4.3 Complexity Metrics

Since the information necessary for the development of complex products is large, the risk exists that problems can arise late in the design phase due to early misses in details. Besides, complex products are often regarded as expensive to develop. Understanding complexity in early design stages can help a developer with cost management and risk mitigation. To enable this, complexity metrics are useful.

The number of components and their interactions within a system are recurring features in definitions of complexity. Complexity is also regarded as a lack of information for fulfilling a task. Metrics of complexity are thus based on a system's network structure or information theory (Shah and Runger, 2013).

Ameri *et al.* (2008) propose two types of complexity for design products: size complexity (a measure of information content within product representation) and coupling complexity (a measure based on the coupling between the types of nodes in a bi-partite graph). Size and coupling complexities are evaluated based on three representations for the product: a parametric associativity graph (PAG) that associates different dimensional parameters to parts of a product, a function structure that indicates the interrelation of different functions of a product and a connectivity graph that represents the connections and the type of connections, such as press fit or snap fit among parts. The size complexity of the PAG is a straightforward measure to calculate product complexity.

Sinha and de Weck (Sinha and de Weck, 2016) propose a complexity metric that is validated based on theoretical criteria. The metric is composed of three separate constituents: (i) a components complexity $C1$, attributed to the complexity of individual components (ii) an interface complexity $C2$, attributed to the different types of interaction among system components and (iii) topological complexity $C3$, attributed to the specific layout of the system under consideration. The underlying system structure is represented using a DSM. Researchers have used the metric for the evaluation of both product platform complexity (Kim *et al.*, 2016) and the level of decomposition in a system (Min *et al.*, 2015).

The metric by Bashir and Thomson (2001) is based on the function structure of a product. Complexity is expressed in terms of the number of levels and the number of functions in each level. For a functionally integrated product, building the function structure is difficult, as a series of transformation functions is difficult to identify.

Many firms adopt modularization to manage complexity (Jiao *et al.*, 2007). When complexity is managed through product modularity, a number of metrics are available to assess it. A modularization function is introduced by Mikkola (2006); the function is dependent on the type of components, interfaces, degree of coupling, and substitutability. The values of the modularization function can vary between 0 and 1, and the higher the function value, the higher the modularity (or the lesser the complexity). Hölttä-Otto and de Weck (2007) propose the singular value modularity index (SMI) for assessing modularity. The index is based on singular value decay patterns for the respective product DSMs. Similar to the modularization function, SMI also varies between 0 and 1, and a larger value of SMI indicates higher modularity. Neither metric considers the interaction among components.

A shared characteristic of all metrics discussed thus far is their focus on multi-part products or multi-component systems. The application of such metrics has not been demonstrated on monolithic products such as aero-engine structures, and none of the metrics directly consider the two key aspects of the product's architecture: the load and flow paths (discussed in Section 2.5). The system-wide complexity metric by Sinha and de Weck, together with the size complexity metric by Ameri and Summers, may be suitable to establish a complexity metric for functionally integrated aero-engine structures.

2.5 Engine Structure Development Concepts

Some concepts of particular relevance to engine structure development are discussed in this section. These include the manner of describing the structures, the different options for the manufacture of the structure and the concepts of mechanical load paths and gas flow paths.

2.5.1 Sections and Manufacturing Options

The size of an engine structure can be very large, with diameters exceeding two meters. This makes it difficult to visually understand the complete structure. The different regions of the structure are often given names for easier identification. This thesis terms different regions of the structure as sections. Examples for the sections of the structure are shown in Figure 9.

Different structures can have different physical shapes for the sections, depending on the engine architecture.

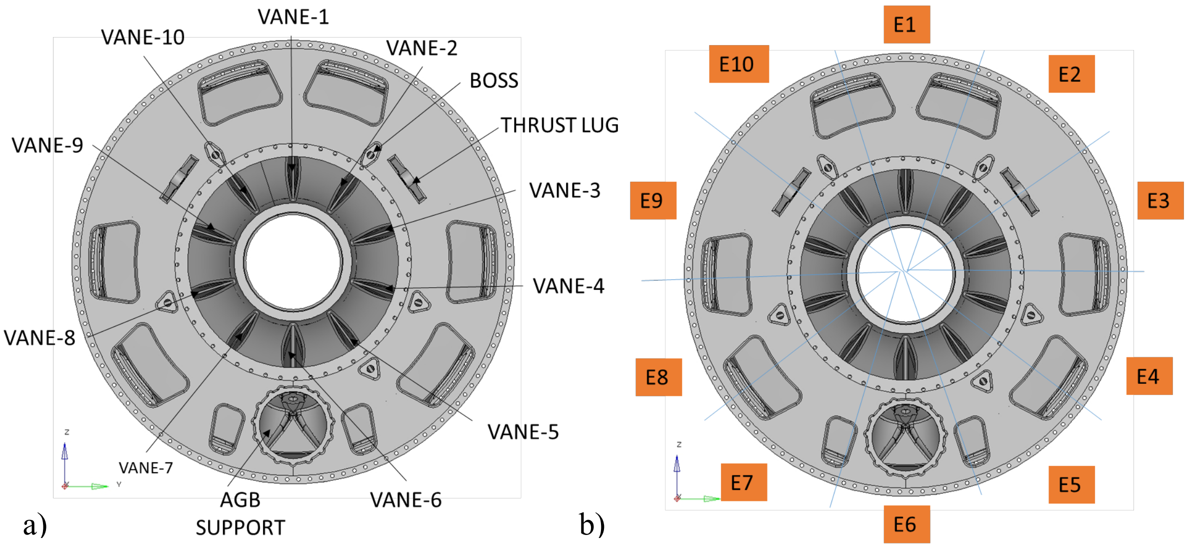


Figure 9 Sections (regions) and manufacturing options for an engine structure a) Different sections are marked on the structure b) The segmenting option with ten sectors (marked as E1 through E10)

The hot- and cold-structures are commonly manufactured, respectively, from super alloys, such as Inconel and titanium alloys (Ti-6Al-4V). Single-piece castings are the preferred manufacturing option, but as the size of the structures increases, they are split into segments and weld fabricated. One potential split for manufacturing a cold structure is shown in Figure 9. The different segments may be manufactured as cast, forged or be made from sheet metal.

The concept of sections, together with the segmenting option adopted for a product's manufacture, can be useful in describing the architecture of the product. A section can be considered identical to a DS in the EF-M modeling approach (see Subsection 2.3.2). The various segments for manufacture are what is physically produced of the structure. Therefore, the segments can be considered identical to COs in CC terminology. The division of the component into sections also enables describing the structure with a "sectional adjacency matrix," which is similar to creating a DSM for the structures, in terms of their sections.

2.5.2 Load Paths

The allowable weight for an engine structure is agreed upon with the OEM early in a development program. The manufacturer, developing and producing the engine structure may incur penalties if the weight is exceeded. Ensuring the mechanical strength of the structure while maintaining the weight is challenging, particularly when the loads on the structure are likely to change after each engine test. Knowledge of how loads are transferred through the structure or load paths is important to meet the mechanical strength and weight demands.

The visualization of the load transfer from the point of its application to the point of support in a structure is the load path (see Figure 10). Load paths help to arrange the structural elements in a product so that the applied loads are borne optimally. They are also useful to identify the change of load flow in case of failure, so that risk mitigation measures can pre-accommodated (such as fail-safe mechanisms). For simple products, structural engineers intuitively understand the load paths and may represent them as force flow diagrams (D. Lefever and Wood, 1996) or load lines (Pugh, 1977). For complicated products with multiple components or multiple load paths, intuitive understanding is difficult. Even though a number of approaches exist to identify and represent load paths, a commonly accepted approach does not exist (Marhadi and Venkataraman, 2009). Kelly *et al.* (2001) propose a method to plot the contours for which load in a certain coordinate direction remains constant, from the load application point to the support point. Another approach is to plot the contours of the strength of connection (called a stiffness index) between the loaded points and non-loaded points in the structure, which are representative of the load paths (Shinobu *et al.*, 1995; Naito *et al.*, 2011). The commercial implementation of both of these methods, which is necessary for industrial usage, is not easily available.

For finding the load path, this thesis uses an approach based on topology optimization (structural optimization), using a commercial software (OPTISTRUCT 2017) for finding the load path. In structural optimization, the minimization of compliance is commonly used as an objective. Compliance is a reciprocal measure of stiffness. Compliance minimization optimization seeks to maximize a structure's stiffness by varying individual elemental densities from zero to the actual material value, satisfying a certain lower mass limit for the design volume. The result is a series of hollow and material-filled regions within the design volume. If the objective is changed to minimize the mass of the structure so that it has a certain maximum compliance, all non-stiffness-essential material presence will be eliminated from the structure. The regions of material left in the structure will be indicative of the load path. This principle is utilized in the load path identification.

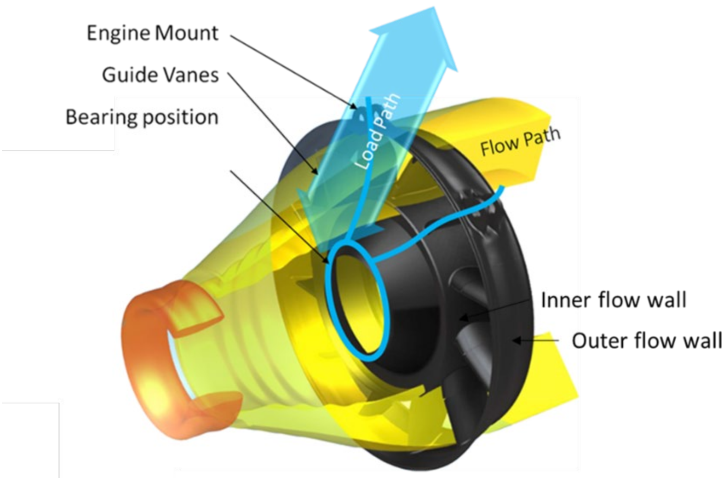


Figure 10 Illustration of the load path and the flow path. The load path is shown using blue lines and shading, while the flow path is illustrated using yellow shading. The load gets transferred from the bearing position to the engine mount through the vanes. The core flow passes through the opening formed by the guide vanes, inner flow wall and outer flow wall (the flow annulus).

A two-step procedure is used for identifying the load path of a structure. In Step 1, the compliance of the structure under a given loading scenario is determined. In Step 2, a mass minimization structural optimization is performed under the same loading scenario. A 5% higher value of compliance from Step 1 provides an upper limit constraint in the optimization. The 5% higher value is specified because this in effect causes a marginal relaxation in the required stiffness of the structure, which helps to initiate material removal from non-stiffness-essential regions. The design variables are elemental densities for all elements, except those at the loading and support regions. The optimization problem can be expressed as

$$
\begin{aligned}
\min_{\boldsymbol{\rho}} \quad & M(\boldsymbol{\rho}) \\
\text{subject to} \quad & L(\boldsymbol{\rho}) \leq L_1 \\
& 0 \leq \boldsymbol{\rho} \leq \boldsymbol{\rho}_{material}
\end{aligned} \tag{6}
$$

In Equation (6), $\boldsymbol{\rho} = [\rho_1, \rho_2, \rho_3, \ldots, \rho_k]$ is the vector of densities for each element, and k is the total number of elements included in the optimization. $M(\boldsymbol{\rho})$ and $L(\boldsymbol{\rho})$ are the mass and compliance of the structure, based on the elements included. $\boldsymbol{\rho}_{material}$ is the structure's material density, and L_1 is the compliance of the structure obtained from Step 1. After the optimization, the locations in the structure where the elemental density is more than 75% of the actual value for the material are noted. The locations will be representative of the load path in the structure for the considered loading scenario.

2.5.3 Flow Paths

The engine structures also transfer the core gas flow from one module to another. The inter-modular core flow (between the high- and low-pressure turbines, for example) is known as the primary flow. A small portion of the primary flow might be re-directed for various purposes which are then known as secondary flows. For example, for a structure located between the low- and high-pressure compressors, the primary flow is the compressed air flowing between the compressors. At the structure, some of this air might be extracted for pressurizing the flight cabin. The extracted air (also called bleed air) forms the secondary air flow. In turbofan engines, the primary air flow is axial, while the direction of the secondary air flow depends on the purpose for which it is utilized.

Similar to load paths, depending on the flow direction, flow paths can also be traced in the structure. In contrast to a load path, flow paths are traced through material-less regions. Flow paths are an important aspect of the structure's architecture and are carefully designed using CFD tools, together with physical testing. Figure 10 illustrates a flow path and load path on an engine structure.

2.6 System and Component Interactions

Component behavior in the system in which operates is vital information for any developer. The functioning of engine structures is influenced by both direct interactions, arising from interface connections (such as load transfer through bolted flanges), and non-direct interactions (such as a compressor operating at certain speed triggering resonance vibration in the structure). The information necessary to evaluate system–component interactions is generally the result of expensive physical testing, which is conducted only during later phases of a development project. In early design phases, such data is impractical, but at the same time, information about possible system–component interactions is needed, so that alternative product architectures can be evaluated for their effects. Computational methods are ideal for such early design evaluations. Better understanding of system–component interaction can be created by adopting a computer-based integrated development approach. A number of EU research projects (CRESCENDO, 2009; TOICA, 2013) have been undertaken for creating integrated system–component design methods.

For large-scale systems, the automatic generation and selection of solutions is needed. An example is a knowledge-based system or KBS (Dixon, 1995), which generates a number of solutions, evaluates the solutions based on predefined criteria, and in the case of failing to satisfy criteria, performs an automatic redesign guided by the evaluations. Reinman *et al.* (2012)

describe "design for variation," which uses various statistical techniques to improve the design of components (a turbine airfoil in the discussed case) at Pratt Whitney, in addition to performing multidisciplinary analyses. Sandberg *et al.* (2011) describe a study performed on rotating machinery using knowledge-based tools. Van Tooren *et al.* (2005) describe the design efforts for a freighter aircraft's composite wing using KBS tools. Jarrett *et al.* (2007) propose an approach to the integrated, multidisciplinary design of turbo machinery, such as an aero-engine's core compressor. The design is achieved by minimizing the difference between an ideal design and currently achievable design, and focus is given to coordinating efforts from different design teams. The general ideas under such approaches are knowledge capture, retention and reuse, which are collectively termed knowledge-based engineering (Verhagen *et al.*, 2012).

An overarching theme in such works is the coordination of design efforts across multiple disciplines, design teams and levels (e.g. overall system level or component level). Generally, software tools (e.g. HEEDS (Siemens PLM Software Inc, 2017); modeFRONTIER (ESTECO SpA, 2019)) are used to perform and manage the coordination. If the coordinated design problem is considered a complex system, then, as Eppinger (1997) notes, its complexity will be dependent on the pattern of interactions among system constituents. The disciplinary coordination is also the subject of multi-level and multidisciplinary optimization (MDO) frameworks. Well-founded mathematical formulation is necessary for the coordinating optimization problem (Martins and Lambe, 2013).

An MDO coordination framework of interest in this thesis is non-hierarchical analytical target cascading (Tosserams *et al.*, 2010). The approach works by splitting the multidisciplinary system optimization problem into smaller sub-problems. The sub-problems interact with each other through coupling variables called targets and responses. The responses of a sub-problem are functions of the responses from its neighboring problems. The coordination framework operates by attempting to minimize the difference between respective targets and responses within each sub-problem. The formulation can be expressed as

$$\begin{aligned} \min_{\bar{\mathbf{x}}_j} \quad & f_j(\bar{\mathbf{x}}_j) + \sum_{n \in R_j} \emptyset(\mathbf{t}_{nj} - \mathbf{r}_{jn}) + \sum_{m \in T_j} \emptyset(\mathbf{t}_{jm} - \mathbf{r}_{mj}) \\ \text{subject to} \quad & \boldsymbol{g}_j(\bar{\mathbf{x}}_j) \leq \mathbf{0} \\ & \boldsymbol{h}_j(\bar{\mathbf{x}}_j) = \mathbf{0} \\ \text{with} \quad & \mathbf{r}_{jn} = \mathbf{S}_{jn}\mathbf{a}_j(\mathbf{x}_j, \mathbf{t}_{jm} | m \in T_j), n \in R_j \\ & \bar{\mathbf{x}}_j = [\mathbf{x}_j, \mathbf{r}_{jn} | n \in R_j, \mathbf{t}_{jm} | m \in T_j] \end{aligned} \quad (7)$$

In Equation (7), $\bar{\mathbf{x}}_j$ are the optimization variables for sub-problem j, $\mathbf{r}_{jn}$are the response variables computed at sub-problem j, related to the targets $\mathbf{t}_{nj}$ computed at the neighboring sub-problem n, and $\mathbf{t}_{jm}$ are the target variables computed at sub-problem j, to be matched by the response variables $\mathbf{r}_{mj}$ at the neighboring sub-problem m. The function f_j is the objective function at sub-problem j, and $\boldsymbol{g}_j$ and $\boldsymbol{h}_j$ are the vector of inequality and equality constraints, respectively. $\mathbf{a}_j$ are the analyses required to compute the responses $\mathbf{r}_{jn}$, and $\mathbf{S}_{jn}$ is a binary selection matrix that selects components from $\mathbf{a}_j$ that are sent to sub-problem n. T_j is the set of neighbors for which the sub-problem j sets targets, and R_j is the set of neighbors to which the sub-problem j sends responses.

For engine structures, the identification of sub-problems is difficult, as there is only one monolithic structure (integrated architecture). A formal decomposition of the architecture using established techniques such as DSMs is necessary to utilize the approach on integrated architecture structures. A problem split based on the identification of load paths and flow paths can facilitate the application of the coordination framework.

3 Research Approach

This chapter details the research approach adopted in this thesis to conduct and validate the research. The research approach is introduced in general and is linked to engineering design research. The research methodologies and validation approaches adopted in this thesis are presented. The chapter concludes by detailing the research progress in terms of the methodologies.

Research approaches are the plans and procedures involved in research, stretching from the detailed methods for data collection, analysis and interpretation to the general assumptions involved (Creswell, 2014). The world view of the researcher, the research field, the methods the researcher intends to use and various designs for research, such as qualitative or quantitative, influence the research approach. The research approach will be reflected in the methodologies adopted.

This thesis concerns methods to assist the design of aero-engine structures. The research area is engineering design, and the approach is informed by the world views in design research. Reich (2010) presents two world views for design research: a correspondence stance that regards the value of research in its utilization and a coherence stance that regards the value of research in terms of its theoretical rigor. The present research is conducted industrially, and the results are for industrial usage. The practical usability of the developed methods is of interest, and the thesis adopts the correspondence stance. The methodological framework for research and validation are influenced by the correspondence point of view.

3.1 Methodologies for Design Research

According to Reich (1995), the study of research methodology is important, as it lets the researchers form a repository of methods with their assumptions, interpretations, successes and failures. The work in this thesis was conducted based on two methodologies which are described in this section.

3.1.1 Jørgensen's Model

Research where the practical applicability of results is important is called applied research (Buur, 1990). Jørgensen's paradigms for research and development (Jørgensen, 1992), depicted in Figure 11, form a descriptive model of applied research. Research begins either from a problem base (an industrial opportunity) or a theory base (a scientific knowledge gap) and proceeds to generate scientific insights by continual analysis and synthesis. The scientific insights from the research phase are utilized in the development phase for practical applications. A similar idea is presented by Horvath (2001), while introducing a reasoning model to form different knowledge categories within engineering design research. Knowledge is transferred from scientific or theoretical enquiry to technical or practical application.

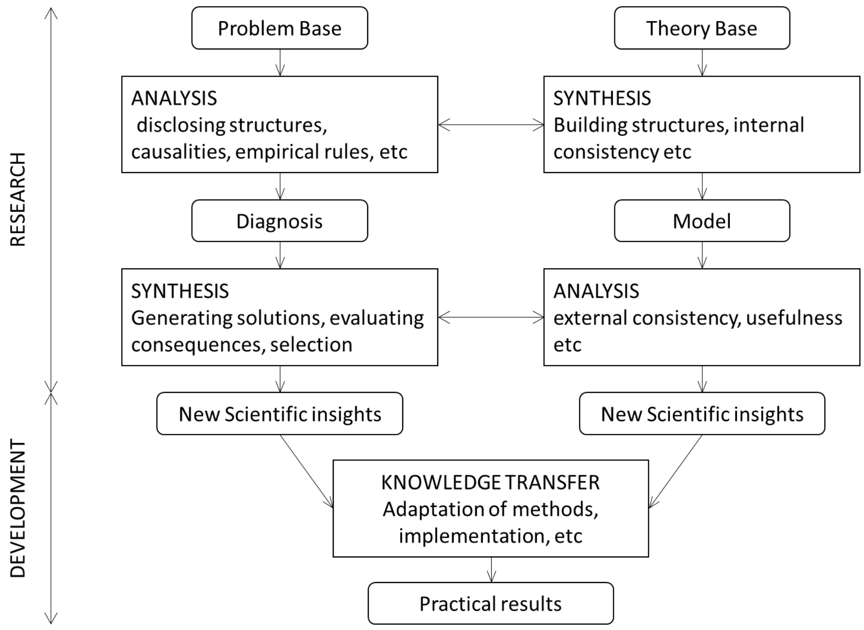


Figure 11 Work paradigms for research and development activities according to Jørgensen, following the translation from Danish by Michaelis (2013).

3.1.2 The Design Research Methodology

The design research methodology (DRM) proposed by Blessing and Chakrabarti was specifically developed to support engineering design research. The methodology enables a phased research execution, similar to phased product development in the industry. Iterations are allowed between the stages in DRM, while the stage-gate approach aims at producing a final product without inter-stage iterations. DRM defines four phases for a research project: research clarification, descriptive study I, prescriptive study and descriptive study II. The phases, means for, and outcomes from each phase are shown in Figure 12.

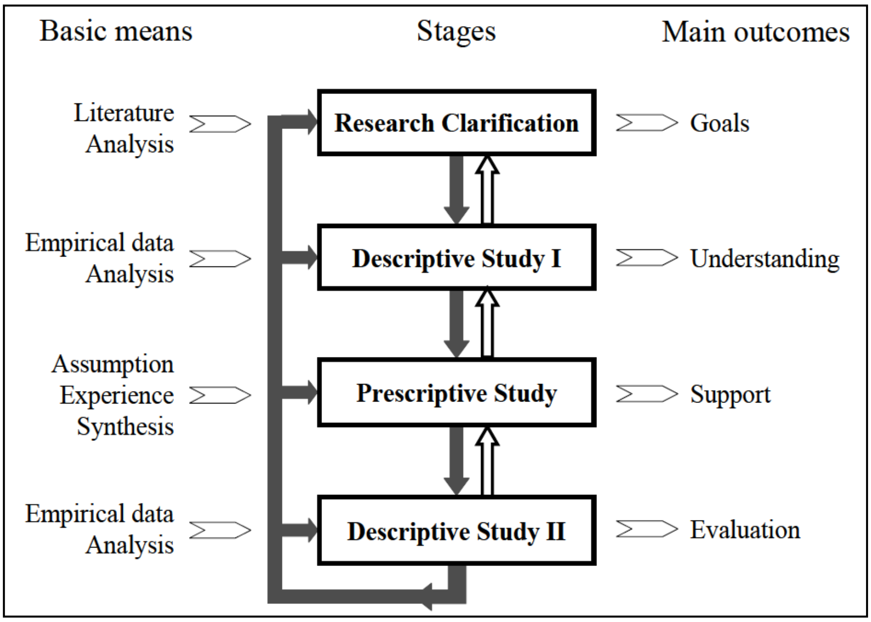


Figure 12 DRM phases from Blessing and Chakrabarti (2009)

During the "research clarification" phase, the goals of the research project are formulated. The present and desired situation are described, and criteria are defined, which will help to indicate how much the present situation has moved toward the desired situation.

During "descriptive study I", increased understanding of the present situation is created through empirical studies, which may involve literature review. This phase identifies a number of factors that influence the existing situation.

During "prescriptive study", the increased understanding of the present situation is used to improve the description of the desired situation. By varying one or several of the factors identified in the descriptive study, a support or a preliminary suggestion to move toward the desired situation is suggested, ensuring that the support is implemented correctly.

"Descriptive study II" focuses on studying the impacts of the proposed support during the prescriptive study. Empirical studies are used to calculate the criteria defined in the research clarification phase to assess whether the support has improved the present situation.

Design research methodology identifies seven types of design research, as shown in Figure 13. In the figure, the review-based study is based only on the literature. A comprehensive study is an empirical study, wherein the researcher develops and evaluates support. Literature review may also be included in the study. In an initial study, the first few steps of any of the stages are performed, where the objective is to show the consequence of the results.

Research Clarification	Descriptive Study I	Prescriptive Study	Descriptive Study II
1. Review-based →	Comprehensive		
2. Review-based →	Comprehensive →	Initial	
3. Review-based →	Review-based →	Comprehensive →	Initial
4. Review-based →	Review-based →	Review-based → Initial/ Comprehensive	Comprehensive
5. Review-based →	Comprehensive →	Comprehensive →	Initial
6. Review-based →	Review-based →	Comprehensive →	Comprehensive
7. Review-based →	Comprehensive →	Comprehensive →	Comprehensive

Figure 13 Types of research according to DRM. This thesis falls into Type 5, as highlighted (see Subsection 3.3.2).

3.2 Research Quality

Design research has elements of both qualitative and quantitative research. Ensuring research quality involves validating both the qualitative and quantitative aspects of the developed methods. In engineering research, validating design methods involves building confidence in the methods with respect to a purpose (Seepersad *et al.*, 2006). Since the research is conducted primarily within industrial settings in cooperation with academia, in addition to ensuring scientific quality, the suitability of the methods for industrial application must be substantiated.

For scientific validation, this thesis adopts the validation square approach proposed by Pedersen *et al.* (2000). For industrial validation, judgments about the technology readiness level (Sadin *et al.*, 1989) for the methods are made.

3.2.1 Scientific Confidence: The Validation Square

Pedersen *et al.* (2000) propose a research validation approach called the validation square, which was introduced due to the multidisciplinary (quantitative and qualitative) nature of design research. It examines both the qualitative and quantitative aspects of the research by examining the theoretical and empirical acceptability of the work. The validation square, adapted from Pedersen *et al.* (2000), is presented in Figure 14.

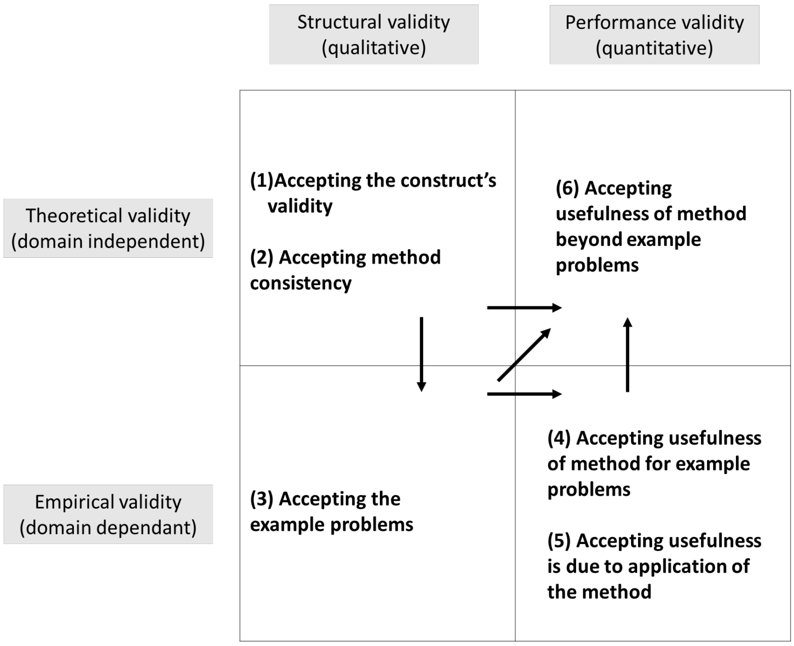


Figure 14 The Validation square, adapted from Pedersen *et al.* (2000)

When each of the acceptance checks are performed and found to be true, the research work's validity is proven. Each acceptance check is described briefly below:

(1) Accepting the construct's validity: During this check, the logical consistency of the proposed design method is evaluated. If the method is composed of a number of constructs, the individual logical consistency of each construct should be established.

(2) Accepting the method's consistency: After establishing construct validity, the internal consistency of the proposed method in its entirety is established. This check builds confidence in how the method is constructed from the individual constructs.

(3) Accepting the example problems: Once the consistencies for the individual constructs and overall method are established, the suitability of the example problems used to validate the method is ascertained.

(4) Accepting usefulness of the method for example problems: The design method is applied to the example problems considered, and the usefulness of the method is established. For instance, in an industrial context, this can occur by noting how much time or cost is reduced due to the proposed method.

(5) Accepting that the usefulness is due to the application of the method: In this step, whether the usefulness exhibited in the example problems is actually due to the application of the method is proven. The proposed design method can be compared with existing practices or

another method to prove the significance of using only the proposed method.

(6) Accepting usefulness beyond example problems: If Steps 1–5 are accepted, it can be stated by induction that the method is applicable to problems having precisely the same characteristics as the example problems.

The validity of the design method must be examined beginning at the 1–2 square and proceeding to the other squares as applicable, as the arrows indicate in Figure 14.

The approach shares similarities with coupon testing in material science. Test coupons made using a certain material according to predefined specifications are tested under different conditions. The test results are used to argue for the integrity of products that use the same material, provided the operating conditions can be considered similar to the test conditions. The validation square can also be considered a research methodology in itself. Within systems engineering research, Muller (2013) presents similar ideas, wherein validating research is the manner in which the research itself is conducted.

The results from this thesis are a set of methods to assist the design of aero-engine static structures. The validation square technique is appropriate to systematically go through the development of methods and validate them. The application of the validation square technique to the methods developed in this thesis is detailed in Subsection 5.2.1.

3.2.2 Industrial Confidence: Technology Readiness Levels

A technology readiness level is a metric that specifies the maturity of a certain technology (Mankins, 1995), such as a manufacturing method. The metric indicates whether the technology is in the conceptual stage or has been validated by tests in real-life environments. The concept of technology readiness levels (TRL) was initiated at NASA for evaluating research technologies so that they could be transferred to and used in actual programs (Sadin *et al.*, 1989). Safe alternatives are preferred over insufficiently demonstrated technology, as the "lure of high risk advances is not often offset by the price of expensive mission failures." A number of organizations around the world have adopted TRLs into their R&D management practices (Mankins, 2009). In GKN Aerospace, a new design method is introduced only after passing a TRL 6 review. For the methods developed in this thesis, formal TRL reviews are yet to be conducted, and the presented scales in Subsection 5.2.2 are judgments based on the descriptions by Mankins (2009). Table 3 presents the nine TRL levels, with descriptions.

Table 3 Technology Readiness Levels (Mankins, 1995) with descriptions. The more mature the technology, the higher the TRL.

TRLs	Description
1	Basic principles observed and reported
2	Technology concept and/or application formulated
3	Analytical and experimental critical function and/or characteristic proof-of-concept
4	Component and/or breadboard validation in “laboratory” environment
5	Component and/or breadboard validation in relevant environment
6	System/subsystem model or prototype demonstration in a relevant environment
7	System prototype demonstration in the planned operational environment
8	Actual system completed and “qualified” through test & demonstration (in the operational environment)
9	Actual system “proven” through successful system and/or mission operation

3.3 Research Approach in the Present Thesis

Jørgensen’s model is the overall guiding framework for the thesis. The DRM has been used to guide the progression of research. The division of research into phases also allows results communication by locating the research publications in different phases. In this research, the supports were developed after comprehensive descriptive phase, while the evaluation of support is only preliminary. The usage of methods in actual development programs has not been performed, even though they have been demonstrated on real-life engine structures. Therefore, the research falls under Type 5, as noted in Figure 13.

3.3.1 Research Data

The research work was performed at the engineering methods department at GKN, in routine interactions with research engineers. The various assumptions and results were discussed with specialist engineers working with structures in both the research and engineering development departments. Working at an industrial site has benefited the collection of relevant and realistic data, such as the CAD geometries of real-life engine components.

The data used in this research is empirical and varied. It includes detailed CAD geometries, functional information and FE boundary conditions. In Table 4, the type of data used in this research is listed, with sources.

Table 4 Types of data and source

Type of Data	Source
Structures operational data (pressure drop)	GKN data based on in-house simulation
Detailed CAD geometries	GKN in-house research models
Simplified CAD geometries	Simplifications based on detailed CAD geometries
Simplified FE geometries	Simplifications based on detailed CAD geometries
Detailed FE geometries	Based on the detailed CAD geometries
Function data	FMEA sheets for products of the same class; unstructured interviews
Sections data	Existing naming practice within GKN; unstructured interviews
FE loading and boundary conditions	Simplified based on typical load sheets for the structures

3.3.2 Research Progression

This section details the progression of the research in terms of the formation of the publications.

The necessity to understand the effects of the structures on engine operation (RQ1) prompted Paper B. A method was formulated to couple an operational parameter (pressure drop caused by the structure in the core gas flow) of a turbine rear structure to engine operation. The exercise showed that designing a turbine rear structure (TRS) well has significant beneficial effects on engine operation. The work was performed in close collaboration with the turbo machinery research group at the Fluid Dynamics division at Chalmers.

Tasks aimed to understand the engine structures' architecture (RQ2) led to Papers A and C. This work progressed simultaneously with RQ1. In Paper A, the EF-M tree was used for the design decomposition of an existing compressor structure. The objective was to systematically identify the regions and functions associated with the regions in the structure. Initial ideas about functional integration were formed after this publication, which led to the research in Paper C. In Paper C, approaches to modeling the architecture of the product and utilizing the model to derive physical conclusions were explored. Graphs were used as the means to model the structure, and the properties of the underlying matrices of the graphs were used to propose physical conclusions. In addition, the publication led to the realization of two important insights, that the architecture of the product is heavily influenced by the manner of splitting the product and that the architecture is determined by the two primary functions of the product: carrying load and transferring flow. This led to forming RQ3 and RQ4 regarding the utilization of architectural insights.

To investigate the practical applicability of the research so far, the load path aspect of the architecture was utilized in Paper D in an MDO context. Even though the flow path aspect of the architecture was not included in the study, the adaptation of the MDO framework for architecture selection is significant. Paper E demonstrated the architecture modeling approach in Paper C on a complete engine structure. It also confirmed the need for a quantitative metric to evaluate and classify different structures, as expressed in RQ4.

The question to form a quantitative metric for the architecture (RQ4) prompted studies in Paper F, wherein a complexity metric for the structures considering the structural aspects of architecture was developed.

The contributions of the papers to answering research questions and the research progression in terms of DRM stages are presented in Figure 15.

		RQ 1					
		RQ 2					
					RQ 3		
						RQ 4	
		Paper A	Paper B	Paper C	Paper D	Paper E	Paper F
		RQ 2	RQ 1	RQ 2	RQ 3	RQ 3 RQ 4	RQ 4
DRM Stages	RC	●	●	●	●		●
	DS-I	●		●	●	●	●
	PS	●	●	●	●	●	●
	DS-II					●	●
		TIME →					

Figure 15 Research progress in terms of RQs and publications, presented across DRM stages. Dots indicate that the works in the publication in the concerned column include those specified by the DRM stage in the concerned row. The duration of RQs and the contribution of publications for answering the RQs are also shown in the figure.

4 Results and Appended Paper Summaries

This chapter provides the summaries of the appended publications, along with the findings.

4.1 Paper A: Function–Means for Engine Structures

Generic Functional Decomposition of an Integrated Jet Engine Mechanical Subsystem Using A Configurable Component Approach

4.1.1 Objective

Assessing the functional impact of a design change on a structure of integrated architecture is difficult. It is necessary to identify the functions that the structure satisfies and the physical regions of the structure that contributes to the functional satisfaction; that is, an F-M identification is necessary. For modular products, F-M identification is relatively easy, but a systematic procedure is needed for functionally integrated products. The work presented in the paper provides a method for F-M extraction for an aero-engine structure.

4.1.2 Approach

A procedure is proposed to systematically identify and associate the functions of the structure, the sections in the structure and the physical constraints associated with each section. Sections are identifiable regions in the structure that can be thought of as satisfying the functions required by the product. Figure 16 illustrates the sections on an engine structure. The sections have constraints associated with them. The resulting function–section-constraint information, after the application of the procedure is tabulated and converted into an EF-M tree. In the EF-M tree, each F-M-constraint construct is encapsulated in a CC. The CCs are in turn served by physical components, or COs (refer to Subsection 2.3.2; not to be confused with the structural component). The CCs can be reused and instantiated in new designs or used for design improvements.

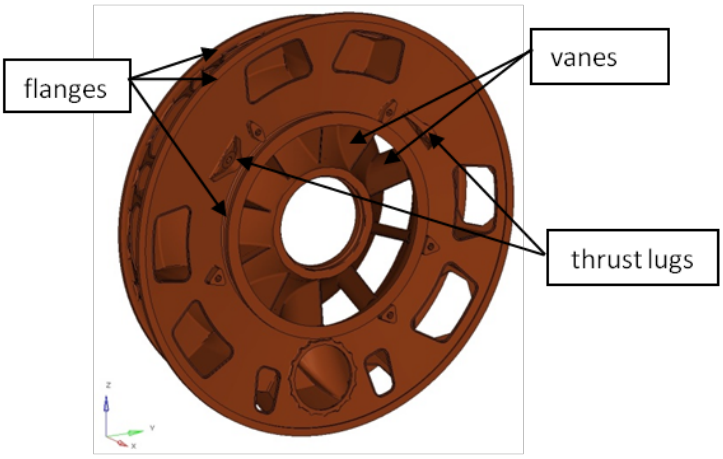


Figure 16 A functionally integrated engine structure with some of its sections marked

4.1.3 Results and Findings

The proposed method systematically identifies the sections in the component and the associated functions with each section. Since the procedure is bottom up, it facilitates beginning the F-M identification from familiar premises (the sections of the structure are well known in the firm) and is intuitive for the development engineers. At least two applications exist for the stored F-M information, of which the first is design documentation. Once the EF-M tree is made, the contribution of different sections toward satisfying the FRs can be easily visualized. The second application is the help provided in generating new designs. In Figure 17(a), the CO "vane" realizes 3 CCs. If the vane is made entirely from sheet metal, the structure may not be able to perform the function "transfer loads towards engine outer frame" effectively. In this case, the CC for load transfer might need to be realized separately from CCs. Another CO, "struts", can satisfy the load transfer function. The generation of new solutions is shown in Figure 17(b).

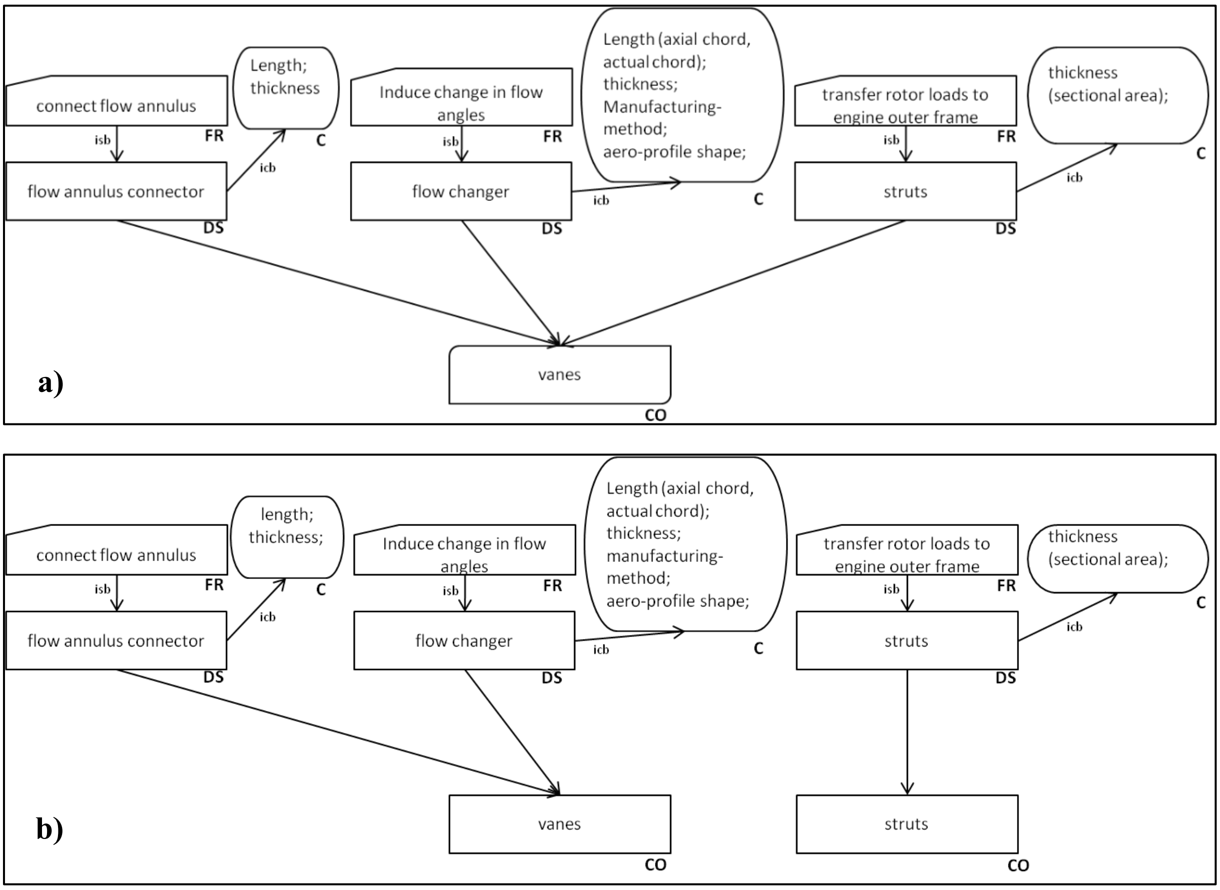


Figure 17 EF-M tree associated with the "vane" section. The markings "isb" and "icb" denote "is solved by" and "is constrained by" relations in the configurable component terminology (see Subsection 2.3.2). **a)** CC for Vane displaying three CCs and the Vane CO **b)** Separation of function from a section and formation of new design

The following findings can be highlighted from the paper:

- The concept of sections is important in dividing and describing the structure. The sections concept allows for an easier application of architecture description techniques.
- Once the sections are identified, the CC methodology and EF-M trees can be used for representing the architecture of functionally integrated products.

4.2 Paper B: System Influence of Engine Structures

Exploring Influence of Static Engine Component Design Variables on System Level Performance

4.2.1 Objective

The objective of this work was to assess the effect of static aero-engine structure design on engine operation. A coupled evaluation between two design disciplines, related to the operations of the engine and the component operation, is adopted to assess the impact. The disciplines are the thermodynamic performance calculations for the engine (generally performed at the OEM) and the structure's detail design calculations (performed at the component developer). If information about influence of the structure's design on system operation is available, the structure can be made more robust to system (engine) level changes.

4.2.2 Approach

A TRS located immediately at the aft of the low-pressure turbine in a turbofan engine is considered for evaluation. The drop in pressure when the engine core flow passes through the TRS and the weight of the TRS both influence engine operation. To estimate a preliminary measure of the effect of TRS design at the engine level, pressure drop calculations performed on different variants of an aerodynamically well-designed (the TRS inflow geometry matches the outflow characteristics from the final turbine stage) TRS are used. The maximum and minimum pressure drops are used to calculate the efficiency for the low-pressure turbine. The turbine efficiency affects the SFC of the engine. Figure 18 displays the range of pressure drops for the TRS design variants and the disciplinary coupling.

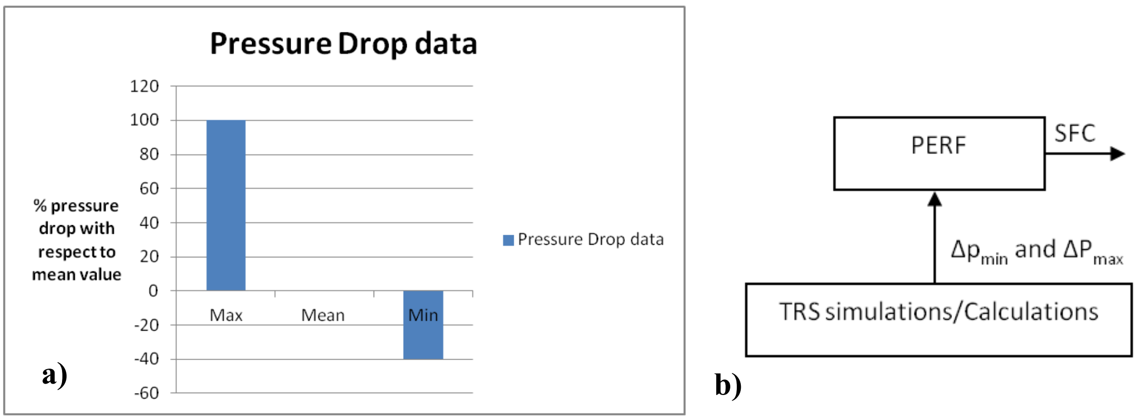


Figure 18 Approach to estimate system influence of engine structures. a) Maximum and minimum percentage pressure drop in engine core flow through the TRS for different design variants b) Disciplinary coupling for the estimation of the structure's system dependency. PERF refers to engine performance calculations.

Similar to the aerodynamically well-designed TRS, the pressure drop for an aerodynamically poorly designed TRS (the TRS inflow geometry does not match the outflow characteristics from the final turbine stage) is calculated, and the corresponding variation in SFC is found. Thus, the engine's SFC variation, corresponding to aerodynamically well-designed and aerodynamically poorly designed TRSs, can be estimated.

4.2.3 Results and Findings

The pressure-drop-dependent SFC variation for the variants of the aerodynamically well-designed TRS is 0.06%. The difference in SFC between aerodynamically well- and poorly designed TRSs is 0.9%. Even though the percentage variation in SFC is small, it can result in significant savings for an airliner when calculated over a year, for the entire flight operations. Thus, even small improvements in engine operation are a significant incentive for the careful design of engine structures.

The following findings can be highlighted from the paper:

- Estimates of the engine structure's system-level influence can be obtained from coupled simulations. The estimates, despite being preliminary, can provide an overall understanding of the effects of the structure's design on engine operation and can direct design efforts.
- For the multidisciplinary coupled evaluations, coordination among different disciplines is important.
- Due to the structure's influence on system-level operation, creating better understanding about its design is of value to the developer.

4.3 Paper C: A Product Architecture Description Method

Describing and Evaluating Functionally Integrated and Manufacturing Restricted Products

4.3.1 Objective

This work presents a method for representing and evaluating the product architecture of functionally integrated products, such as aero-engine structures. In conventional DSM or node link diagram-based architecture modeling, different parts of a product are represented in matrix or graph form, respectively. For an integrated architecture structure, the terms "part," "component" and "product" share the same meaning. Only one row and column or a single node will be present if a matrix or a graph is used to represent the architecture. Thus, an improved scheme is necessary to represent the architecture for functionally integrated products.

4.3.2 Approach

Engine structures are often weld fabricated from a number of segments. The manner of splitting the structure into segments for fabrication affects the architecture. An architecture description should incorporate the manner in which the structure is split for manufacture. Two methods for describing and evaluating the architecture are proposed in the paper. The first is based on creating sets of FRs for the structure, sections of the structure and segments for fabricating the structure and establishing relations among the sets. The functions and sections are considered generic for a class of products and are termed generic functions (GF) and generic sections (GS). Let $\mathtt{GF}$, $\mathtt{GS}$ and $\mathtt{M}$ represent the sets for generic functions, sections and manufactured segments. Relationships can be established between the set of functions and manufactured segments as well as the set of segments and sections. A composition of these two relations will provide a relation between the set of functions and the set of sections, based on the manufacturing split. The composition of relations can be expressed as

$$f1 : GF \rightarrow M$$

$$f2 : M \rightarrow GS$$

$$f2 \text{ o } f1 : GF \rightarrow GS$$

The relation $GF \rightarrow GS$ is representative of the architecture of the structures, taking into account the manner of splitting them for fabrication.

The second method of describing the architecture is based on creating an EF-M tree. The functions in GF are directly associated to the sections in GS. In this, the GSs are considered identical to the DSs in an EF-M tree. Similarly, the segments in M are considered identical to COs in the configurable component terminology. The consideration of sections as DSs and manufactured segments as COs helps to manually identify the cross influences of functions and sections according to the type of split chosen. Thus, different manufacturing segmenting options result in different EF-M trees.

Once the two architecture representations are created, they are evaluated in two ways. Both methods of evaluation involve the creation of graphs (Subsection 2.3.5) from the representations. The evaluation examines the singular value decay pattern (Subsection 2.3.6) for the adjacency matrices from the graphs. A gradual fall in the decay pattern indicates modularity, while a sharp fall indicates integrality (Hölttä-Otto and de Weck (2007). Based on the pattern, insights about the function–section groupings in the structure can be ascertained. The second method of evaluation is based on finding centralities of nodes and edges (Subsection 2.3.5) in the created graph. Centralities are a means of identifying important nodes and relationships in a graph with respect to some criteria. Here, the nodes are the functions and sections of the structure and edges are the relationships between functions and sections. A summary of the approach is provided in Table 5.

Table 5 Summary of architecture representation and evaluation schemes in Paper C.

	Representation	**Conversion Form**		**Evaluation**
Scheme 1	Set theory based	Graphs	**Scheme 1**	Graph centralities
Scheme 2	EF-M tree based		**Scheme 2**	SVD decay pattern

4.3.3 Results and Findings

The method for architecture description and evaluation was applied to a pump casing as well as a vane structure. The results from the application to a pump casing, assumed to be manufactured from a four-piece casting, are shown in Figure 19.

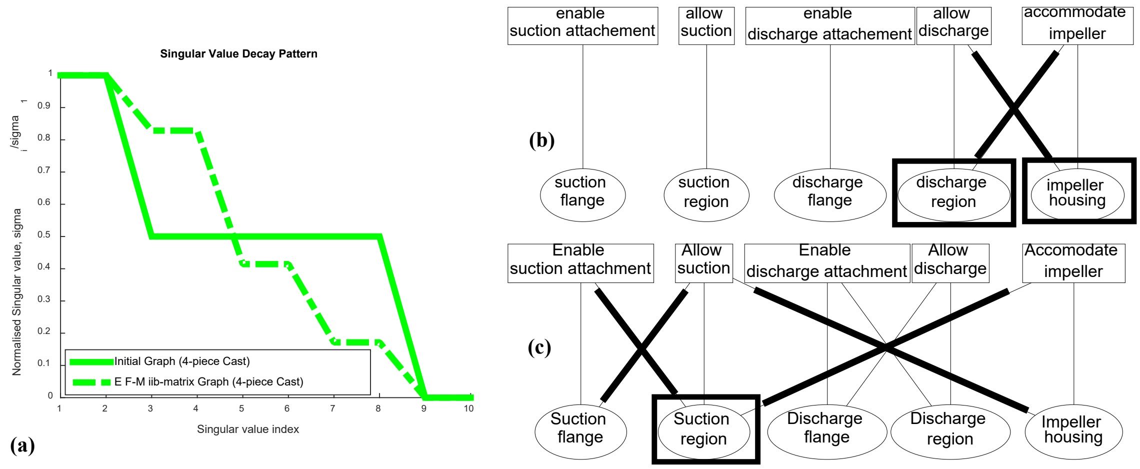


Figure 19 Results from the architecture description method. Functions are denoted in rectangles, and sections are denoted in ovals. (a) SVD decay pattern of the graph descriptions for a four-piece cast pump casing, using set theory and EF-M-tree-based architecture description approach (b) Graph from the set theoretical architecture description with nodes and cross edges with high centralities highlighted. (c) Graph from EF-M tree description with nodes and cross edges with high centralities highlighted

Insights about the architecture of the structure can be derived from the analysis results. For example, in Figure 19, the SVD decay pattern for the pump casing based on EF-M trees graphs is more gradual (more steps) compared to the pattern for the set theoretical graphs. This indicates that more function–section groupings (one section satisfying several functions, or one function served by several sections) are present in the structure when relations are manually identified in the EF-M tree. A shortcoming of this analysis approach is the inability to isolate from the pattern exactly which functions or sections are grouped.

Using the evaluation approach using graph centralities, important sections and function–section cross relationships can be identified for the structure. In the graph for the set theoretical description, two sections (discharge region, impeller housing) are identified as important (as they are connected to the highest number of functions), but in a later EF-M tree graph, only one section is identified as important (suction region).

The following findings can be highlighted from the paper:

- For functionally integrated products, architecture descriptions must incorporate the manufacturing options.
- EF-M-tree-based architecture description is superior in identifying important sections and function–section cross relationships than the initial set theoretical scheme. The set theoretical scheme is in contrast quicker and more intuitive for engineers (since it does not demand knowledge of the CC method) and can provide an overview of the architecture, based on how the product is split for manufacture.
- A combined look at all of the predicted critical cross relationships can assist a development engineer to focus attention to a select number of sections for a certain manufacturing split.
- The description and evaluation schemes can be used in a sequence (a set theoretical approach for the component as a whole followed by EF-M trees for sections in the component) to generate and evaluate information about integrated products.

4.4 Paper D: Optimization-Based Architecture Decisions

An Optimization-based Approach for Supporting Early Product Architecture Decisions

4.4.1 Objective

The engine structures can be considered to have two primary functions: providing a flow path for the core flow in the engine and providing a mechanical load path for the transfer of interface loads. In this paper, the creation of a preliminary load path design based on optimization techniques and already available information with the component developer is discussed. Optimizing an initial load path configuration against a number of loading scenarios will help to establish a configuration that is robust to possible future load changes and can be used as the basis for detail design.

4.4.2 Approach

A simplified model for the engine structure is created, as shown in Figure 20(a). The interfaces for the structure (shown as Points 1, 2 and 3 in Figure 20[b]) is estimated from previous knowledge of such structures. The load carrying configuration (LCC; the arrangement of structural elements to carry loads) is shown by the red-highlighted area in Figure 20(a). The engine structure must withstand a number of loading scenarios from the operation of the engine and flight maneuvers. Loads are carried to and from the interface points through the structure by means of the LCC. The configuration can be changed by moving Points 4, 5, 6 and 7 axially and radially, as indicated in Figure 20(b). The position of the points are the design variables in the load path optimization. A typical loading scenario on the structure is shown in Figure 20(c).

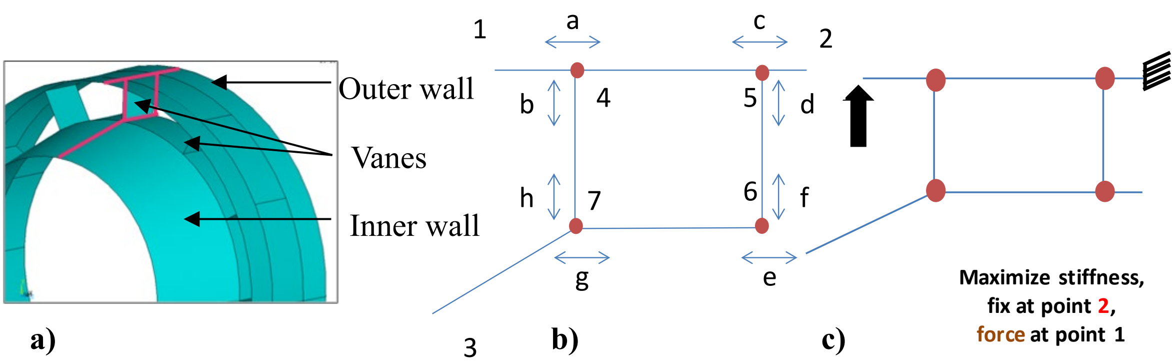


Figure 20 Load carrying configuration (LCC) optimization arrangement for an engine structure. a) Simplified geometry representation for the structure. LCC is highlighted in red. A “3D” structure can be constructed from the “2D” LCC using appropriate geometric operations. b) Construction and interface points for the LCC. Movement of the red-highlighted construction points axially and radially changes the LCC. C) Typical loading scenario and associated optimization statement for the structure.

A two-stage optimization approach is adopted to optimize the LCC. Individual loading scenarios are considered on the LCC first, and optimum configurations for the LCC for each individual loading scenario are determined. A total of eight scenarios were considered. Based on the results for the individual loading scenarios, a smaller number of scenarios are down selected. The down-selected problems are solved using an optimization coordination scheme, such that a compromise LCC is determined. The coordinated optimization scheme is the non-hierarchical analytical target cascading (Tosserams *et al.*, 2010).

4.4.3 Results and Findings

The results of the coordinated optimization are shown in Figure 21. In the figure, (a) represents the starting LCC, and (b) represents the optimized LCC. The LCCs in (c) and (d) represent the solutions from individual optimizations for the two problems considered for coordination. The positions of Points 4, 5, 6 and 7 (see Figure 20[b]) in (b) lie between the positions for the same points in (c) and (d). Clearly, results from the coordination problem are a compromise between the solutions for individual problems.

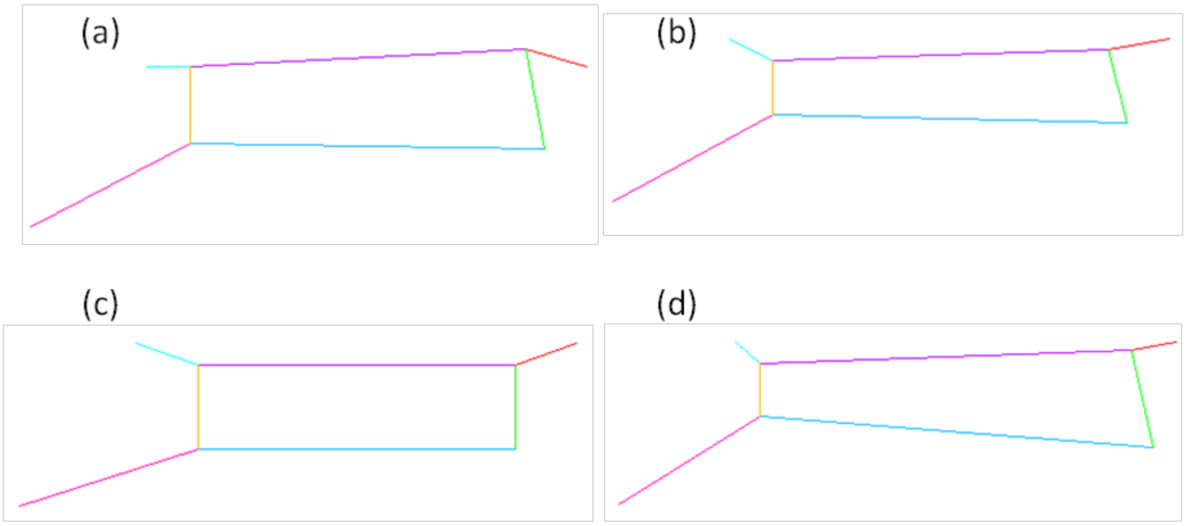


Figure 21 Results from the two-problem optimization coordination. Note that the two problems are not shown in this summary. (a) Starting LCC (b) Optimized LCC after two-problem coordination (c) Individual optimum for the first problem (d) Individual optimum for the second problem

The following findings can be highlighted from the paper:

- The end points of the LCC represents the allowable volume for eventual or concurrent aero-thermodynamic design for the flow path.
- The applicability of MDO schemes for functionally integrated products is demonstrated.
- The mechanical load path and the gas flow path are the chief aspects of the architecture of engine structures. A wider implementation of the method including gas flow path aspect of the architecture is desirable.

4.5 Paper E: Demonstration of Architecture Description

Modeling Integrated Product Architectures: An Aero Engine Component Example

4.5.1 Objective

The objective of this work is to demonstrate the architecture description method described in Paper C. In the paper, the method was applied to a section of an engine structure, and here, it is applied to a complete engine structure in sequence.

4.5.2 Approach

The architecture descriptions are created for the engine structure developed as part of the VITAL (European Commision Transport Research and Innovation Monitoring and Information System, 2005) engine program. An image of the structure is provided in Figure 16. Descriptions based on two manufacturing options are made (the options can be seen in the appended paper). The application sequence for the method suggested in Paper C is used here, which is presented in Table 6.

Table 6 Steps for sequential architecture description of an aero-engine structure

Step	Description
1	Create set theoretical architecture descriptions for all manufacturing options
2	Generate graphs for the descriptions; calculate degree centrality of nodes and edge-betweenness centrality of relations
3	Identify relevant nodes and relations based on the centrality values. The nodes are the functions and the sections of the structure, and, the edges are the function–section relations
4	Model the sections and functions of interest from Step 3 in detail using EF-M trees
5	Generate graphs based on the EF-M trees. Calculate centralities and analyze the results

4.5.3 Results and Findings

The EF-M-based graph descriptions for a function identified as important ("transfer core flow") in both the manufacturing options are displayed in Figure 22. The figure demonstrates that the section "vane" has the highest degree centrality in both options (two incoming edges). The core flow transfer function is critically dependent on the "vane" section, which is confirmed by the high degree centrality. A high edge-betweenness centrality is reported for the cross-relationship "vane-contain flow at inner radius." This implies that irrespective of the manufacturing options, inner flow containment is affected by the vanes.

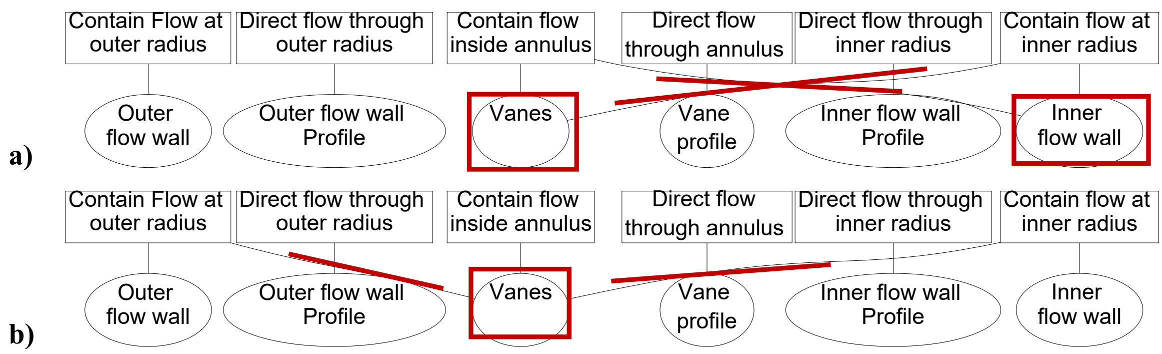


Figure 22 Graphs corresponding to EF-M trees for the "transfer core flow" function. a) Split option-01 b) Split option-02

The following findings can be highlighted from the paper:

- The paper demonstrates the systematic architectural description of complete aero-engine structure, the analysis of the description and, following this analysis, the examination of individual regions. This can result in the identification of previously unseen relationships and thereby improve the design or offer possibilities for a beneficial redesign.

- The calculation of centrality offers the possibility of quantitative evaluations of the architecture.

4.6 Paper F: A Complexity Metric for Engine Structures

A Simulation-assisted Complexity Metric for Design Optimization of Integrated Architecture Aero-engine Structures

4.6.1 Objective

The objective of this work is to develop a quantitative means to compare the architecture of aero-engine structures. Such a metric is necessary for the design evaluation and optimization of the structures and for obtaining cost projections for the structure's development efforts. The comparison is achieved by adapting separate system-wide and product-specific complexity metrics so that they can be applied to engine structures. Adaptation is necessary because the metrics are developed for multi-part products or multi-component systems.

4.6.2 Approach

The system-wide scheme of complexity is adapted from Sinha and de Weck (Sinha and de Weck, 2016). The system complexity is expressed in terms of three constituent complexities as

$$C = C1 + C2C3 \tag{8}$$

The first constituent $C1$ is the sum of individual complexities of system components. For the integrated architecture structure, $C1$ is calculated for the different sections in the structure.

The complexity for individual sections is determined based on a PAG (Ameri *et al.*, 2008). The nodes of the graph are the section types and dimensions for the sections. The edges of the graph are the relations between section types and dimensions. Based on the PAG, complexity can be calculated as

$$C_{PAG} = (n_{nodes} + n_{edges})\, ln\big(n_{node\ types} + n_{relation\ types}\big), \tag{9}$$

where n_{nodes} is the total number of nodes in the PAG, n_{edges} is the total number of edges, $n_{node\ types}$ is the number of types of nodes and $n_{relation\ types}$ is the number of types of relations in the PAG.

The two types of nodes in the PAG are the names of the sections and dimensions of the sections. The section–dimension relation is the only type of relation. Thus, the values of $n_{node\ types}$ and $n_{relation\ types}$ are 2 and 1, respectively.

The second constituent of complexity, $C2$, is called interface complexity. For a system, it is based on the type of interaction (flow of material, energy or signal) among components. The expression to determine $C2$ is given as

$$C2 = \sum_{i=1}^{n}\sum_{j=1}^{n} \beta_{ij}A_{ij} \tag{10}$$

where $A_{ij}$are the elements of the adjacency matrix for each type of interaction, and β_{ij} are the complexities for each interaction. For the structures, $C2$ is determined based on sectional interactions estimated as load paths corresponding to various loading scenarios. The load path determination approach is detailed in Subsection 2.5.2. The elements of the adjacency matrix are given as

$$A_{ij} = \begin{cases} 1 \text{ if } (\mathrm{i} \neq \mathrm{j}) \text{ and components i and j interact} \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{11}$$

The interaction of components in Equation (11) is the passage of a load path through sections i and j. Sections are assumed to have equal importance in transferring the load and $\beta_{ij} = 0.5$.

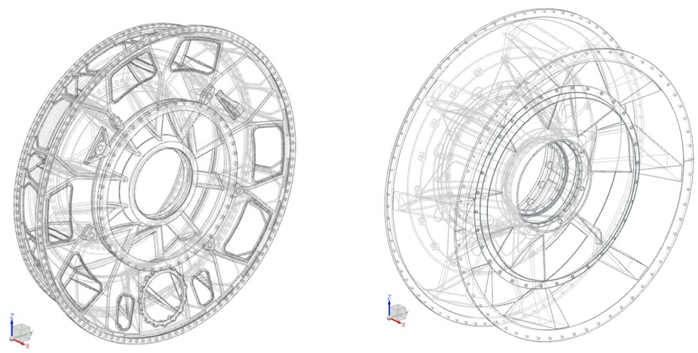
The third constituent of complexity, $C3$, called topological complexity, is based on the layout of components in a system. In order to compute $C3$, an aggregated ajacency matrix is created based on all relevant interactions present in the system. For the structures, the aggregated adjacency matrix is simply the mechanical connectivity of the sections. Based on the mechanical connectivity matrix, $C3$ is calculated as

$$C_3 = \frac{1}{k}\sum_{i=1}^{k} \sigma_i \tag{12}$$

where k is the total number of sections, and σ_i is the i^{th}singluar value. When a number of structures are to be compared, k will be the number of elements in the set of all sections, for all considered structures.

4.6.3 Results and Findings

The complexities of the structures can be calculated using the method just described. The resulting complexities of two engine structures are displayed in Figure 23. Even though, the structures are seemingly similar, they have very different values of complexity.



Structure 01, C = 465 Structure 02, C = 583

Figure 23 Complexity for two aero-engine structures.

The following findings can be highlighted from the paper:

- Functional integration is a source of complexity for aero-engine structures.
- Complexity metrics should consider the architecture of the structure and thereby the primary functions. This enables architecture comparisons using the metric.
- The complexity metric can be used as a proxy for design effort and related cost

projections. The more complex the structure, the more the likelihood of an increased development cost.

- Good designs are often identified as being less complex (Pugh, 1990), and the metric can form an objective function in design optimization for the structures.

4.7 Summary of Major Results

Although a number of results and findings proceeded from this research, the major results are a collection of five methods. These methods are listed below.

R1. A systematic procedure to identify F-M information for functionally integrated products and to represent the information using EF-M trees and CCs

R2. An approach to estimate system–component interaction effects, using disciplinary coupling between system (engine) and component (engine structures) simulations

R3. A component architecture description and evaluation scheme for functionally integrated aero-engine structures based on set theory, graph theory and EF-M representations

R4. An approach for determination of the preliminary LCC for engine structures based on a MDO coordination technique

R5. A complexity metric for aero-engine structures incorporating the structural layout of the component and the main function of load transfer

5 Discussion

The answers to the research questions formed in Chapter 1 are provided in this chapter, based on the results and findings from the appended publications. This chapter also examines the quality of the research by reviewing each publication using the methods discussed in Chapter 3.

5.1 Answering the Research Questions

The answers to the four RQs formed in the introduction (see Section 1.4) are formulated in the following subsections.

5.1.1 RQ1

What are the effects of engine structures on the performance of the engine?

The performance of an engine is expressed in terms of thrust specific fuel consumption, or simply the SFC (the amount of fuel needed in gram to produce unit thrust for one second, expressed in g/Ns (Saravanamuttoo, 2009)). A number of engine characteristics, such as weight and component efficiencies, affect SFC. Assessing the effect of engine structures on these characteristics will enable estimating a static structure's influence on engine performance.

A partial answer to this question is provided by the study in Paper B, considering the functioning of a TRS in the engine. The TRS leads the core gas flow from the low-pressure turbine to the exit nozzle. As the gas flow is led through the structure, an undesirable drop in pressure occurs, which reduces the efficiency of the low-pressure turbine. The reduced turbine efficiency results in reduced engine SFC. The pressure drop may be reduced by improving the aerodynamic design for the core gas flow path. From the findings in Paper B, a good aerodynamic design of the TRS flow path can result is 0.9% savings in SFC. When computed over a year, the reduced SFC can result in significant fuel-related cost savings for a commercial airliner.

In addition to the pressure drop in the core flow, the weight of the TRS and the stiffness at the interfaces are all factors that can affect engine performance. Detailed evaluations with systematically defined coupling interfaces are necessary to further develop the knowledge regarding RQ1. This is discussed further in Section 6.3.

In summary, one of the effects of an engine structure on the performance of the engine is its influence on the SFC. The influence is due to the variation in the effectiveness of the structure's flow transfer function. Enhancing the effectiveness of flow transfer (reducing the pressure drop) can reduce the SFC.

5.1.2 RQ2

What characterizes the architecture of the product and how to represent the architecture?

For ascertaining the architecture of the engine structures, prevailing definitions (Section 2.2) can be used that state that architecture is contained in the way a product is organized so that the functions required of it are satisfied. For products such as engine structures, one monolithic structure satisfies all functions required of it. To demonstrate the organization of the product, it is necessary to identify the functions of the product and regions in the product that participate in satisfying the functions. This leads to the concept of “sections” in the structure. Sections are identifiable regions in the structure that can be thought of as satisfying one or several functions. The engine structure can be completely described using section names, and the functions of the structure can be associated with relevant sections using any of the existing architecture representation schemes presented in Section 2.3. Here, an EF-M tree was used for modeling the architecture. The EF-M tree allows the encapsulation of the function–section relations in a CC object that allows the later reuse of the relations. This approach is detailed in Paper A.

Due to the physical size of the structures, they are weld fabricated from a number of segments. The manner of splitting the structure into segments is an important characteristic of the structure, and this affects the architecture by affecting the functional satisfaction. For example, it is better not to split sections that carry loads, as a weld line in the load path can create crack initiation problems. The identification, association and representation of function–section information cannot ascertain the architectural influence due to the manufacturing split options adopted. Therefore, a representation scheme that enables the influence of segmentation options is necessary.

Two approaches can be adopted for including the segmenting schemes in the architecture representation, as detailed in Paper C. The first is to consider the functions, segmenting options and sections in the product as belonging to separate sets. Using methods in set theory (composition of relations), relationships can be identified between the set of functions and sections through the set of manufacturing segments. This results in different function–section relationships based on the chosen segmenting option. The second approach is to use the EF-M tree. Different segmenting options can be directly represented in the EF-M tree, and the effects of choosing a certain option can be manually determined. Both approaches result in function–section relationships (architecture representations) that can be converted into graphs. Using the properties of graphs, insights about the architecture and the influence of manufacturing segmenting option on the architecture can be derived. The mathematical representation of the architecture in a graph enables the emergence of previously overlooked relations among functions and sections and the visualization of the product in a non-physical (not using CAD geometries) manner.

Answering RQ2 led to the identification of two fundamental aspects of the architecture of the product, which are the mechanical load path and the gas flow path. The function of any aero-engine static structure considered in this thesis can be considered the provision of a mechanical load path and a gas flow path at a specified location in the engine.

In summary, the architecture of the product lies in the relationship between the various functions of the product and the sections in the product, depending on the chosen manufacturing segmenting option. Architecture representation can be achieved using an EF-M tree, a graph or a combination of both, each representation offering insights about the architecture.

5.1.3 RQ3

How can the architectural insights derived for the product be utilized in initial design stages?

The answer to RQ2 proposes the idea of a mechanical load path and a gas flow path as the fundamental aspects of the architecture of aero-engine structures. The manner of splitting the integrated structures for manufacturing affects the architectural aspects and consequently the design. Paper C, after proposing the architecture description method, hinted at a sequential delineation and examination of the structure’s architecture. Once the geometrical description of the structure and the preliminary manufacturing split options are available, implications for different regions can be examined, and appropriate design steps can be taken. Thus, the two insights – the load path-flow path aspect of the architecture and the sequential exposition of the product's architecture – can be used in early design stages.

The load path aspect can be considered in isolation by studying the effects of possible loading scenarios on a so-named preliminary LCC. An LCC is the arrangement of the structural elements such that an applied load at one point is transmitted to another desired point. The LCC represents the desired preliminary load path in the structure. By varying the configuration according to the possible loading scenarios, an LCC that is applicable to a variety of scenarios can be derived. This provides a validated structural load path for detail design.

The suggested procedure for architecture exposition in Paper C was used in Paper E. Graph-based component descriptions for an entire component were first described, followed by the detailed consideration of individual functions or sections using EF-M trees. The properties of the graphs (centralities for nodes and edges) can be used to identify critical functions and sections of the product for which detailed descriptions are needed. Such methods allow both visual and quantitative identification of the primary functions, their interdependencies and the relations of the functions with physical regions in the product. A shortcoming of the architecture description method is the non-inclusion of the load path and the flow path in the representation scheme and the failure to provide a single quantitative means for comparison. This led to RQ4, wherein a quantitative means for architectural comparison was sought.

In summary, the immediate usability of the architectural insights lie in the setting of a preliminary LCC for the structure and the early identification of the consequences of manufacturing choices on functions of the structure and the regions that satisfy the functions. Increasing the architectural understanding of existing structures will further assist in making correct decisions in early design stages.

5.1.4 RQ4

How can a quantitative metric be created for integrated architecture products so that comparison among products of the same class is made easier?

For generating quantitative metrics for integrated architecture products, two methods are possible. The first is to use the graph-based architecture description scheme and use properties of the graphs as metrics. The second is to create a complexity metric for the products based on measures in the existing literature.

The first method to create the comparison metric is discussed in Paper E. In the paper, the degree centralities (number of incident edges on a node) of graph-based architecture descriptions were used to identify the functions that require the most sections and the sections that contribute to satisfying the most functions. A function with a high degree centrality will be connected to several sections. Such functions can be considered most vulnerable as damages to any of their connected sections will adversely affect them. Design efforts may be directed to lessen the sectional dependency of functions with high degree centralities. Conversely, a section with a high degree centrality will be connected to several functions. Since damages to this section can affect a number of functional requirements, measures to insure this section against damages must be adopted so that no functional failure occurs. A ranking of the functions and sections based on their centralities may also be performed, which can be considered indicative of their architectural impact.

The second method for creating a comparison metric is based on existing product complexity measures in the literature. Separate measures for system-wide and component-specific complexities must be adapted for application to functionally integrated products. A system-wide metric, proposed by Sinha and de Weck (2016), and a component-specific complexity metric, proposed by Ameri *et al.* (2008), can be considered for this purpose. The system-wide metric is created based on the complexities for individual components, the layouts of the components and their interaction. For an integrated architecture product such as an aero-engine structure, the metric can be created based on the complexities for individual sections in the component (sections are elaborated in the answer to RQ2), the layout of the sections and the interaction among the sections, estimated as load paths.

Engine structures of the same class share similar functional features. Two different engine structures can have a similar number and shape of sections. However, different load paths will exist in the structures, depending on the loading scenarios from the respective engines the structures serve. Since the load paths for various loading scenarios in the structures will differ, the complexity of the considered structures will also differ, thus enabling comparison. In addition to comparison, the metric can also be used as an objective in the design optimization of structures and as a proxy for development effort (the higher the complexity, the higher the development effort).

In summary, comparison metrics for functionally integrated products can be developed from both the graph-based architecture description for the product and the adaptation of existing measures of product complexity. To develop the metrics of complexity, parallels to the foundational concepts (the presence of multiple components and interaction among components) of existing metrics must be found in functionally integrated products. Considering a sectional division and interaction of sections through load paths enables the metric to be created.

5.2 Research Quality

5.2.1 Scientific Confidence: Applying the Validation Square

The validation square approach, used to ascertain the quality of research, was discussed in Section 3.2. In this chapter, the validation square is applied to the methods presented in all publications. Each step in the validation square is applied to the respective publications that propose the methods.

(1) Accepting the construct's validity

Paper A The expected outcome of this work is in expressing the function to section relationships in EF-M trees followed by encapsulating them in CCs. The constructs (individual methods) involved are the SOP, EF-M trees and the CC method, which are well founded in the literature.

Paper B The constructs used (engine performance calculations and turbine efficiency calculations) are well-established analytical equations in the literature. The pressure drop data was generated using validated in-house simulation tools at GKN.

Paper C The constructs used in this publication are founded on EF-M trees, CC methodology, graph theory and set theory. All methods are well established in the literature and have been used in a number of publications of similar nature.

Paper D Liner static FE simulations using shell elements, the coordination of MDO problems based on non-hierarchical analytical target cascading (Tosserams *et al.*, 2010; Kang *et al.*, 2014) and the solution of the MDO problem using an industrial strength research solver NOMAD (Le Digabel, 2011) were the individual constructs used in this work. Similar to other publications, the used methods are well established in literature and have been used on similar problems.

Paper E Paper E is based on the methods proposed in Paper C. Thus, constructs in Paper E are validated using the same reasoning for Paper C.

Paper F The constructs used in Paper F are a system-wide complexity metric (Sinha and de Weck, 2016), a component-specific complexity metric (Ameri *et al.*, 2008) and the evaluation of load paths using structural optimization. All constructs are well founded in the literature.

(2) Accepting the method's consistency

Paper A At all stages of evaluation in the method, all necessary data is available to each successive step from the preceding step. Provided the labels for functions and sections are defined at the beginning and do not change, the method results in no inconsistencies.

Paper B The constructs adopt no assumptions that affect the trustworthiness of the results generated from a given input. At all stages of evaluation, all necessary data is available to each successive step from the preceding step.

Paper C At all stages of evaluation, all necessary data is available to each successive step from the preceding step. Provided the labels for functions and sections are defined at the beginning and do not change, the method results in no inconsistencies. Further, the calculation of centralities and singular value decay patterns are based on validated algorithms in commercial software. The method is judged to not result in any inconsistencies. The method was first tested on a pump casing, which is functionally integrated in a similar manner as an engine structure.

Paper D The methods adopt no assumptions that affect the trustworthiness of the results generated from a given input. At all stages of evaluation, all necessary data is available to each successive step from the preceding step.

Paper E The same reasoning for Paper C is followed here, as Paper E demonstrates the method proposed by Paper C.

Paper F At all stages of evaluation, all necessary data is available to each successive step from the preceding step. Provided the sections are consistently identified for all structures, the method results in no inconsistencies. The method was tried on two illustrative structures that have similar load carrying functions as engine static structures.

(3) Accepting the example problems

Paper A A real-life engine structure was considered for evaluation.

Paper B The coupling was achieved for data generated from real engine structures.

Paper C The vanes considered for modeling are typical of engine structures.

Paper D The example problem is a simplification of an actual engine structure LCC.

Paper E A real-life engine structure was considered with manufacturing options that represent those adopted in actual situation.

Paper F Two real-life engine structures were considered for modeling.

(4) Accepting usefulness of the method for example problems

Paper A The presented method clearly delineates and systematically associates sections and functions for the structure.

Paper B The estimation of the SFC variation is due to the application of the method.

Paper C The method clearly supports architecture evaluation based on manufacturing splits.

Paper D For selecting a preliminary load path, the method functions satisfactorily.

Paper E Similar to Paper C, the method clearly supports architecture evaluation based on manufacturing splits.

Paper F The method can produce quantitative measure for the engine structure's architecture, based on the layout of various sections in the structure and the load path aspect of the architecture.

(5) Accepting that usefulness is due to application of the method

Paper A Compared to experience-based judgments, the method offers the systematic documentation and reuse of design data, which was non-existent at the collaborating firm in the presented form.

Paper B Estimating the influence of the functioning of engine structures on engine SFC at the collaborating firm was based on information supplied by engine OEMs. The method offers a preliminary estimate of how SFC is likely to be affected by the operation of the structure (causing a pressure drop) without relying on OEM data.

Paper C Previously, methods did not exist for evaluating the architecture of functionally integrated structures based on manufacturing splits. The proposed method presents a clearly defined method to evaluate the functional influence of the manner of splitting functionally integrated structures for manufacture.

Paper D The current method to evaluate the load path aspect of architecture is based on evaluations on detailed geometry that are time consuming. The proposed method explicitly considers the load path and provides a fast and acceptable solution. This is in addition to demonstrating the applicability of MDO approaches to the design of functionally integrated structures.

Paper E Paper E demonstrates the method in Paper C for a complete engine structure; the same reasoning is adopted here as for Paper C.

Paper F Compared to experience-based judgments, the proposed method offers a clearly defined quantitative metric for comparison of structures.

(6) Accepting usefulness beyond example problems

Based on the validation square technique, since Steps 1–5 are justified, it can be concluded that for products having precisely the nature of functionally integrated engine structures, the methods are fit to be applied.

5.2.2 Industrial Confidence: Technology Readiness Levels Assessment

The TRL levels for the various methods proposed in the publications are assessed in this subsection. Paper E involves the demonstration of the method in Paper C on a complete engine structure, and TRL judgment is not provided. Except for in Paper E, the methods in all papers are judged to have TRLs, which are displayed in Table 7.

Table 7 TRL judgments for methods presented in appended publications

Listing in Section 4.7	**Method For:**	**Publication**	**TRL Level**
R1	Function–Means modeling for engine structures	Paper A	2–3
R2	Interdisciplinary coupling for engine structures design	Paper B	3
R3	Architecture representation and evaluation for engine structures	Paper C	4
R4	Optimization-based selection of the LCC of engine structures	Paper D	3
R5	Creating a complexity metric for engine structures	Paper F	4

In Table 7 the method for function–means modeling in Paper A is based on the adaptation of an academically developed method (EF-M tree), so that it can be applied on an engine structure. The underlying technology (understood here as method) is well developed, and the method in the publication serves as proof-of-concept. The method for interdisciplinary coupling in Paper B also uses well-developed underlying methods, although the present procedure to estimate SFC requires additional variables (such as the weight of the structure) for it to be accurate. Therefore, Paper B serves as proof-of-concept, and the TRL for the method in it is estimated as 3. For the LCC selection method in Paper D, the final estimate of the configuration requires additional inputs from the flow path aspect of the architecture. Therefore, the method in Paper D is also considered as proof-of-concept with TRL 3. Paper C, together with Paper E, demonstrates the application of the architecture description method on real-life engine structures. Even though the structures are actual engine structures, they have been developed as part of research programs rather than commercial product development programs. Thus, the method in Paper C can be considered to have crossed proof-of-concept stage and validated on laboratory environments, thus obtaining a TRL of 4. In the case of the method in Paper F, the underlying methods are well-established in a variety of cases in the literature. The developed method can be directly applied to an engine structure. The demonstration of the method is on structures developed as part of research programs, similar to application in a laboratory environment. This gives a TRL level of 4 to the method in Paper F.

6 Conclusion and Further Work

This chapter provides concluding remarks for the research in this thesis, together with suggestions for future work.

This thesis has developed methods to enable the visual and quantitative representation of an engine structure's architecture. Architecture representations using function–section graphs or EF-M trees can directly discover previously unidentified relationships between the structure's functions and sections. This increased architecture knowledge is useful in future product improvements. For example, the interface attachment function of the structures at flanges are not generally thought to influence the core flow transfer function. Manufacturing splits coupling both functions will make losing one function adversely affect the other. Such splits can be avoided for the concerned structure. The developed complexity metric can comprise the basis for benchmarking functionally integrated products with similar functional features. Determining the relationship between the metric and design efforts for completed projects can provide guiding cost projections for future product development projects. Based on the metric, decision support for accepting development projects for products of high complexity can be provided. The quantitative metric can also be used as a "design efficiency" (D. Lefever and Wood, 1996) metric that indicates the desirability of a certain product design with respect to its functional fulfillment. For example, a lower value for the metric indicates a simpler mechanical load path. Using the metric as an objective function in an optimization scheme can create designs with simpler load paths. In summary, with the set of methods developed in this thesis, development engineers can "dissect" their products and compare alternative architectures for them.

Even though the developed methods are judged to have TRL of up to 4, implementation in actual development programs can take time. One reason is the pressure placed on projects by the time necessary to learn new methods (Eckert, 2013), which might delay the methods' adoption. However, founding the methods on familiar mathematical concepts and exemplifying their application on real-life engine structures can result in quicker adoption by development engineers.

6.1 Scientific Contribution

The scientific contributions from this thesis are summarized as follows:

- Advancement of the *functionally integrated product architecture* concept
    - This thesis advances the concept of integrated architecture products by providing the definition for functionally integrated product architecture and developing methods to support the design of such products. Despite a large number of literatures and methods being available for modular architecture products (Gershenson *et al.*, 2004; Jose and Tollenaere, 2005), the design and development of integral architecture products has received less attention.
- Methods for architecture modeling and evaluation for functionally integrated products
    - The methods proposed (listings R1 and R3 in Section 4.7), making use of graph and set theory in addition to EF-M trees, are novel and have not been applied on functionally integrated products. In addition, the implicit incorporation of manufacturing options in architecture descriptions enables the evaluation of alternative product architectures, in light of manufacturing options (as addressed in Paper C).

- Adaptation and development of a complexity metric applicable to functionally integrated products (listing R5 in Section 4.7)
    - The application of complexity metrics to monolithic real-life engine structures has not been demonstrated, although complexity metrics are well-developed in the literature. The incorporation of load path into the formulation of the metric is novel and has not been previously considered.
- Demonstration of the applicability of MDO coordination techniques to functionally integrated products (listing R4 in Section 4.7)
    - Due to the inseparability of the products, the application of various MDO techniques on functionally integrated products is not straightforward, as specific sub-problems are difficult to identify. This thesis presents an approach to apply MDO concepts to functionally integrated products by considering relevant aspects of the architecture (such as the mechanical load path).

6.2 Industrial Contribution

The primary intention of this research project was to influence the industrial practice of aero-engine structure design. The industrial contributions of this thesis are summarized as follows:

- Ability to describe the product architecture of functionally integrated aero-engine components
    - Based on the developed methods (listings R1 and R3 in Section 4.7), development engineers obtain the capability to visually represent and quantitatively evaluate alternative product architectures. A TRL-4 judgment for one of the methods indicates the possibility for straightforward implementation in development projects.
- Ability to compare alternative engine structure designs based on a product complexity metric
    - The complexity metric (listing R5 in Section 4.7) can be used to estimate development efforts for engine structures and obtain realistic cost projections. Similar to the architecture description method, a TRL-4 judgment signifies the possibility of simpler implementation in new development projects.
- Capability to visualize load paths using existing FE simulation software
    - An approach is developed to visualize the existing load paths on engine structures. This is helpful for development engineers to grasp the effects of their design changes on the architecture of the structure (the load path).
- Ability to formulate and solve MDO problems for integrated architecture aero-engine structures
    - By isolating key aspects of the architecture of the structure (the load path and flow path, in this case), a problem decomposition and consequent application of MDO techniques is facilitated (listing R4 in Section 4.7).
- Ability to obtain preliminary estimates of the effects of component operation in the system (listing R2 in Section 4.7)
    - Performing coupled simulations between calculations for engine performance and component detail design can create insights that lead to component design improvements. Such exercises also facilitate cooperation among various disciplinary engineers involved in the development program (e.g. engine thermodynamic performance engineers and CFD simulation engineers).

The suite of proposed methods (R1–R5 in Section 4.7) together enhance a development engineer's ability to compare and evaluate alternative architectures for an engine structure.

6.3 Further Research

To answer RQ1, a single factor (drop in pressure in core gas flow through a TRS) was coupled to system performance (engine SFC) calculations. A number of other factors, such as the weight and stiffness of the component, can affect system performance. An extensive study considering all factors can provide a more complete understanding of the static structure's influence on the system. This enables the evaluation of the susceptibility of the system to a changed component architecture. Careful consideration of system–component interfaces, depending on the type of analysis, will be necessary for such studies.

Since the way of organizing a system (its architecture) depends on the number, arrangement and interconnection of its components, a structural complexity metric is representative of system architecture. For a functionally integrated structure, its architecture is strongly related to its manufacturing segmenting options (the architectural implications of different manufacturing options is presented in Paper C). The inclusion of manufacturing options in the complexity metric is identified as a potential future work. Additionally, possibilities to utilize the metric for decision support in knowledge based systems and related software is relevant to be explored. It is of interest to embed the metric in the in-house, computer-based multidisciplinary design platform called the "engineering work bench". The metric can be the basis of selecting or eliminating the designs generated using the platform.

The application of MDO techniques to the structures is another direction for additional research. This thesis has considered the coordinated optimization of an engine structure's mechanical load path. A combined consideration of the mechanical load path and the gas flow path in a coordinated design optimization could reveal useful insights about the product's architecture. The careful decomposition of both the load and flow paths in the structure is necessary to perform such a work.

Different schemes for identifying the load path in the structure can also be investigated. The possibility to visualize load lines in the structure according to design changes is particularly interesting and would aid development engineers.

REFERENCES

ACARE (2017) *Strategic Research & Innovation Agenda: 2017 Update Volume 1*. [Online]. Available at: https://www.acare4europe.org/sites/acare4europe.org/files/document/ACARE-Strategic-Research-Innovation-Volume-1.pdf.

Alexander, C. (1964) *Note on the Synthesis of Form*. Cambridge, Massachusetts: Harvard University Press.

Altfeld, H.-H. (2016) *Commercial Aircraft Projects: Managing the Development of Highly Complex Products*. New York: Routledge.

Ameri, F., Summers, J. D., Mocko, G. M. and Porter, M. (2008) 'Engineering design complexity: an investigation of methods and measures', *Research in Engineering Design*, 19(2-3), pp. 161-179.

Andersson, F., Nilsson, P. and Johannesson, H. (2000) 'Computer based requirements and concept modelling-information gathering and classification', *DETC'00 2000 ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference*. Baltimore, Maryland.

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

Andreasen, M. M. and Hein, L. (2000) *Integrated product development*. Lundtofte: IPU.

Aurisicchio, M., Bracewell, R. and Armstrong, G. (2013) 'The function analysis diagram: Intended benefits and coexistence with other functional models', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 27(03), pp. 249-257.

Avellan, R. (2011) *On the Design of Energy Efficient Aero Engines: Some Recent Innovations* PhD thesis. Chalmers University of Technology.

Bashir, H. A. and Thomson, V. (1999) 'Estimating Design Complexity', *Journal of Engineering Design*, 10(3), pp. 247-257.

Bashir, H. A. and Thomson, V. (2001) 'Models for estimating design effort and time', *Design Studies*, 22(2), pp. 141-155.

Blessing, L. T. M. and Chakrabarti, A. (2009) *DRM, a design research methodology*. London: Springer.

Browning, T. R. (2016) 'Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities', *IEEE Transactions on Engineering Management*, 63(1), pp. 27-52.

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

Chakrabarti, A. (2001) 'Sharing in Design - Categories, Importance and Issues', *Proceedings of the International Conference on Engineering Design (ICED01), Design Methods for Performance and Sustainability*. Glasgow. pp. 563-570.

Chakrabarti, A. and Blessing, L. (2015) 'A review of theories and models of design', *Journal of the Indian Institute of Science*, 95(4), pp. 325-340.

Chakrabarti, A. and Singh, V. (2007) 'A method for structure sharing to enhance resource effectiveness', *Journal of Engineering Design*, 18(1), pp. 73-91.

Chalmers Research (2013) *Design for performance – advanced powerplant architecture and integration* [Web Page]. Available at: https://research.chalmers.se/en/project/?id=5770 (Accessed: 20 February 2019).

Chalmers Research (2017) *Design for Performance* [Web Page]. Available at: https://research.chalmers.se/en/project/?id=7860 (Accessed: 20 February 2019).

Chartrand, G. and Lesniak, L. (1996) *Graphs & digraphs*. Third edn. Boca Raton: Chapman & Hall.

Chu, D., Strand, R. and Fjelland, R. (2003) 'Theories of complexity', *Complexity*, 8(3), pp. 19-30.

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

Cooper, R. G. (1983) 'A process model for industrial new product development', *IEEE Transactions on Engineering Management*, EM-30(1), pp. 2-11.

Cooper, R. G. (2001) *Winning at New Products : Accelerating the Process From Idea to Launch*. Third edn. Cambridge, Massachusetts: Perseus Press.

COPE Sweden AB (2014) *Object Oriented Platform Development: About* [Web Page]. Available at: http://www.copesweden.se/About.html (Accessed: 22 Feb 2017).

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

CRESCENDO (2009) *Collaborative & Robust Engineering using Simulation Capability Enabling Next Design Optimisation* [Web Page]. Available at: http://cordis.europa.eu/project/rcn/93958_en.html (Accessed: 20 April 2016).

Creswell, J. W. (2014) *Research design : qualitative, quantitative, and mixed methods approaches*. Fourth edn. Los Angeles: SAGE.

Crilly, N. (2013) 'Function propagation through nested systems', *Design Studies*, 34(2), pp. 216-242.

D. Lefever, D. and Wood, K. (1996) 'Design for Assembly Techniques in Reverse Engineering and Redesign', *ASME Design Theory and Methodology Conference*.

Dixon, J. R. (1995) 'Knowledge-based systems for design', *Journal of Mechanical Design*, 117(2), pp. 11-16.

Dong, Q. (2002) *Predicting and managing system interactions at early phase of the product development process*. PhD thesis. Massachusetts Institute of Technology.

Donner, R. (2010) 'The GEnx Engine', *Aircraft Maintenance Technology*, 21(5), May 2010, pp. 12-15.

Eckert, C. (2013) 'That which is not form: The practical challenges in using functional concepts in design', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 27(3), pp. 217-231.

Eisenbart, B., Gericke, K., Blessing, L. T. M. and McAloone, T. C. (2017) 'A DSM-based framework for integrated function modelling: concept, application and evaluation', *Research in Engineering Design*, 28(1), pp. 25-51.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. and Monostori, L. (2012) 'Complexity in engineering design and manufacturing', *CIRP Annals*, 61(2), pp. 793-814.

Eppinger, S. D. (1997) 'A Planning Method for Integration of Large-Scale Engineering Systems', *International Conference on Engineering Design*. Tampere, Finland.

Eppinger, S. D. and Browning, T. R. (2012) *Design Structure Matrix Methods and Applications*. Cambridge, Massachusetts: MIT Press.

ESTECO SpA (2019) *modeFRONTIER* [Web Page]. Available at: https://www.esteco.com/modefrontier (Accessed: 11 March 2019).

Estefan, J. A. (2007) *Survey of Candidate Model-Based Engineering (MBSE) Methodologies* INCOSE MBSE Focus Group.

European Commision Transport Research and Innovation Monitoring and Information System (2005) *VITAL Environmentally Friendly Aero-Engine* [Web Page]. Available at: https://trimis.ec.europa.eu/project/environmentally-friendly-aero-engine#tab-outline (Accessed: 12 September 2018).

Ferjan, K. (2016) 'Another Successful Year', *IATA Airline Cost Conference 2016* Geneva.

Fujimoto, T. (2007) *Competing to be really, really good : the behind-the-scenes drama of capability-building competition in the automobile industry*. Translated by Miller, B. Tokyo: LTCB Int. Library Trust/International House of Japan.

GE Aviation (2018) *GE9X Commercial Aircraft Engine* [Web Page]. Available at: https://www.geaviation.com/commercial/engines/ge9x-commercial-aircraft-engine (Accessed: 19 February 2019).

Gershenfeld, N. (1999) *The Nature of Mathematical Modeling*. Cambridge, UK: Cambridge University Press.

Gershenson, J. K., Prasad, G. J. and Zhang, Y. (2003) 'Product modularity: Definitions and benefits', *Journal of Engineering Design*, 14(3), pp. 295-313.

Gershenson, J. K., Prasad, G. J. and Zhang, Y. (2004) 'Product modularity: measures and design methods', *Journal of Engineering Design*, 15(1), pp. 33-51.

Gokpinar, B., Hopp, W. J. and Iravani, S. M. R. (2010) 'The Impact of Misalignment of Organizational Structure and Product Architecture on Quality in Complex Product Development', *Management Science*, 56(3), pp. 468-484.

Hague, D. C. (2001) *Description of a turbofan engine product development process*. Masters thesis. Massachusetts Institute of Technology.

Hansen, C. T. and Andreasen, M. M. (2002) 'Two approaches to synthesis based on the domain theory', in Chakrabarti, A. (ed.) *Engineering Design Synthesis: Understanding, Approaches and Tools*. London: Springer London, pp. 93-108.

Hitchins, D. K. (2003) *Advanced Systems Thinking, Engineering, and Management*. Norwood: Artech House, Incorporated.

Hongxia, L., Shifu, Z., Chunhua, L. and Qi, L. (2015) 'Development Strategy Study for Commercial Aero Engine Companies in European Countries and the United States', *Procedia Engineering*, 99, pp. 804-811.

Horvath, I. (2001) 'A contemporary survey of scientific research into engineering design', *International Conference on Engineering Design*. Glasgow - Bury St Edmunds.

Hubka, V. (2013) *Principles of Engineering Design*. Burlington: Elsevier Science.

Hubka, V. and Eder, W. E. (1996) *Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*. London: Springer London.

Högman, U. and Johannesson, H. (2013) 'Applying stage-gate processes to technology development—Experience from six hardware-oriented companies', *Journal of Engineering and Technology Management*, 30(3), pp. 264-287.

Hölttä-Otto, K. and de Weck, O. (2007) 'Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints', *Concurrent Engineering*, 15(2), pp. 113-126.

ICAO (2017) *The World of Air Transport in 2017* [Web Page]. Available at: https://www.icao.int/annual-report-2017/Pages/the-world-of-air-transport-in-2017.aspx (Accessed: 17 August 2018).

Isaksson, O., Keski-Seppälä, S. and Eppinger, S. D. (2000) 'Evaluation of design process alternatives using signal flow graphs', *Journal of Engineering Design*, 11(3), pp. 211-224.

Jankovic, M. and Eckert, C. (2016) 'Architecture decisions in different product classes for complex products', *Artificial Intelligence for Engineering Design, Analysis and Manufacturing : AI EDAM*, 30(3), pp. 217-234.

Jarrett, J. P., Dawes, W. N. and Clarkson, P. J. (2007) 'An Approach to Integrated Multi-Disciplinary Turbomachinery Design', *Journal of Turbomachinery*, 129(3), pp. 488-494.

Jiao, J., Simpson, T. W. and Siddique, Z. (2007) 'Product family design and platform-based product development: a state-of-the-art review', *Journal of Intelligent Manufacturing*, 18(1), pp. 5-29.

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

Johnston, P. (2017) 'The Aero-Engine Business Model: Rolls-Royce's Perspective', in Richter, K. and Walther, J. (eds.) *Supply Chain Integration Challenges in Commercial Aerospace: A Comprehensive Perspective on the Aviation Value Chain*. Cham: Springer International Publishing, pp. 237-248.

Jose, A. and Tollenaere, M. (2005) 'Modular and platform methods for product family design: literature analysis', *Journal of Intelligent Manufacturing*, 16(3), pp. 371-390.

Jung, S., Asikoglu, O. and Simpson, T. W. (2018) 'A method to evaluate direct and indirect design dependencies between components in a product architecture', *Research in Engineering Design*, 29(4), pp. 507-530.

Jørgensen, K. (1992) *Videnskabelige arbejdsparadigmer [Scientific work paradigms]*. Institute for Production, Aalborg Univesity of Technology, Denmark

Kang, N., Kokkolaras, M. and Papalambros, P. Y. (2014) 'Solving multiobjective optimization problems using quasi-separable MDO formulations and analytical target cascading', *Structural and Multidisciplinary Optimization*, 50(5), pp. 849-859.

Keller, R., Eckert, C. M. and Clarkson, P. J. (2006) 'Matrices or node-link diagrams: which visual representation is better for visualising connectivity models?', *Information Visualization*, 5(1), pp. 62-76.

Kelly, D. W., Hsu, P. and Asudullah, M. (2001) 'Load paths and load flow in finite element analysis', *Engineering Computations*, 18(1/2), pp. 304-313.

Kim, G., Kwon, Y., Suh, E. S. and Ahn, J. (2016) 'Analysis of Architectural Complexity for Product Family and Platform', *Journal of Mechanical Design*, 138(7), pp. 071401-071401-11.

Kolmogorov, A. N. (1968) 'Three approaches to the quantitative definition of information', *International Journal of Computer Mathematics*, 2(1-4), pp. 157-168.

Kreimeyer, M. (2014) 'Implementing Architecture in Industry', *International Design Conference - DESIGN 2014*. Dubrovnik, Croatia.

Le Digabel, S. (2011) 'Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm', *ACM Transactions on Mathematical Software (TOMS)*, 37(4), pp. 44:1--44:15.

Levandowski, C., Michaelis, M. T. and Johannesson, H. (2014) 'Set-based development using an integrated product and manufacturing system platform', *Concurrent Engineering*, 22(3), pp. 234-252.

Lindemann, U., Maurer, M. and Braun, T. (2009) *Structural complexity management: an approach for the field of product design*. Berlin: Springer.

Malmqvist, J. (1997) 'Improved Function-means Trees by Inclusion of Design History Information', *Journal of Engineering Design*, 8(2), pp. 107-117.

Mankins, J. (1995) *Technology Readiness Level – A White Paper*.

Mankins, J. C. (2009) 'Technology readiness assessments: A retrospective', *Acta Astronautica*, 65(9-10), pp. 1216-1223.

Marhadi, K. and Venkataraman, S. (2009) 'Comparison of quantitative and qualitative information provided by different structural load path definitions', *International Journal for Simulation and Multidisciplinary Design Optimization*, 3(3), pp. 384-400.

Martin, R. (2016) 'The Race for the Ultra-Efficient Jet Engine of the Future', *MIT Technology Review*, 23 March.

Martins, J. R. R. A. and Lambe, A. B. (2013) 'Multidisciplinary Design Optimization: A Survey of Architectures', *AIAA Journal*, 51(9), pp. 2049-2075.

Maurer, M., Schneller, R. and Omer, M. (2014) 'A survey on complexity management in systems engineering', *8th Annual IEEE International Systems Conference, SysCon 2014*. Ottawa, Ontario, Canada.

Michaelis, M. T. (2013) *Co-Development of Products and Manufacturing Systems Using Integrated Platform Models*. PhD thesis. Chalmers University of Technology.

Mikkola, J. H. (2006) 'Capturing the Degree of Modularity Embedded in Product Architectures', *Journal of Product Innovation Management*, 23(2), pp. 128-146.

Min, G., Suh, E. S. and Hölttä-Otto, K. (2015) 'System Architecture, Level of Decomposition, and Structural Complexity: Analysis and Observations', *Journal of Mechanical Design*, 138(2), pp. 021102-021102-11.

Muller, G. (2013) 'Systems Engineering Research Methods', *Procedia Computer Science*, 16, pp. 1092-1101.

Naito, T., Kobayashi, H., Urushiyama, Y. and Takahashi, K. (2011) 'Introduction of New Concept U*sum for Evaluation of Weight-Efficient Structure', *SAE International Journal of Passenger Cars - Electronic and Electrical Systems*, 4(1), pp. 30-41.

National Academies of Sciences, Engineering, and Medicine (2016) *Commercial Aircraft Propulsion and Energy Systems Research: Reducing Global Carbon Emissions*. Washington, DC: The National Academies Press.

Newman, M. E. J. (2010) *Networks an introduction*. Oxford: Oxford University Press.

Otto, K., Hölttä-Otto, K., Simpson, T. W., Krause, D., Ripperda, S. and Ki Moon, S. (2016) 'Global Views on Modular Design Research: Linking Alternative Methods to Support Modular Product Family Concept Development', *Journal of Mechanical Design*, 138(7), pp. 071101-071101-16.

Pahl, G., Wallace, K. and Blessing, L. (2007) *Engineering design: a systematic approach*. Third edn. London: Springer.

Park, Y., Fujimoto, T. and Hong, P. (2012) 'Product architecture, organizational capabilities and IT integration for competitive advantage', *International Journal of Information Management*, 32(5), pp. 479-488.

Pedersen, K., Emblemsvag, J., Bailey, R., Allen, J. K. and Mistree, F. (2000) 'Validating design methods and research: the validation square', *ASME Design Theory and Methodology Conference*.

Prasad, B. (1996) *Concurrent Engineering Fundamentals* (2 vols). Upper Saddle River, New .Jersey: Prentice Hall PTR.

Pugh, S. (1977) 'Load lines: an approach to detail design', *Production Engineer*, 56(1.2), pp. 15-18.

Pugh, S. (1990) *Total design: integrated methods for successful product engineering*. Wokingham: Addison-Wesley.

Reich, Y. (1995) 'The Study of Design Research Methodology', *Journal of Mechanical Design*, 117(2A), pp. 211-214.

Reich, Y. (2010) 'My method is better!', *Research in Engineering Design*, 21(3), pp. 137-142.

Reich, Y. and Subrahmanian, E. (2013) 'Philosophy of design, science of design, engineering (of) design: what is your choice?', *Research in Engineering Design*, 24(4), pp. 321-323.

Reinman, G., Ayer, T., Davan, T., Devore, M., Finley, S., Glanovsky, J., Gray, L., Hall, B., Jones, C., Learned, A., Mesaros, E., Morris, R., Pinero, S., Russo, R., Stearns, E., Teicholz, M., Teslik-Welz, W. and Yudichak, D. (2012) 'Design for Variation', *Quality Engineering*, 24(2), pp. 317-345.

Robertson, D. and Ulrich, K. (1998) 'Planning for product platforms', *Sloan Management Review*, 39(4), pp. 19-31.

Sadin, S. R., Povinelli, F. P. and Rosen, R. (1989) 'The NASA technology push towards future space mission systems', *Acta Astronautica*, 20, pp. 73-77.

SAFRAN (2017) 'Open Rotor - Press Update October 03 2017'. Available at: https://www.safran-group.com/file/download/20171003_press_kit_event_open_rotor_istres_english.pdf (Accessed: 07 March 2019).

Sandberg, M., Tyapin, I., Kokkolaras, M., Isaksson, O., Aidanpaa, J. O. and Larsson, T. (2011) 'A Knowledge-based Master-model Approach with Application to Rotating Machinery Design', *Concurrent Engineering*, 19(4), pp. 295-305.

Saravanamuttoo, H. I. H. (2009) *Gas turbine theory*. Sixth edn. Upper Saddle River, New Jersey: Pearson Prentice Hall.

Sarkar, S., Dong, A., Henderson, J. A. and Robinson, P. A. (2013) 'Spectral Characterization of Hierarchical Modularity in Product Architectures', *Journal of Mechanical Design*, 136(1), pp. 011006-011006-12.

Seepersad, C. C., Pedersen, K., Emblemsvåg, J., Bailey, R., Allen, J. K. and Mistree, F. (2006) 'The Validation Square: How Does One Verify and Validate a Design Method?', in Lewis, K. E., Chen, W. and Schmidt, L. C. (eds.) *Decision Making in Engineering Design*. New York: ASME.

Shah, J. J. and Runger, G. (2013) 'What is in a name? On the misuse of information theoretic dispersion measures as design complexity metrics', *Journal of Engineering Design*, 24(9), pp. 662-680.

Shai, O. and Reich, Y. (2004) 'Infused design. I. Theory', *Research in Engineering Design*, 15(2), pp. 93-107.

Shibata, T., Yano, M. and Kodama, F. (2005) 'Empirical analysis of evolution of product architecture', *Research Policy*, 34(1), pp. 13-31.

Shinobu, M., Okamoto, D., Ito, S., Kawakami, H. and Takahashi, K. (1995) 'Transferred load and its course in passenger car bodies', *JSAE Review*, 16(2), pp. 145-150.

Siemens PLM Software Inc (2017) *HEEDS Software* [Web Page]. Available at: https://www.redcedartech.com/index.php/solutions/heeds-software (Accessed: 11 March 2019).

Sinha, K. and de Weck, O. L. (2016) 'Empirical Validation of Structural Complexity Metric and Complexity Management for Engineering Systems', *Systems Engineering*, 19(3), pp. 193-206.

Sobek, D. K., II, Ward, A. C. and Liker, J. K. (1999) 'Toyota's principles of set-based concurrent engineering', *Sloan Management Review*, 40(2), pp. 67-83.

Sosa, M. E., Eppinger, S. D. and Rowles, C. M. (2004) 'The Misalignment of Product Architecture and Organizational Structure in Complex Product Development', *Management Science*, 50(12), pp. 1674-1689.

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

Suh, N. P. (2005) *Complexity: Theory and Applications*. New York: Oxford University Press.

Summers, J. D. and Shah, J. J. (2010) 'Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability', *Journal of Mechanical Design*, 132(2), pp. 021004-021004-11.

Tamaskar, S., Neema, K. and DeLaurentis, D. (2014) 'Framework for measuring complexity of aerospace systems', *Research in Engineering Design*, 25(2), pp. 125-137.

TOICA (2013) *Thermal Overall Integrated Conception of Aircraft*. Available at: http://www.toica-fp7.eu/ (Accessed: 20 April 2016).

Tosserams, S., Kokkolaras, M., Etman, L. F. P. and Rooda, J. E. (2010) 'A Nonhierarchical Formulation of Analytical Target Cascading', *Journal of Mechanical Design*, 132(5), pp. 051002-051002-13.

Ullman, D. G. (2010) *The Mechanical design process*. Boston: McGraw-Hill.

Ulrich, K. (1995) 'The role of product architecture in the manufacturing firm', *Research Policy*, 24(3), pp. 419-440.

Ulrich, K. T. and Eppinger, S. D. (1995) *Product Design and Development (International Editions 1995)*. Singapore: McGraw-Hill, Inc.

Ulrich, K. T. and Seering, W. P. (1990) 'Function Sharing in Mechanical Design', *Design Studies*, Volume 11(Issue 4), pp. 223-234.

Waller, J., Parker, B., Hodges, K. and Walker, J. (2015) *Nondestructive Evaluation of Additive Manufacturing State-of-the Discipline Report Outbrief*.

van Eck, D., McAdams, D. A. and Vermaas, P. E. (2008) 'Functional decomposition in engineering: A survey', *ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2007*. Las Vegas, Nevada, USA.

Van Tooren, M., Nawijn, M., Berends, J. and Schut, E. (2005) 'Aircraft design support using knowledge engineering and optimisation techniques', *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 13th AIAA/ASME/AHS Adaptive Structures Conference*.

Weber, C. (2005) 'WHAT IS “COMPLEXITY”?', *The 15th International Conference on Engineering Design*. Melbourne, Australia.

Verhagen, W. J. C., Bermell-Garcia, P., van Dijk, R. E. C. and Curran, R. (2012) 'A critical review of Knowledge-Based Engineering: An identification of research challenges', *Advanced Engineering Informatics*, 26(1), pp. 5-15.

Whitney, D. E. (2002) *Physical Limits to Modularity*. Massachusetts Institute of Technology. Engineering Systems Division.

Wolfram Research Inc (2015) *Wolfram Mathematica 10.4 Documentation: EdgeBetweennessCentrality*. [Online]. Available at: http://reference.wolfram.com/language/ref/EdgeBetweennessCentrality.html.

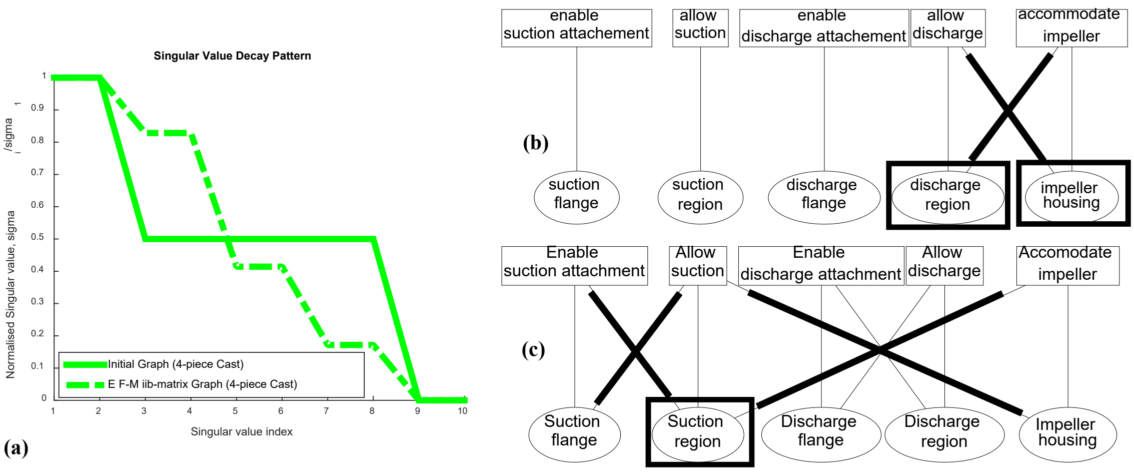
Wyatt, D. F., Wynn, D. C., Jarrett, J. P. and Clarkson, P. J. (2011) 'Supporting product architecture design using computational design synthesis with network structure constraints', *Research in Engineering Design*, 23(1), pp. 17-52.

Yassine, A. A. and Wissmann, L. A. (2007) 'The implications of product architecture on the firm', *Systems Engineering*, 10(2), pp. 118-137.

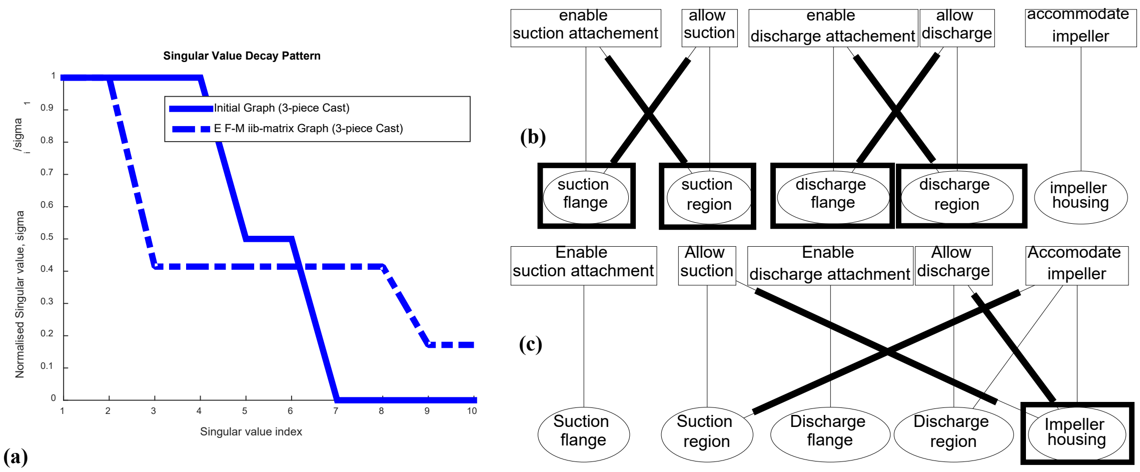
Appendix: Errata to Paper C

Errata to: *Describing and evaluating functionally integrated and manufacturing restricted product architectures*

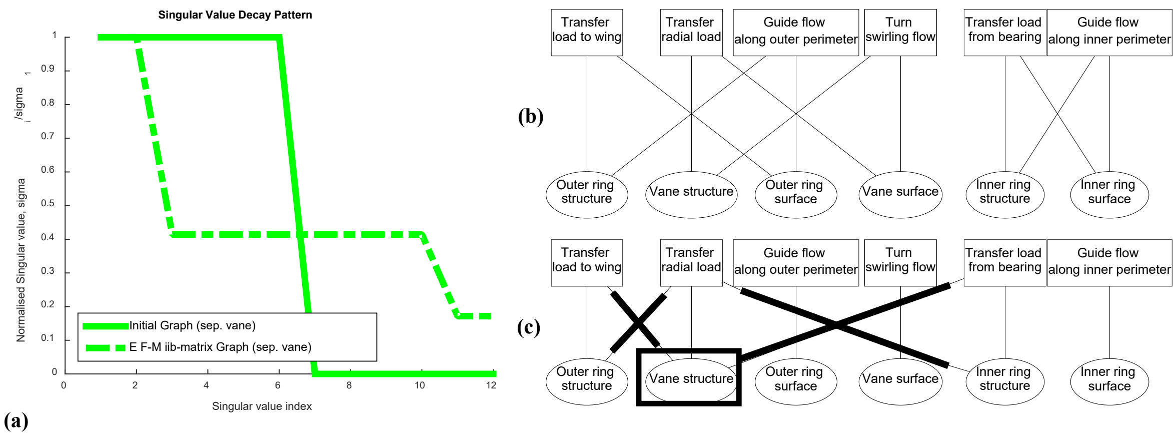
1. *The highlights for edge-betweenness centrality in Figure 18 should be as follows:*



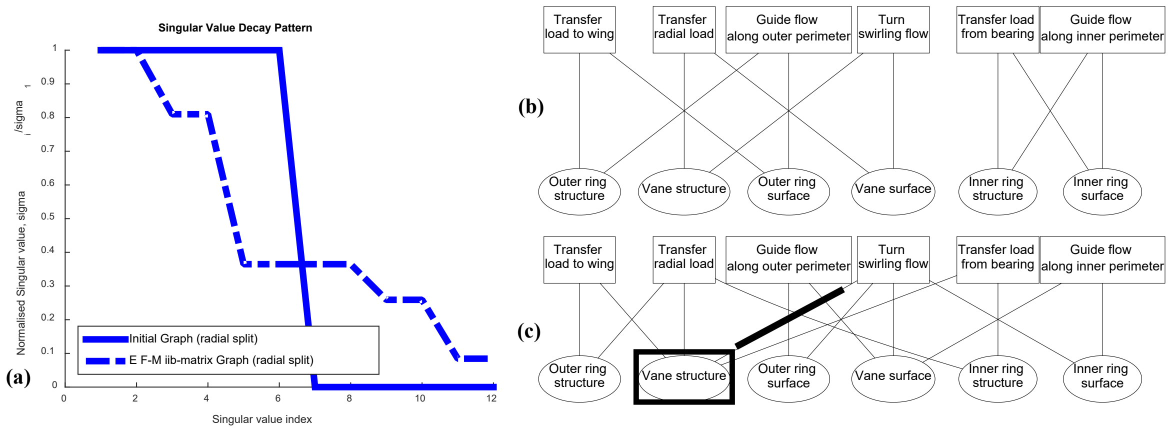
2. *The highlights for edge-betweenness centrality in Figure 19 should be as follows:*



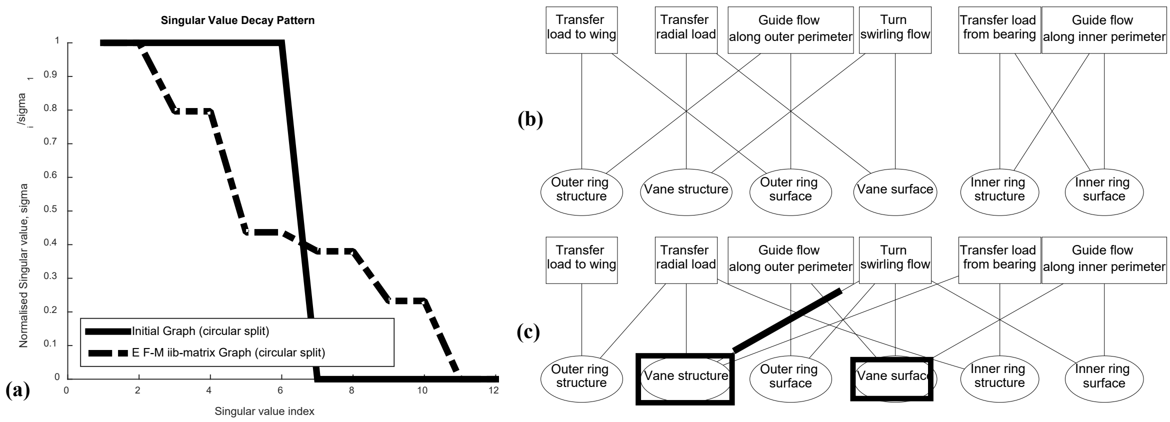
3. *The highlights for edge-betweenness centrality in Figure 23 should be as follows:*



4. *The highlights for edge-betweenness centrality in Figure 24 should be as follows:*



5. *The highlights for edge-betweenness centrality in Figure 25 should be as follows:*



6. *Page 383, the last sentence, last but one paragraph in Section 3.5 should read as:*

However, with the EF-M tree generated graph, where relationships have been manually modified, only three high betweenness centrality cross-relationship are identified.

7. *Page 386, the last but one sentence in Section 4.2 should read as:*

The radial split vane option (Fig. 24b) as well as the circular split vane option (Fig. 25b) have the same GF to GS relationship as the most important, which is the turn swirling flow–vane structure relation.