Variety and Costs - The Effects of Product Variety on Costs and Costing System Accuracy

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation (Monografie)

von

Ole Jan Meßerschmidt

aus

Greifswald

2025

1. Gutachter: Prof. Dr. habil. Matthias Meyer

2. Gutachter: Prof. Dr. Christoph Fuchs

Tag der mündlichen Prüfung: 24.03.2025

Acknowledgements

During the process of my dissertation, which was written at the Institute of Management Accounting and Simulation at the Hamburg University of Technology, I was supported by a number of people.

My greatest thanks go to Prof. Dr. habil. Matthias Meyer, who supervised me during my doctorate. I have good memories about the many discussions in which I was always able to take away new ideas for my dissertation. I would also like to thank Prof. Dr. Christoph Fuchs for taking on the role of second examiner and for the exchange of ideas. Many thanks also to Prof. Dr. Kathrin Fischer for taking over the chairmanship of my doctoral procedure.

Another thanks goes to Dr. Sandra Szech and Dr. Thomas Gumpinger, who gave me the necessary freedom. You made it possible for me to write my dissertation while working full-time as a management consultant at Odego GmbH.

I would also like to thank some of my colleagues at the institute and Odego. In particular, I would like to mention Dr. Kai Mertens, Mark Schmidt, Marcel Bornholdt, Sebastian Achter, Alexandra Eckert, Dr. Jannick Plähn and Lasse Kehrhahn.

Not to be left unmentioned are Larissa, Nina and Florian, who have always motivated me on the way to my doctorate.

My special thanks also go to my family. First and foremost Sarah, who has always had my back. From now on, the weekend is all ours again - I promise.

Summary

Following the microeconomic objective of profit maximization, firms offer their products on heterogeneous markets characterized by individual customer needs, different legal restrictions, competitor products, or cultural factors. Limited in influencing the external complexity, as it is outside their control, firms can decide how to respond by offering a specific product variety. While a large product variety has the advantage of addressing customer needs more precisely, it comes with downsides for firms, such as increased costs, higher lead times, poorer quality, or higher errors in product costing systems. Although research investigated these effects, existing studies are either on a conceptual level or investigate a single cost effect in isolation. Therefore, the latest studies call for operationalizing the various and complicated cause-effect relationships between internal complexity and its economic consequences.

Following these calls, this work investigates the effects of internal complexity on total costs and product costing system accuracy on an operational level. Numerical experiments are chosen as a research method as they allow for creating observations that are difficult to gain in practice, such as the product family design or full information environments. A simulation model for creating product family designs, estimating costs, and combining those designs with various product costing systems is introduced. This model comes with a set of validated measures, operationalizing internal complexity.

Large-scale numerical experiments reveal the moderating effect of component commonality and overdesign on costs. Overdesign and, as a result, the increased component commonality are levers to offer product variety cost-efficiently. However, too much overdesign increases costs as the increased economies of scale cannot compensate for additional material costs. Therefore, overdesign and costs form a U-shaped curve where the apex indicates the optimal degree of overdesign (component commonality). This work further finds that the cause-effect relationships are much more complicated, as suggested by conceptual studies. As a result, local minima exist along the U-shaped cost curve.

A second experiment investigates the effects of internal complexity on product costing accuracy. It is observed that component commonality is not only a lever to reduce costs but also a lever to increase costing system accuracy since it leads to more homogenous resource consumption patterns. A product-level analysis highlights characteristics of products with highly biased product costs under different product costing systems.

This work is placed between engineering design and cost accounting, aiming to strengthen interdisciplinary research. By using numerical experiments, empirical observed cause-effect relationships are generalized. This work is a further step toward an increased understanding of design decisions' economic consequences, which is crucial for decision-making within firms. In practice, these consequences are difficult to observe since cost assignment and cost occurring differ, and the cost advantages of increased commonality mainly affect the indirect costs.

Table of Contents

List of Abbreviations

ABL Anand, Balakrishnan, Labro Framework for Conducting Product Cost System Design Experiments
ACP Activity Cost Pools
CD   Customer Domain
CN   Customer Needs
CO Cost Objects
DCT Discrete Choice Theory
DoE Design of Experiments
DP Design Parameter
EAD Extended Axiomatic Design
FD Functional Domain
FR   Functional Requirement
MADM Multi-Attribute Decision-Making
MC *Mass Customization*
OC Over-Costing
PC Product Costs
PCS Product Costing System
PD Physical Domain
PFCM   Product Family Complexity Measures
PrD Process Domain
PT Production Technology
PV Process Variables
RBV Resource-Based View
RCP Resource Cost Pools
RD Resource Domain
RES_CONS_PAT Resource Consumption Pattern
SC Structural Complexity
SDC System Design Complexity
TC Total Costs
UC Under-Costing

1 Introduction

1.1 Motivation

Following the microeconomic objective of profit maximization, firms aim to maximize revenue and minimize costs (Demski, 2008). In doing so, they offer their products on multiple markets, characterized by individual customer needs, different legal restrictions, competitor products, or cultural factors (Vogel & Lasch, 2016). As a result, they are facing a so-called external complexity. Limited in influencing the external complexity (as it is outside their control), firms can decide on how to respond by offering a certain product variety (Child, Diederichs, & Sanders, Falk-Hayo and Wisniowski, Stefan, 1992; Collinson & Jay, 2012; Götzfried, 2013; Grimm, Schuller, & Wilhelmer, 2014; Maurer, 2017; Piller, 2003). While some firms have a moderate product variety, like Pepsi, with around 3500 individual products (Meinrenken, Sauerhaft, Garvan, & Lackner, 2014), others, such as car manufacturers, offer up to 10^{17} individual product variants for a car model (S. J. Hu, Zhu, Wang, & Koren, 2008). An increased product variety allows for a more specific fulfillment of heterogeneous customer needs. However, it comes with the downside of additional costs as it induces internal complexity which manifests in the product family design, processes, and the organization (Wilson & Perumal, 2010). Reviews by Trattner, Hvam, Forza, and Herbert-Hansen (2019) and Hackl et al. (2020) investigate the economic consequences of the so-called internal complexity on a conceptual level. Empirical studies confirm this picture. For example, a survey among 162 manufacturing firms within the UK reveals that product variety increases costs across the entire firm, such as development, material, manufacturing, or inventory costs (Lyons, Um, & Sharifi, 2020).

Facing this situation, firms search for a trade-off between maximizing product variety and minimizing the cost of providing those products. This dilemma is well known under the term mass customization, which describes reducing costs by gaining the advantages of mass production and increasing sales by individualizing products (Pine & Kotha, 1994). Several methodologies, such as modularity, component commonality, or product platforms (Jacobs & Swink, 2011; Krishnan & Gupta, 2001; Labro, 2004; Wouters, Morales, Grollmuss, & Scheer, 2016), were developed for mastering mass customization.

Besides the effect of internal complexity on costs and operational performance (Hackl, 2022; Trattner et al., 2019), Mertens (2020) notes that it further influences the accuracy of product costing systems. Accurate product costing information are crucial for firm's decision-making as they are used for *"capacity acquisition, planning and pricing decisions"* and *"customer portfolio management, inventory management, and competitive decisions"* (Labro, 2018, p. 307). Balakrishnan, Labro, and Sivaramakrishnan (2012) add that capacity acquisition and allocation are some of firm's information-extensive and complex decision. Internal complexity influences product costing systems' accuracy in several ways. First, an increased product variety and the resulting product family design affect the resource consumption pattern. While homogeneous resource consumption patterns are associated with fewer errors in product costing systems, heterogeneous patterns decrease product costing system accuracy (Balakrishnan, Hansen, & Labro, 2011). Second, increased internal complexity affects the cost structure. An increase in indirect costs, for example, increases the costing error since more costs must be allocated by the product costing system (Labro & Vanhoucke, 2007). Other authors note the effect of non-unit level (fixed) costs on the accuracy of product costing systems (Mertens, 2020; Schmidt, Mertens, & Meyer, 2023).

1.2 Problem Statement and Research Objectives

These challenges are anything but new for both research and firms. For example, Krishnan and Gupta (2001) discuss the power of product platforms to address diverse markets, Labro (2004) summarizes the advantages of component commonality, and Salvador, Forza, and Rungtusanatham (2002) demonstrate how modularity can compensate for the negative impact of product variety. Latest studies summarize the economic consequences of product family design (Hackl et al., 2020; Hackl, 2022), internal complexity (Trattner et al., 2019), or product variety (Lyons et al., 2020; Santos, Sampaio, & Alliprandini, 2020) as noted over the recent years. While these studies provide an overview of the general cause-effect relationships, they are on a conceptual level or verify these effects using a limited number of observations (e.g., a single case). However, this is by no means a criticism of the studies. Rather, it highlights two fundamental problems: a data availability problem and the need for further in-depth understanding. Regarding the first problem, information such as the cost-effective realization of product variety is also relevant for competitors. For this reason, firms do not want to show their hands. Additionally, and related to the second issue, the cause-effect relationships of internal complexity are manifold and difficult to observe (Wilson & Perumal, 2010). Thus, empirical observations are rare, and a deeper understanding of those causal relationships becomes essential. Therefore, recent studies call for an operationalization of internal complexity's economic consequences in their outlook section (Hackl et al., 2020; Lyons et al., 2020; Trattner et al., 2019). For example, Stäblein, Holweg, and Miemczyk (2011, p. 351) state: *"we still have only a limited understanding of how variety impacts on the manufacturing system".* Following these calls, the first research objective is:

RO I: to investigate the economic consequences of internal complexity on an operational level.

With the operationalization of internal complexity, a couple of problems occur since complexity is a latent variable that is not directly measurable. Instead, proxies are used to operationalize internal complexity. However, the literature notes problems regarding the validity and variety of complexity measures, resulting in a scattered state (Hennig, Topcu, & Szajnfarber, 2022). Several authors introduced new measures over recent years (e.g., Ameri, Summers, Mocko, & Porter, 2008; Bashir & Thomson, 2001; Jung, Sinha, & Suh, 2022; Summers & Shah, 2010), increasing the challenge for practitioners and researchers. Practitioners must choose appropriate measures out of a large pool of existing ones. For researchers, various measures hamper comparing existing studies and replication (Hennig et al., 2022), crucial for fast theory development and cumulative science. A second concerns complexity measures' validity, where insufficient validation (Blecker & Abdelkafi, 2006; Sinha & Suh, 2018) and the variety of existing validation approaches (Hennig et al., 2022) are highlighted. Hennig et al. (2022) and Jung et al. (2022) note that valid measures are essential as they enable effective complexity management and support decision-making. Therefore, the second research objective is:

RO II: to summarize existing complexity measures in the context of product design, analyze their similarities, and select a set of validated complexity measures to operationalize internal complexity.

Relatively less discussed but not less relevant is the impact of internal complexity on product costing systems. Product costing systems provide information for various decisions within firms (Balakrishnan et al., 2012; Balakrishnan & Sivaramakrishnan, 2002; Labro, 2018), where more accurate product costs enable to make better-informed decisions. However, Labro and Vanhoucke (2007, p. 940) note that product costing systems are always a *"trade-off between accuracy and the cost of accuracy"*. The impact of biased product costs becomes essential, especially on the product level. Products for which the reported costs are below their true costs are under-costed, and vice-versa for over-costed products (Horngren, Datar, & Rajan, 2012). Firms may drop over-costed products as they seem unprofitable (M. Gupta & Galloway, 2003; D. R. Hansen & Mowen, 2006). On the other end, under-costed products give the illusion of being profitable, although their true costs are above and, in the worst case, even higher than their price tag (Cooper & Kaplan, 1988a; Horngren et al., 2012). Dropping seemingly unprofitable products and being unaware of selling products with negative margins can lead to a vicious circle over multiple periods, as described by accounting and product management literature (V. Anand, Balakrishnan, & Labro, 2017; Baxendale, 2001; Schuh, 2005). Except for the studies of Mertens (2020), neither cost accounting nor product management literature has investigated how internal complexity drives product costing systems' accuracy so far. Although cost accounting has investigated the effect of cost system sophistication, limited information, and measurement errors on accuracy under varying resource demand patterns (V. Anand, Balakrishnan, & Labro, 2019; Balakrishnan et al., 2011; Labro & Vanhoucke, 2007, 2008), these studies are limited in two aspects. First, they assume an activity-based costing (ABC) system. However, empirical studies reveal that only 15-40 % of the firms use such systems in practice (e.g., Al-Omiri & Drury, 2007; Al-Sayed & Dugdale, 2016; Bjørnenak, 1997; Schoute, 2011) while the others rely on simple volume-based systems. Second, these studies do not connect internal complexity with questions of product costing system accuracy[1]. Therefore, the third research objective is:

RO III: to investigate how internal complexity affects the accuracy of product costing systems on the system and product level.

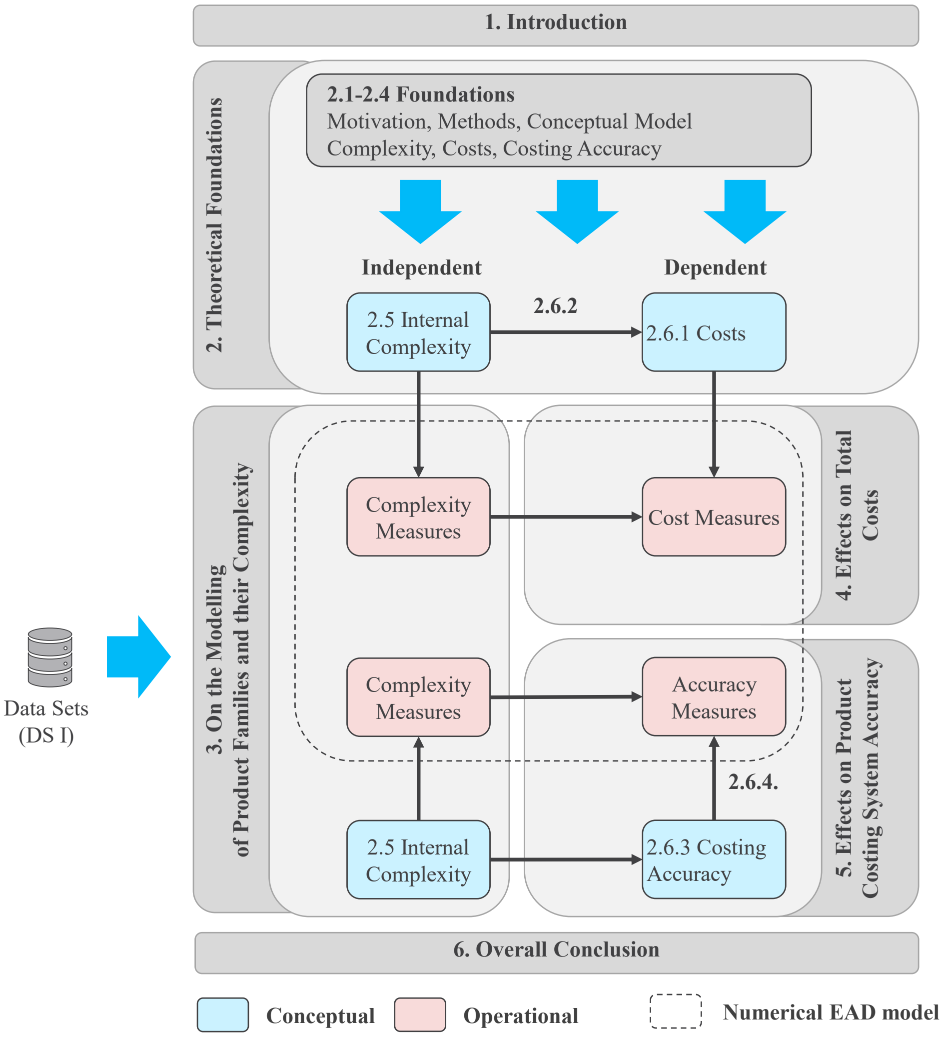
[1] An exception is the work of Mertens (2020)

The research objectives are tackled by means of numerical simulations. Numerical simulations have several advantages in the context if this work. They address the problem of data availability as *"simulation produces its own"* (Harrison, Lin, Carroll, & Carley, 2007, p. 1230), allow for generalization since case studies represents only a specific aspect (Balakrishnan & Penno, 2014), and allow for identify complexity interactions (Axelrod, 1997; Hamada & Wu, 2009).

1.3 Structure of this Thesis

This work is separated into four parts, where Figure 1.1 provides an overview. According to Libby's (1981) predictive validity framework, the work differs between a conceptual and operational layer. Section 2 focuses on the conceptual layer and discusses the theoretical foundations of product variety, complexity, and numerical experiments. It further sheds light on the reasons why firms face a certain complexity, its manifestation within firms, and the economic consequences of complexity. Each of the following three sections (3 - 5) is related to one research objective, as noted in the previous section. Section 3 (research objective RO II) operationalized complexity in the context of product family designs. This section also introduces the numerical EAD framework. A large-scale numerical experiment analyzes similarities among complexity measures, and a set of complexity measures is suggested. Related to the first objective (RO I), section 4 investigates the economic consequences of product family design complexity. In doing so, models reported by the literature are integrated into the numerical EAD framework. Numerical experiments allow for investigating the cost effects of product family design changes and increased product variety. Addressing objective three (RO III), section 5 analyzes the effects of internal complexity on product costing system accuracy. In this section, the product costing system model, introduced by V. Anand et al. (2019), is integrated into the numerical EAD model. Finally, this thesis ends with an overall conclusion, summarizing the main findings of this work (section 6).

Figure 1.1:
Structure of this work



2 Theoretical Foundations

This chapter introduces the theoretical foundations used in later sections of this thesis. It starts with a brief history of product variety (section 2.1), followed by an introduction to the basics of complexity (section 2.2) and numerical experiments (section 2.3). A connection between product variety and complexity is made in the remainder of this chapter. Section 2.4 introduces different views on firms, which are a pre-condition to show how complexity generally affects firms (section 2.5). The economic consequences of product variety and the resulting complexity are discussed in section 2.6.

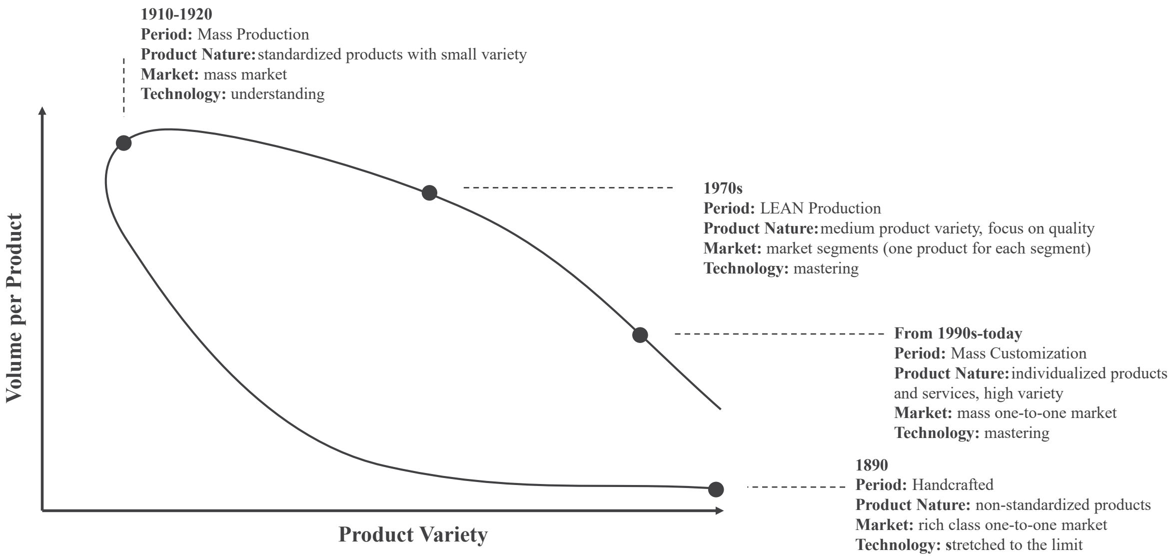
2.1 A Brief History of Product Variety

The story of product variety starts during industrialization in the 19th and early 20th centuries. Products beyond the daily demand become available for a broad market since manufacturing costs dramatically dropped. A downside was that the amount of product variations was limited as existing production technologies were not flexible enough. As markets get saturated, customers ask for a more individualized fulfillment of their needs, and firms shift from mass-production to mass-customization, resulting in a renaissance of product variety (Chinnaiah & Kamarthi, 2000). Aiming to increase customer satisfaction and addressing new markets are still the product variety-driving forces today. The car's history is a well-documented example of this transformation from craft production via mass production to mass customization.

Womack (1990) investigates how the car changed from the end of the 19th century until the end of the 20th century (see Figure 2.1). In the early days, cars were handcrafted and fitted individually to each customer. This resulted in many individual products since each customer had individual needs and a low overall quantity since only wealthy people could buy a car (Chinnaiah & Kamarthi, 2000). The upcoming mass production, strongly driven by Henry Ford’s Model T in the 1910s, dropped the costs for a single car. From now on, more people could buy cars. However, Ford had to dramatically reduce the number of different products to gain economies of scale and reduce diseconomies of scope. Therefore, the Ford Model T was available in only nine different variants. During those 30 years, the variety of cars reduced and number of cars increased. Henry Ford had the idea to offer standardized products at low costs. It turned out that this idea was successful until other companies entered the market and adopted the mass production principle. In the 1930s, Alfred Sloan, the chairman of General Motors at this time, came up with the idea of what is called sloanism today (Medeiros, 2000). His strategy was to address customer preferences more individually by introducing different engines, interiors or other features for the same car to gain competitive advantage (Rothschild, 1974). The renaissance of product variety, as shown in Figure 2.1, began

(Womack, 1990). Improved mass production capabilities through principles like LEAN in the 1970s led to further cost drops and, therefore, product prices. An even broader market was now available where customers had individual needs. As a result, product variety was further increased (Chinnaiah & Kamarthi, 2000; Womack, 1990).

Figure 2.1:
Renaissance of product variety, according to Womack (1990, p. 126) and Chinnaiah and Kamarthi (2000).



Gaining the advantages of both mass production (economies of scale) and individualization (satisfying more markets) leads to what Pine and Kotha (1994) call mass customization (MC) . It describes the production of highly individualized products or services offered on a mass one-to-one market (Chinnaiah & Kamarthi, 2000). Today, car manufacturing companies deal with a product variety of 10^{17} and more individual product variants for their cars (S. J. Hu et al., 2008). The numbers show that there are hypothetical more product variants for a single car model available than people on the planet (approximately 8 $*10^{9}$). Even if each person buys a car, there are still product variants that are never sold. How does this development continue? Fuchs and Golenhofen (2019) argue that customers' power compels companies to offer even more individualized products at low prices. A further development, therefore, is characterized by a collaboration of customers and firms during the product design process to fit products even more to individual customer needs (Fuchs & Golenhofen, 2019, p. 123). However, the renaissance of product variety is not limited to the automotive industry. Fuchs and Golenhofen (2019, p. 53) note that at the beginning of a technology lifecycle, such new technologies are stretched *"to their limits to reach the performance needed to satisfy minimum customer requirements"*. At this stage, product performance is more important than price or convenience, and customers are willing to make trade-offs (Fuchs & Golenhofen, 2019). Inventing new technologies results in products handcrafted by highly specialized companies. Since the price gets increasingly important, companies are forced to switch to mass production later in the technology lifecycle. If firms manage mass production, they aim to increase product variety to match the individual customer needs more precisely. While this cycle took decades for the car, increasing innovation speed forces firms to go through such steps much faster nowadays.

It seems that mass customization is the final goal for companies to succeed. However, it has many challenges for firms, customers, and suppliers. Even in a mass customization environment, not all customer requirements are fully fulfilled as the customization process is constrained by physical, technological, and economic limits (M. Zhang & Tseng, 2007). While physical limits represent an insurmountable wall, technological limits can be pushed by additional research. All that remains are the economic constraints. There is a consensus in the literature that there is a positive association between product variety and costs widely known as complexity costs (Blecker & Abdelkafi, 2006; Hackl et al., 2020; Lyons et al., 2020; Roy, Evans, Low, & Williams, 2011; Trattner et al., 2019; M. Zhang & Tseng, 2007). For example, producing individualized products at a compatible price requires more setup changes in production sophisticated logistical processes to ensure that the right colored door is at the right time at the right product line to get mounted to the frame. In less automated environments, like job shops, increasing variety reduces the chance to benefit from learning curve effects caused by task repetition. However, challenges of increasing product variety and the resulting complexity within firms are not limited to production. All departments, like engineering, sales, logistics, and after-sales, are affected by the increasing complexity due to product variety within a firm (Hackl et al., 2020; Lyons et al., 2020). Section 2.5 takes a closer look on the reasons for the complexity, firms face. The economic consequences are discussed in section 2.6. The following three sections introduce three foundations used in this work's further proceeding.

2.2 A Definition of Complexity

Complexity has its origin in the Latin word 'cumplectere'. Cumplectere means multi-layered, twisted nature (Efatmaneshnik & Ryan, 2016) and describes the interweaving of characteristics. Beyond this abstract definition, literature shows a variety of definitions for complexity across different disciplines (Jacobs & Swink, 2011). Page (2010, p. 24) notes that the variety of definitions represents *"less a lack of agreement than an inability of any single approach to capture what scientists mean by complex"*. Analyzing studies across twelve different research fields, Jacobs and Swink (2011) find that a system's complexity is characterized multiplicity, diversity, and interrelatedness. Although their work focuses primarily on complexity in the context of product families, definitions across other fields show similar characteristics (e.g., N. F. Johnson, 2007; Simon, 1962), indicating that these characteristics are also valid for other fields. Often, complexity is associated with the presence of an underlying system (systemic context), where literature defines a system as a set of elements with certain relations among each other (Bertalanffy, 1979; J. G. Miller, 1965). Multiplicity refers to the number of elements where a system of six elements is more complex than a system containing four. Diversity describes the differences or dissimilarity between elements. A set of elements containing different properties like shape or color is more complex than a homogenous set where all elements have the same properties. The third dimension, interrelatedness, refers to the relations among elements. Adding more relations to a system while keeping the number of elements and their differences constant will increase its complexity. Figure 2.2 visualizes these dimensions.

Figure 2.2:
The three dimensions of complexity are represented in a three-dimensional space. Total complexity is a result of elements multiplicity, diversity, and interrelatedness.

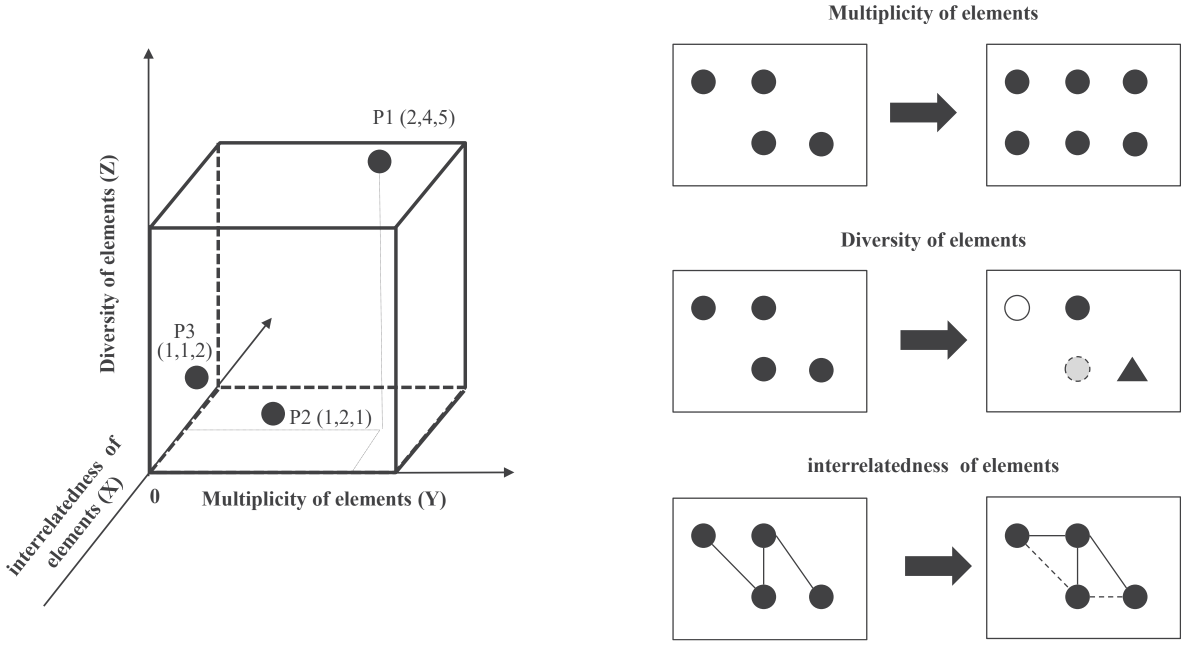


Figure 2.2 visualizes complexity using a cube where each axis represents one dimension. Let D be a generic distance metric, then the distance of any system P1-P3 from the origin (O) is used as a proxy for systems' complexity. However, many authors point out that *"complexity is in the eye of the beholder"* (Meijer, 2006, p. 11); *"what is complex to someone might not be complex to another"* (Vogel & Lasch, 2016, p. 2), or, *"a system is complex when it is composed of a group of related units (subsystems), for which the degree and nature of the relationships is imperfectly known"* (Sussman, 1999, p. 14). These statements highlight a subjective and objective view of complexity (Efatmaneshnik & Ryan, 2016; Gell-Mann, 1995; Keuper, 2004). While objective complexity is independent of a subject and its knowledge, subjective complexity depends on the observer and measures complexity relative to a so-called *"reference simplicity "* (Efatmaneshnik & Ryan, 2016, p. 535). Subjective complexity is illustrated using an example by Foerster (1978) and replicated in Brosch (2014, p. 9). Two sequences of numbers are given:

1. {1,2,3,4,5,6,7,8}
2. {8,5,4,1,7,6,3,2}

While the first sequence looks intuitive, the second one looks arbitrary. However, once the observer gets additional knowledge, it becomes easy to see that the numbers are ordered in alphabetical order of their words (Brosch, 2014). This example illustrates that subjective complexity depends on observers' context (pre-knowledge) and differs across individuals. Efatmaneshnik and Ryan (2016) note that each observer has a subjective simplicity. This is visualized by different origins for each observer in Figure 2.2. On the other hand, objective complexity is independent of observers' context as it estimates the degree of complexity based on a given set of rules. Such rules, for example, define a reference simplicity and measure to differentiate any given system from another (Efatmaneshnik & Ryan, 2016). Based on the example from Figure 2.2, the origin (O) is an example of a reference simplicity. It defines the simplest possible system without any elements ($Y = 0$), any relations ($X = 0$), and with no differences across elements ($Z = 0$). Measures further define rules for measuring system differences from the reference simplicity. In Figure 2.2, the Euclidean distance $d(P, O)$ between a system (P) and the origin (O) is an example of such rules. Based on the given reference simplicity, each observer would rate $P1$ as the most complex system ($d_{P1} = 6.7$) followed by $P2$ and $P3$ as equal complex ($d_{P2} = d_{P3} = 2.45$). Since the rules allow to reproduce the results shown in this work independently from readers' subjective simplicity, this work focuses on objective complexity, denoted as complexity hereafter.

Complex systems have additional properties that are less dominant or absent in simple systems. Sheard and Mostashari (2009) provide an overview of the characteristics of complex systems. As an atomic layer, elements define the smallest unit of a system. Elements are grouped into sub-systems on higher hierarchical layers, leading to the final system. Therefore, Sheard and Mostashari (2009) define hierarchical ordering as the first property of complex systems that allows a decomposition or recompositing into smaller/higher structures. Maier (1998) notes that complex systems are to be understood less as a singular construct but rather as a system-of-systems. The interaction of elements on different layers of hierarchy within a system leads to an emergent behavior of complex systems, where elements or sub-systems follow a super additive composition rule (Foerster, 1962; J. G. Miller, 1965). Emergence leads to the perceived phenomenon that a complex system is "*more than the sum of parts*" (Bertalanffy, 1979, p. 55; Simon, 1962, p. 468) or "*more than the sum of [its] characteristics*" (J. G. Miller, 1965, p. 217). N. F. Johnson (2007) further adds that emergence is also caused by the open character of complex systems and the non-trivial interactions among elements. Feedback loops are an example of such interactions[2] and a system's time dependency. Due to these loops, interactions of earlier phases may still oscillate within the system and influence the system's behavior in the present (N. F. Johnson, 2007). The openness describes the interaction of complex systems' elements with their environment and further leads to fuzzy boundaries (Sheard & Mostashari, 2009) as there may exist no clear system boundary and complicates the definition of a closed system, which is essential for most physical theories (N. F. Johnson, 2007).

To sum up, complex systems are characterized by their elements' multiplicity, interrelatedness, and diversity. They have several properties leading to a complicated behavior that is difficult to predict (N. F. Johnson, 2007). To tackle questions involving complex systems, an appropriate method is required. Numerical Experiments are such a method and introduced in the following section.

2.3 Numerical Experiments to Analyze Complex Systems

Since systems interact with their environment, they react to a given input with a specific output, which observers notice as an occurring phenomenon. Driven by human curiosity and

[2] N. F. Johnson (2007, p. 14) notes: „*the system appears to be alive*".

manifested in scientific progress across various fields, the necessity to explain these phenomena emerges. Among others, Roethlisberger (1977) introduces the 'Knowledge Enterprise' to describe the research and knowledge development process. The work of Kaplan (1986) applies this concept to management accounting research. From a methodological perspective, it starts with knowing the phenomenon's acquaintance, characterized by observers' skill, as it allows a general understanding. Followed by the level of clinical knowledge, phenomena are classified and described via observations or interviews. Kaplan (1986, p. 434) notes, at this stage, *"definitions are non-operational, measurements are not objective, and hypotheses are actually questions of the form, 'What is going on here?'".* After the description, researchers have sufficient understanding to measure the size of phenomena coded in variables and enter the level of analytical knowledge (Kaplan, 1986). The phase of analytical knowledge starts by formulating assumptions on cause-effect relationships. These relationships are discovered and verified in the next step by operationalizing the concept via measures and testing the hypotheses via an experimental design and statistical techniques (Kaplan, 1986). Cause-effect relationships are essential as they allow them to make predictions (Kaplan, 1986) and rule out alternative explanations (Luft & Shields, 2014). Knowledge Enterprises' summit is reached when a smaller set of general propositions allows an explanation of verified empirical hypotheses (Kaplan, 1986). According to the Knowledge enterprise, knowledge development is separated into a theoretical and empirical research.

Consequences of assumptions are deducted a priori in theoretical research. Analytical models used in theoretical research contain assumptions, often expressed as mathematical formulations that represent tautologies[3] (Balakrishnan & Penno, 2014). A major limitation of analytical models is that mathematical formulations must be tractable (Labro, 2015b). There exist several models for which analytical solutions do not (yet) exist, such as the Navier-Strokes equations to describe the flow of Newton fluids (Rogers, 1992) or where solutions exist only for special cases such as general relativity under the assumption of spherical symmetry (Harrison et al., 2007). Balakrishnan and Penno (2014) add that analytical models are also limited in the number of factors and variables to keep model complexity manageable. Alternatively, empirical analyses represent the induction where observational data is analyzed to identify variable relationships. In line with the concept of the Knowledge Enterprise, part of inductive research is the test of assumptions derived by deduction (Harrison et al., 2007). However, a critical aspect of empirical research is the availability of data. In empirical research, variable measurement can be challenging, resulting in high costs or the necessity to create a construct representing the objective (e.g., via a latent variable model). Archival studies, for example, contain observations that lie in the past, and the subsequent measurement of additional variables can be a challenging task. In experimental studies, researchers face the trade-off between increasing the number of variables and the sample size. However, both activities are associated with increasing efforts for researchers and participants.

---

[3] Balakrishnan and Penno (2014, p. 531) define a tautology as *„an argument whose conclusion follows logically from its premises."*

A third pillar are numerical experiments, which lie in between the deduction of theoretical research and induction of empirical research (Harrison et al., 2007). They provide numerical answers to socio-technical problems when other methods (e.g, archival, field studies, analytical, …) cannot be used, are not suitable to use, or fail (Carley, 2002; Vahl Davis, 1986). Body's deformation under an applied force in engineering is an example for such a problem. Analytical solutions are suitable for small deformations where stress and displacement are linear, however, they fail for large displacements. Finite element models solve this problem as they divide the body into smaller parts (via meshing) for which a solution of stress and displacement can be calculated more easily and then aggregate to the initial body. As the example shows, numerical research extends the deduction as it overcomes the limitation of tractable mathematical relationships and allows to investigate problems where no known (exact) analytical solution exists yet (Law, 2015). Additionally, numerical simulations solve the problem of data availability *"since a simulation produces its own"* (Harrison et al., 2007, p. 1230). Balakrishnan and Penno (2014) note that a hand full of examples such as derived from empirical field studies might be powerful as it illustrates a specific aspect and support the understanding of analytical insights for a given phenomenon. However, examples are only able to show conditions under which a causal relation does not hold. Monte Carlo simulations represent the other end where the models' output is calculated for several combination of input variables. Numerical experiments are within these bounds where the almost inexhaustible number of observations enables to identification of models' behavior under a large range of variable's input values and variable's interactions (Axelrod, 1997; Hamada & Wu, 2009; Kleijnen, 1998; Lorscheid, Heine, & Meyer, 2012). Empirical research may oversee these interactions as the necessary combination of inputs did not occur in the data set (Balakrishnan & Penno, 2014). The data availability in numerical experiments allow researchers to identify causal relations, and draw generalized predictions under limited information as more cases are present from which overarching rules can be derived (Balakrishnan & Penno, 2014; Kaplan, 1986). Weather models are an example, for the power of numerical experiments. As argues, fluids' flow is described by Navier-Stroke equations where no analytics solutions exist. Additionally, atmospheres' current state is not fully known. Numerical experiments, however, allow to make weather predictions as the calculate run the model several times with slightly different conditions resulting in a certain probability for the weather within the next days. Lastly, Law (2015) notes that numerical experiments are often more cost-effective compared to real world experiments indicating another advantage.

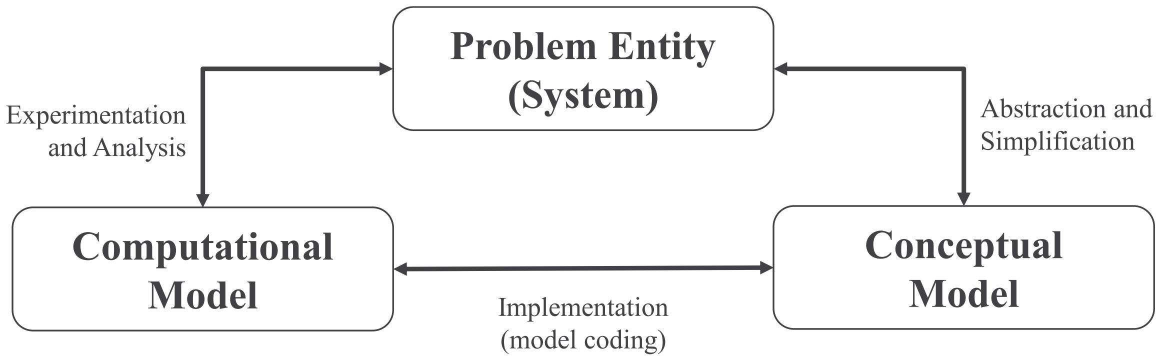
Although data availability is a great strength of numerical experiments, it can also become problematic if improperly handled. Generating many observations is time-consuming, results in computational costs, and is not always needed. Some variable combinations can be dropped, for example, due to physical reasons, pre-knowledge, or existing lower or upper bounds. A methodology is required to structure the range and combination of input parameters. The Design of Experiments (DoE) is a method that defines input variables, their steps, boundaries, and combinations. It describes *"the process of planning, designing and analyzing the experiment so that valid and objective conclusions can be drawn effectively and efficiently"* (Antony, 2014, pp. 8–9). Randomization is a vital principle of the DOE as it allows the inclusion of uncertainty in the experiment by observing models' response under unknown conditions (like for the weather model mentioned above). Due to economic or technical constraints, not all input variables are always measurable and, therefore, uncontrollable (such as the atmospheric pressure for any given point on earth). Randomization allows for averaging uncontrollable factors, removing their influence on the output (Antony, 2014). It is important to note that a DoE is also a method for empirical or archival research since it only defines variable ranges and their combinations for efficient and effective measurement. However, the limited number of observations in empirical studies caused by higher data acquisition costs and the limitation of archival research that some input combinations may be absent result in smaller averaging effects. Additionally, a DoE is a crucial aspect of replication as it provides an overview of the experimental setting and allows for investigating systems robustness, optimizing systems response, or screening variables (Hamada & Wu, 2009; Lorscheid et al., 2012). The latter is essential for cumulative research. Variable screening provides pre-knowledge of variables' impact and, therefore, reduces efforts, for example, in field experiments as the number of measures can be limited to the relevant ones (Balakrishnan & Penno, 2014), and regions of interest can be measured more granular (Harrison et al., 2007).

Combined with a DoE, numerical experiments have several advantages, making them an excellent choice to investigate complex systems[4]. Complex systems generally are characterized by a non-trivial behavior due to overlapping effects, emergent behavior, and an open character. Numerical experiments allow for simulating systems' responses under various input settings. The open character and the corresponding fuzzy boundaries induce uncertainty, as some variables are uncontrollable or measurable, such as market response for a given product mix or the true product costs (see section 2.6.4). Variable interactions, for example, are identified when potentially overlapping input variables are varied independently from each other. Numerical experiments are beneficial when empirical data is difficult to obtain. Usually, firms hold product family design, the corresponding supply chain, and cost information confidential, limiting the possibility of collecting empirical data. There still exists case studies analyzing product design (G. Kim, Kwon, Suh, & Ahn, 2016; Robertson & Ulrich, 1998; Sawai, Nomaguchi, & Fujita, 2017) or cost system design (Al-Omiri & Drury, 2007; Brierley, Cowton, & Drury, 2001), however, they are limited in their number of observations which makes it difficult to generalize results. In distinction to analytical research, numerical experiments provide solutions for complicated analytical models such as product cost system design (Balakrishnan et al., 2011) or the impact of cost information on firms decision making (V. Anand et al., 2017). Despite all the advantages, there are some disadvantages of numerical experiments. Vahl Davis (1986) notes that numerical models are often more abstract, for example, compared to single examples, and require mathematical understanding. However, the advantages outweigh the disadvantages of numerical experiments concerning the challenges presented in this work. In doing so, this work follows the argumentation of Harrison et al. (2007, p. 1229), which state, *"well suited for the study of complex behavioral systems, simulations show greatest utility for gaining theoretical insight through developing theories and exploring their consequences."*

[4] See also the discussion by Harrison et al. (2007)

The problem entity is defined at the beginning of each research project regardless of inductive, deductive, or numerical research. Depending on the problem entity, experiments are performed on the actual system itself or a model (a replica) of the system (Law, 2015). Models are further divided into physical and mathematical models. Physical models have a physical representation of the system, such as a car in a wind tunnel. Mathematical models are described by logical and quantitative relationships between variables (Law, 2015). They are also called formal (Harrison et al., 2007) or conceptual models, representing a simplified version of the real system and are independent of any computer code or software (Robinson, 2008). Via model coding, conceptual models are transformed into computational models representing a specific manifestation of a conceptual model. Experimentation using computational models, finally, provides solutions for an initial problem or allows a better understanding of the initial problem (Robinson, 2008). Figure 2.3 summarizes the dependencies of the problem entity with conceptual and computational models. It becomes clear that these elements form a cycle in which the results of computational models provide additional knowledge on the problem, which, again, influences the conceptual model. Robinson's (2008) life cycle shows the individual stages research needs to go through in numerical research. This work makes no exception, and sections 2.4.1 - 2.4.3 introduce the problem entity and section 2.4.4 the conceptual model. The basic computational model is introduced in section 3.5 and further extended in sections 4.3 and 5.3.1.

Figure 2.3
Mathematical model development cycle according to Robinson (2008), adapted from Mertens (2020, p. 7)



2.4 The Firm System

As argued in the previous chapter, model development starts with describing the problem entity. In doing so, the firm as the thesis's protagonist is introduced from different perspectives. Starting with economic firm theory (section 2.4.1), via the microeconomic view (section 2.4.2), and ending with an engineering perspective in section 2.4.3. These views are unified in section 2.4.4 where Mertens (2020) conceptual Extended Axiomatic Design (EAD) model is introduced.

2.4.1 Economic Firm Theory

The Economic firm theory is a broad field where even a summarizing handbook of the firm by Dietrich and Krafft (2012) contains nearly 600 pages. Therefore, this section focuses on three main views as their concepts become relevant later in this work. Specifically, this section starts with the Marshallian perspective, followed by Nelson and Winter's (1982) evolutionary theory, and ends with the modern resource-based view (RBV), introduced at the end of this section.

The neoclassical economist Alfred Marshall (1842-1924) defines a firm as a construct consisting of an internal organization, external trade connections and a plant (Marshall, 1920; Ravix, 2012). Firms trade their goods on markets where the firm's supply meets buyer's demand. They aim "*to obtain […] the greatest net excess of receipts over […] 'expenses' or 'money cost' of production; with due allowance for his own trouble and risk, and for the use of his capital*" (Marshall, 1920, p. 127). Facing this objective, a firm can increase product selling prices, push their demand, or decrease internal efforts. Demand and market prices, however, are related as an increase in prices will result in lower demand according to the concept of elasticity (Marshall, 1920). Besides these effects, market prices are also affected by exogenous conditions, which result in a continuous variation of demand. As Marshall's theories came up at the end of the beginning of the 20th century, he used fish sold on a market as an example of exogenous market factors. At this time, only a few households had fridges. Therefore, the weather was an exogenous market factor since most buyers could not cool the fish on hot summer days (Marshall, 1920). A more actual is the COVID pandemic. Due to personal safety or governmental restrictions, the flight demand decreased by 50% in 2020 compared to 2019 (Hasegawa, Chen, & Duong, 2021).

According to Marshall, firms differ as they have different internal organizations, trade connections, and dynamics. A better internal organization and trade connections allow firms to produce products at lower costs, resulting in more products sold, assuming price elasticity. Trade connections put them into a position to sell their products with less effort, giving a competitive advantage since customers and producers know and trust each other (Marshall, 1920). Trades are fulfilled on the market, where a firm's supply answers individual entities' demands (Marshall, 1920, p. 126). Market entities are a proxy for natural persona - like in business-to-consumer trades - or other firms - like in business-to-business trades. While classical economists like Adam Smith argued that increased productivity is created solely by labor division, Marshall states that better internal organization in terms of knowledge and the resulting division of labor affects firms' productivity (Loasby, 1998, p. 142). Later, Veblen (1857-1929) extended this view by adding that firms hold a certain set of routines and knowledge resulting from experimentation and experience, providing a competitive advantage. His ideas of organizing knowledge and capabilities set the foundation for the evolutionary theory (Becker & Knudsen, 2012; Hodgson, 2012).

Influenced by these streams, Nelson and Winter (1982) introduce the evolutionary theory of the firm. The theory explains the long-run phenomena of firms' evolution in terms of technology and organization (Becker & Knudsen, 2012; Nelson, 1991; Nelson & Winter, 1982). *"Develop, screen, and selectively adopt new and better ways of doing things"* (Nelson, 1991, p. 66) are the driving forces in the long run. The theory defines firms as a set of routines that, in conjunction with environmental conditions, describe the actual behavior of the firm. From an evolutionary perspective, routines are what genes are to an organism (Nelson & Winter, 1982). Following this analogy, firms undergo an evolutionary process regarding technology and organization. According to this theory, innovation forces competitors to imitate the innovator's routines as they define firms' internal behavior. Therefore, firms with a similar set of routines are in a better position for imitation than those with a more different set as they must perform less adaption. Markets, among other external factors, take the position of natural selection as they decide which set of routines (genes) are successful, or according to Nelson (1991, p. 69), "*some of the innovations will be winners, other losers." F*irms' technological and organizational evolution are interwoven. Both require each other as the organizational evolution, such as in R&D departments, enables firms to focus part of their workforce on innovation in terms of new products and processes instead of solving daily problems (Hounshell, 1989; Reich, 1985) and on the other side, innovation requires new ways of organization.

The resource-based view (RBV) combines several aspects of previous theories, such as the evolutionary theory, the knowledge-based view, and the capability-based view (Foss & Stieglitz, 2012). The traditional interpretation of the RBV is founded on the work of Penrose (1959). Her theory contributes, among other things, to today's knowledge of creating and sustaining competitive advantage (Kor & Mahoney, 2004). The RBV is founded on the belief that economic value is created by effective and innovative usage of resources rather than resource possession (Kor & Mahoney, 2004; Mahoney, 1995). Brand names, in-house knowledge of technology, employment of skilled personnel, trade contacts, machinery, efficient procedures, and capital are examples of resources (Wernerfelt, 1984). Resources are sourced on strategic factor markets (Barney, 1986), whereas products are sold on product markets (Foss & Stieglitz, 2012). Therefore, firms transform input resources into output goods, which are then sold on product markets. At a certain point, firms hold a set of resources (tangible and intangible assets) that are temporarily tied to the firm (Caves, 1980; Wernerfelt, 1984). From a short-run perspective, resources are tied to the firm. In the long run, firms can control their resources in terms of type and quantity. Manufacturing machines are an example of resources tied in the short run but replaced in the long run. Empirical studies, for example, show that the average lifetime of machines in production is around 26 years (Erumban, 2008). A vital concept of the RBV is its internal resource accumulation (Dierickx & Cool, 1989; Foss & Stieglitz, 2012), which states that firms gain a competitive advantage if they use their resources in order to evolve a value-creating strategy which is hard to adapt by competitors. As long as competitors are not able to adopt the strategy itself or, at least, the strategy's benefits, a firm can defeat its competitive advantage (Barney, 1991). Lippman and Rumelt (2003, p. 1082) summarize that firms should focus on developing *"complex home-grown resources"* to gain competitive advantages since resources from factor markets are also available for competitors. This interpretation suggests that firms create value from the inside through managerial capabilities and technical talents (Kor & Mahoney, 2004). Managers take an organizational role, and technical talents play the executive role to ensure an effective and innovative resource allocation as well as the combination and to transform resources either in new firm capabilities (internal) or new products (external, market) (Kor & Mahoney, 2004). The modern interpretation of RBV further adds a dynamic aspect to learning within firms. In doing so, aspects of the evolutionary theory by Nelson and Winter (1982), the knowledge-based theory by Kogut and Zander (1992), and the (dynamic) capability view by Teece, Pisano, and Shuen (1997) are integrated. In contrast to the traditional view, competitive advantage is not only a result of the value-creating strategy driven by resource organization but also a result of a firm's dynamic capabilities (Foss & Stieglitz, 2012) such as changing its set of resources "*to address rapidly changing environments*" (Teece et al., 1997, p. 516). Current firms' dynamic capabilities reflect a crucial asset as they result from a previous learning process on generating and modifying routines for improving performance and, therefore, competitive advantage (Foss & Stieglitz, 2012).

2.4.2 A Microeconomic Perspective on Firms

While the RBV provides a rather abstract understanding of the firm, the microeconomic view provides an analytical explanation. For this work, it builds the bridge between the economic and engineering perspectives (section 2.4.3) on firms. Microeconomics distinguishes between single- and multi-product firms, where a multi-product firm produces more than one good or service for a market (Demski, 2008). Since this work focuses primarily on good-producing companies rather than service-producing ones with a variety of individual products, they are multi-product firms (hereafter: firm). From an analytical perspective[5], firms are the link between factor (resource) markets and output (product) markets. This middle position of firms allows to treat them as a function that transforms inputs ($z \in \mathbb{R}^{N_{RD}}$) into products (outputs) at a certain quantity ($q \in \mathbb{R}^{N_{PROD}}$) via a production function:

$$q = f(z) \tag{2.1}$$

N_{PROD} refers to the number of unique output goods and N_{RD} refers to the number of inputs. Assuming perfect markets, customer's requested product demand[6] ($\hat{q}$) is fulfilled by firms' product quantities (q). Perfect markets assume that $\hat{q} = q$ for all periods, which implies that no stock is required as all products are sold (V. Anand et al., 2017; Balakrishnan & Sivaramakrishnan, 2002). On output markets, the firm gets the sum of their products' output prices $\hat{P}$ weighted with their sold quantity ($\hat{q}$) from customers, which leads to the revenue (R) defined as:

$$\boldsymbol{R} = \sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N_{PROD}}} \boldsymbol{\hat{P}_i} * \boldsymbol{\hat{q}_i} \tag{2.2}$$

To produce the requested quantity of products, firms appear as customers on factor markets to source inputs (resources) at certain prices (P). Firms' total costs TC (TC) are defined as the resource unit prices multiplied by input's quantity ($z_i, 1 = 1 \ldots N_{RD}$):

$$TC = \sum_{i=1}^{N_{RD}} P_i * z_i \tag{2.3}$$

[5] The analytical model relies primarily on the work of Demski (2008) and Christensen and Hemmer (2006)

[6] Perfect markets mean that there is no uncertainty on product demand and prices (transparency), no reaction time for the firm (e.g., no production time), and stable customer preferences over time.

According to the economic objective of profit maximization, the following objective function completes the basic firm definition.[7]

$$\max: R - TC = \max: \sum_{i=1}^{N_{PROD}} \hat{P}_i * \hat{q}_i - \sum_{i=1}^{N_{RD}} P_i * z_i \qquad (2.4)$$

subjected to:

$$q, z, P, \hat{P} \geq 0 \qquad (2.5)$$

A production function defines the required inputs and their quantity for each product. In doing so, it links the cost function with resource quantities (Sickles & Zelenyuk, 2019). Several different production functions (e.g., Leontief, Cobb-Douglas, or the translog model) exist where the Leontief model "*dominated the empirical productivity literature*" (Sickles & Zelenyuk, 2019, p. 188) and is used in current cost accounting research (V. Anand et al., 2019; Balakrishnan et al., 2011; Mertens, 2020; Schmidt et al., 2023). It assumes a linear relation between input and output and neglects product interactions (Christensen & Hemmer, 2006). Each output is defined as a set of inputs with a fixed proportion (Sickles & Zelenyuk, 2019), and substitution of inputs is not allowed (Labro, 2018). If a linear model is assumed, the production function turns into a matrix called resource consumption pattern (RES_CONS_PAT) , representing the constraints during the transformation of inputs into outputs (Christensen & Hemmer, 2006). According to recent studies (V. Anand et al., 2019; Balakrishnan et al., 2011), the variable product costs (PC) are defined as:

$$PC = RES\_CONS\_PAT * RCU \qquad (2.6)$$

where, RCU is the vector of resource costs per unit. Multiplication of product costs with its produced quantity (q) returns the total costs as:

$$TC = PC * q \qquad (2.7)$$

The following example provides a visual representation of a resource consumption pattern under the Leontief production function.

$$f(z) = RES\_CONS\_PAT = \begin{bmatrix} z_{1,1} & \cdots & z_{1,RCP} \\ \vdots & \ddots & \vdots \\ z_{PROD,1} & \cdots & z_{PROD,RCP} \end{bmatrix} = \begin{bmatrix} 1 & 5 & 0 \\ 2 & 1 & 3 \\ 0 & 5 & 3 \end{bmatrix} \qquad (2.8)$$

Fixed resource proportions (columns) are required to produce three products (rows). Product one is defined by one unit of input z_1 and five units of z_2. Since each input has certain unit costs (RCU), product costs are calculated as:

$$PC = \begin{bmatrix} 1 & 5 & 0 \\ 2 & 1 & 3 \\ 0 & 5 & 3 \end{bmatrix} * \begin{bmatrix} 10\$ \\ 12\$ \\ 7\$ \end{bmatrix} = \begin{bmatrix} 70\$ \\ 68\$ \\ 81\$ \end{bmatrix} \qquad (2.9)$$

[7] There are some studies which focus only on revenue maximization or cost minimization. Banker, Chang, and Cunningham (2003, p. 278), for example, argue that *"revenue maximization is a necessary condition for and consistent with profit maximization"*. However, section 2.6.2 shows that this assumption may breaks under certain conditions.

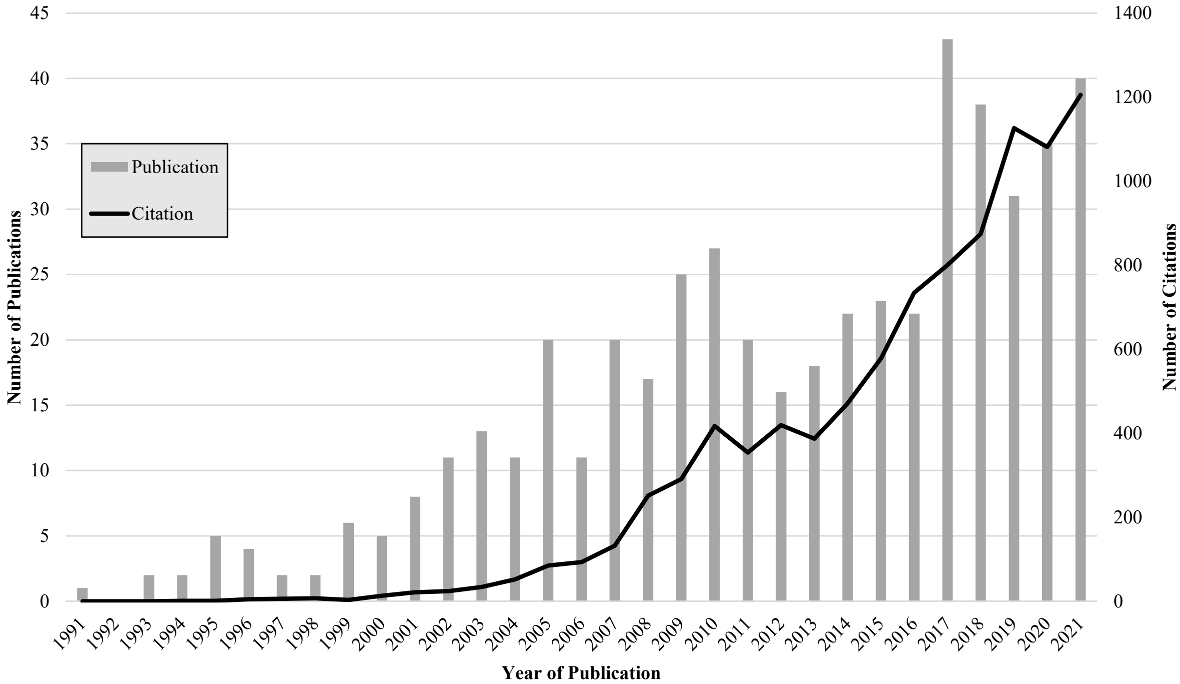
All assumptions and equations in this section are only valid under the strict conditions of perfect markets and the absence of cost function separability, which assumes that individual product costs are not affected by the overall output through effects like economies of scale and scope (Demski, 2008). However, this work will return to this aspect later in section 2.6.2 as an increased number of products affects resource consumption. For now, the summary of the idealized model is sufficient to understand the linkage of the Axiomatic Design Theory, presented in the following section.

2.4.3 An Engineering Perspective on Firms

Introduced by Suh (1990), the Axiomatic Design (AD) is a problem analysis tool supporting decision-making within firms (Gonçalves-Coelho & Mourão, 2007). As an abstraction from successful design processes in practice, it aims to support the *"analytical process of determining whether the proposed solution is correct or rational"* (Y.-S. Kim & Cochran, 2000, p. 80). Solutions developed by ADs' principles result from a systematic search in a solution space rather than random choices (Suh, 1995). It follows the belief that a good design is more than the result of individuals' experience, as AD's principles structure the design process which, allows to come up with more efficient and effective solutions in less time as the design space is systematically explored (Kulak, Cebi, & Kahraman, 2010). As a generalized form of the design process, ADs' principles can be applied to various fields where Kulak et al.'s (2010) review provides an overview of its applications, such as in product family and platform design (Jiao, Simpson, & Siddique, 2007), system design (Togay, Dogru, & Tanik, 2008), manufacturing design, software design, decision making (Gonçalves-Coelho & Mourão, 2007; Kahraman & Cebi, 2009) and other fields. Latest publications combine the classical AD with economic firm theory and, therefore, apply the AD in the field of cost estimation and cost system design (Mertens, 2020; Meßerschmidt, Gumpinger, Meyer, & Mertens, 2020; Matthias Meyer, Meßerschmidt, & Mertens, 2019). During the last decades, AD has received increasing attention. A search of publications containing the term 'axiomatic design' in the Web of Sciences core collection shows a substantial increase in publications and citations (Figure 2.4).

Figure 2.4:

The Number of publications and citations containing 'axiomatic design' in their title or abstract.

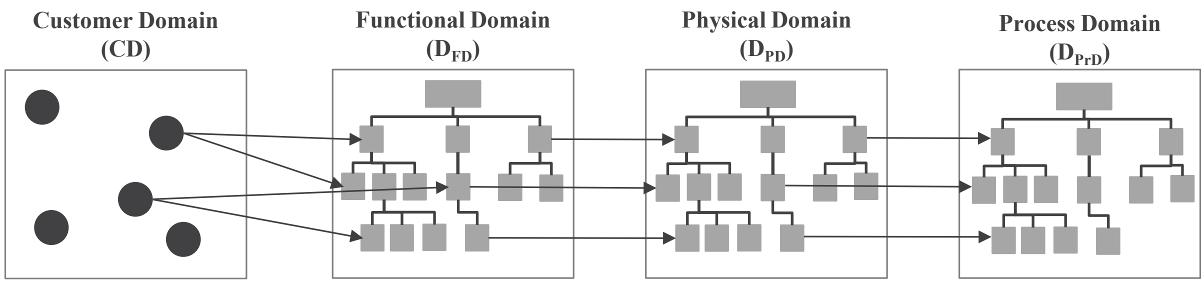


Note. Search for 'axiomatic design' in all fields, web of science core collection on February 10th, 2022

The Axiomatic Design is separated into four domains (Figure 2.5): the customer domain (CD), the functional domain (FD), the physical domain (PD), and the process domain (PrD). Each domain on the left represents the '*what to achieve*' and is answered by the '*how*' in the successor domain on the right. Therefore, each domain (except the process domain) contains requirements mapped to solutions in the following domain (E. M. Benavides, 2012; Suh, 1990, 1995). For example, customer needs (CN) in the CD are satisfied by functional requirements (FR), and functional requirements in the FD are realized by design parameters (DP) in the PD, which are realized by process variables (PV) in the PrD. Depending on the application of AD, the domains have different meanings (Suh, 1995).

Figure 2.5:

The four domains of the axiomatic design by Suh (1990). Each domain on the left is mapped to a domain on the right. The functional, physical, and process domains contain a hierarchy.



For this work, the following domain structure is assumed. The CD contains customer preferences, widely used in choice-based models such as conjoint analysis (V. R. Rao, 2014). The functional domain contains the functional requirements satisfied by components (PD).

Therefore, FD, PD, and the mapping between those two domains represent the product architecture as defined by (Ulrich, 1995). According to the product architecture, components are treated as functional carriers, which gives a proper theory for the domain definition in this work (Mertens, Rennpferdt, Greve, Krause, & Meyer, 2022). The process domain contains the individual processes required to manufacture the components in PD. Although domains may have different meanings depending on the specific application, the mapping is always the same. Elements from the source domain (E_{src}, left) are mapped to elements in the target domain (E_{tgt}, right) via design matrices (A), defined as:

$$E_{src} = A * E_{tgt} \tag{2.10}$$

The design matrices are the heart of each design as they are the results of decisions made during the design process (Suh, Cavique, & Foley, 2021). For example, let $E_{src} = \{E_{src,1}; E_{src,2}; E_{src,3}\}$ the source elements, $E_{tgt} = \{E_{tgt,1}; E_{tgt,2}; E_{tgt,3}; E_{tgt,4}\}$ the target elements and $A \in \mathbb{R}^{3x4}$ the design matrix defined as:

$$A = \begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 3 \end{pmatrix} \tag{2.11}$$

Then, the transition between source and target domain is than given as:

$$\begin{aligned} E_{src,1} &= 1E_{tgt,1} + 2E_{tgt,2} \\ E_{src,2} &= 1E_{tgt,2} + 1E_{tgt,3} \\ E_{src,3} &= 2E_{tgt,2} + 3E_{tgt,2} \end{aligned} \tag{2.12}$$

As the name suggests, the Axiomatic Design has two axioms to differentiate between good and bad designs. If both axioms are satisfied, a design is more favorable over another where only one or no axiom is satisfied. As a first axiom, the independence axiom states that FRs should be as independent as possible. Suh (1995) differentiates between three different types, as shown in Figure 2.6.

Figure 2.6:

Three different types of design. The main diagonal is highlighted.

uncoupled **decoupled** **coupled**

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$E_{tgt,3} = E_{tgt,4}$ $E_{tgt,3} = E_{tgt,4}$

Under an uncoupled design, source elements (rows) are linked independently to target elements (columns). A change in one target element is limited to a single source element.

Uncoupled design exists when the sum of each column in A is one, and there are no empty rows[8] (E. M. Benavides, 2012). A decoupled design exists if A is a triangular matrix. It does not matter whether A is a lower or upper triangular matrix, as the independence axiom is fulfilled if target elements are changed in a proper order (Suh, 1995). As permutation is allowed, a decoupled or uncoupled design exists if A has full rank. All other designs are coupled. Figure 2.6 (right) shows a coupled design as $E_{src,3} = E_{src,1} + E_{src,2}$. Uncoupled designs are more favorable than decoupled design, which are more favorable than coupled designs. Suh et al. (2021) note that coupled designs are equivalent to feedback loops within the system, another characteristic of complex systems, as argued in section 2.2. Therefore, uncoupled designs are less complex than decoupled ones, which are less complex than coupled designs.

The second axiom is called the information axiom, which states that designs with the lowest amount of information required should be preferred. Suh (1990) summarizes that a good design should satisfy both the independence and information axiom. The informational content for a design with N source elements is given by:

$$I = \sum_{i=1}^{N} \log\left(\frac{1}{p_j}\right) \tag{2.13}$$

where p_j is the probability of a target element ($E_{src,j}$) to fulfill the requirements of a source element $E_{src,i}$. Under diagonal design matrices, each source element has a one-to-one mapping into a target element. Thus, p_j, is defined as the proportion of common range (A_{cr}) to system range (A_{sr}) (Suh, 1995):

$$p_i = \frac{A_{cr}}{A_{sr}} \tag{2.14}$$

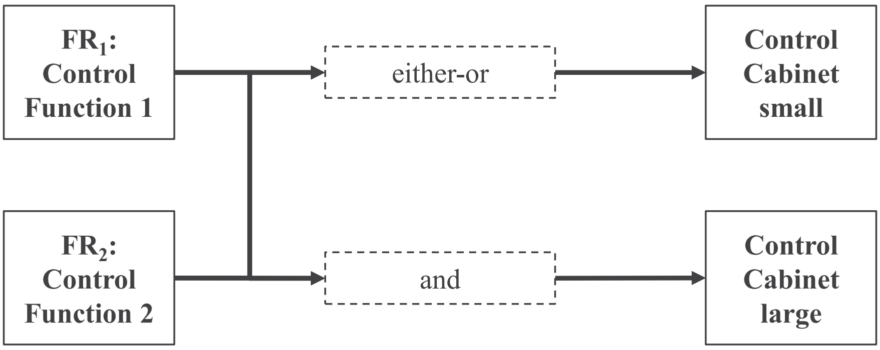
Suh (1995) takes the example of a rod that needs to be cut to one meter in length within a tolerance of plus-minus 1 µm. This defines the design range. The system range is defined as the probability of the system (e.g., a saw) meeting the tolerance. For many processes, a normal distribution can be assumed for these probabilities. The overlap of the design range with the system range A_{sr} is called the common range A_{cr}. Therefore, cutting a rod with a low-tolerance tool like a hacksaw is less preferable as its tolerance is larger and, therefore, A_{cr} is smaller than cutting the same rod with a high-precision tooling (Suh, 1995).

As stated in this form, the AD is only suitable for linear relationships across domains. AD principles, however, apply to non-linear designs, too. In such cases, the design matrix turns into a function that projects source elements to target elements (Suh, 2005). This more general form also covers cases where only the combination of source elements requires a specific target element. In such cases, checking whether the independence and information axioms are

[8] Suh (1995) states that A must be a diagonal matrix (permutation allowed) for an uncoupled design. However, this neglects cases where A is a non-square matrix. As shown in Figure 2.6, if the column sums of A are one, source elements are independent of each other and the changes of one target element influence only one source element. These cases also fulfill the requirements of an uncoupled design, as two target elements, depending on the same source element, can be merged, resulting in a diagonal square matrix again.

fulfilled becomes more difficult. Figure 2.7, for example, shows a non-linear design. Two control functions (FR_1 and FR_2) require the small control cabinet. However, if customers choose both functions simultaneously, the large cabinet is required to ensure that the components fit into the cabinet. This work, however, follows the assumption of linear relationships.

Figure 2.7:
Example for a non-linear design function. If either control system one or two are required, then the small cabinet is sufficient; otherwise the large cabinet is required.

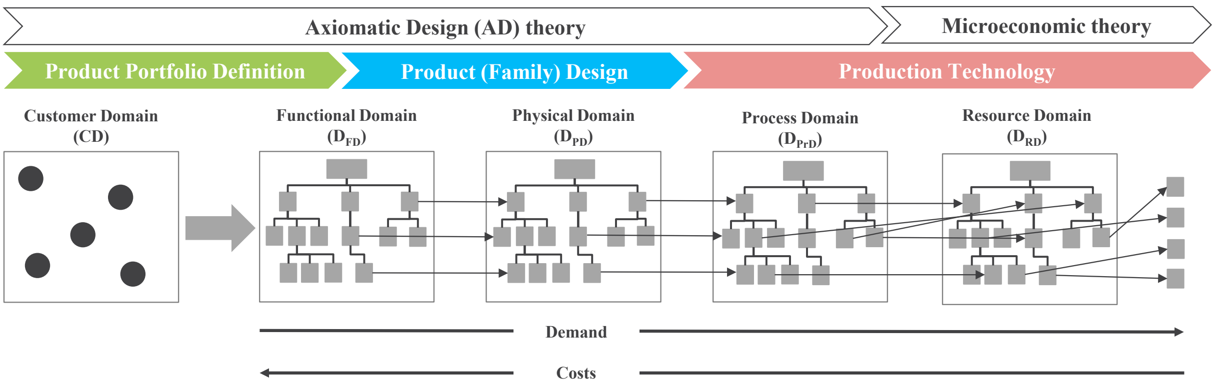


In contrast to other problem-solving tools such as TRIZ, FMEA, or TQM, the multi-domain view of the AD allows for an effective problem definition, analysis, and solving (Shirwaiker & Okudan, 2008). Lindemann, Maurer, and Braun (2009) argue that multi-domain approaches are suitable for complex system design. Since complex systems are characterized by emergence, complexity arises not solely from within systems' domains but also from the interaction between domains. Additionally, the domain view provides a pre-structuring of the system into sub-systems, allowing the investigation of individual aspects and effects within each sub-system (Lindemann et al., 2009). However, the AD is a tool for a systematic design process and its evaluation in terms of good or bad designs. Currently, it does not quantify solutions regarding cost, which is essential during product development, as design decisions affect costs (Fixson, 2006; Skirde, Kersten, & Schröder, 2016). In order to combine the widely accepted and frequently used principles of the AD with an economic perspective, Mertens (2020) introduces the Extended Axiomatic Design (EAD).

2.4.4 The Conceptual Extended Axiomatic Design

While microeconomics provides a general analytical foundation for maximizing firm profits, engineering aims to optimize products and processes in terms of costs and quality on a more detailed level, resulting in different views on the firm (Mertens, 2020). Although both views have slightly different objectives, they are somehow linked. The optimization of products and processes affects the profit as it reduces the total costs. Product and process modularity, for example, allows firms to increase flexibility (Campagnolo & Camuffo, 2009; Sawhney, 1998; Ulrich, 1994), positively affecting firms' dynamic capabilities. Therefore, several authors note the need for an interdisciplinary view combining the demand creation process, cost evaluation as well as product family design and planning (Jiao et al., 2007; Kumar, Chen, & Simpson, 2009; Mertens, 2020; Mertens, Schmidt, Yildiz, & Meyer, 2021; Meßerschmidt et al., 2020; Matthias Meyer et al., 2019; Simpson, 2004). Neither the AD nor the microeconomic theory alone is appropriate to address questions of product designs and their related costs. While the AD details product design, guides designers across different phases during the product development process, and allows them to differentiate good from bad designs, it lacks economic evaluation, a crucial aspect during product development (Fixson, 2005, 2006). Ulrich and Eppinger (2016) note that economic analyses are performed at the beginning of product development. Other authors highlight that engineering has a large lever in influencing later product costs in early-development stages (Fixson, 2006; Skirde et al., 2016). Conversely, microeconomic theory oversimplifies product design (Mertens, 2020). Due to the potential overlap, economics and engineering may profit from each other. Therefore, Mertens (2020) introduces the Extended Axiomatic Design (EAD), which connects engineering design theory, using the AD, with microeconomic firm theory (Figure 2.8). As a theory connecting framework, the EAD allows building more integrative models and offers the opportunity to standardize the variety of existing models (Mertens, 2020). Research on component commonality, for example, is characterized by a wide variety of different models in economics (Collier, 1982; Dogramaci, 1979; Eynan & Rosenblatt, 1996; Thonemann & Brandeau, 2000) and engineering (Kota, Sethuraman, & Miller, 2000; Simpson & D'Souza, 2004). By reflecting the need for engineering (granularity in product design) and economics (cost calculation), the EAD can potentially replace the individual model with one standardized one. A case study (section 3.2.5) demonstrates EAD's ability to model component commonality in sufficiently high granularity while relying on microeconomic principles.

Figure 2.8:
The conceptual Extended Axiomatic Design (EAD) as introduced by Mertens (2020)



The EAD is separated into three parts: the product portfolio definition, product family design, and the production technology. Product portfolio definition aims to collect customer needs, for example, via questionnaires, key-user interviews, or conjoint analysis, and transfer them into functional requirements (product specifications). Often, customer needs are qualitative and abstract. Product management's task is interpreting and transferring these needs into functional requirements. Therefore, based on the customer needs, a set of functional requirements is derived (Ulrich & Eppinger, 2016). These functional requirements are mapped to physical elements such as hardware, software, or service components. Processes further realize the physical elements. Originally designed as a framework to formalize product design, the AD does not include cost information. Mertens (2020) closes this gap by adding the microeconomic view after the process domain. As introduced in section 2.4.2, the microeconomic view connects processes (activities) with resources and adds the resource domain (RD).

Products consume these resources, resulting in costs[9]. Since the model holds information on element's connection across domains, these costs can be traced back to processes, physical elements, or functional requirements. At this point, it is essential to note that the microeconomic term 'resources' differs from resources as understood by the economic firm theory. While economic firm theory has a broader understanding of resources, such as capabilities or routines, microeconomics defines resources as all elements necessary to create a product. The EAD refers to the latter, and resources are, for example, raw materials, workforce, or manufacturing machines. Another difference to the AD is its product and demand view. While the AD solely focuses on the design, Mertens (2020) suggests taking a more global perspective on product design. Instead of optimizing a single product, firms take advantage if they optimize the design for the entire product family (Jiao et al., 2007; D. Krause & Gebhardt, 2018). For example, a sub-optimal design of low-running products might be acceptable when the design for high-volume products is highly optimized. The product view is added by defining products as a set of domain elements. Product management, for example, describes products via functional requirements in specification sheets (D. Krause & Gebhardt, 2018; C. Weber, 2007), product engineering via components in a bill-of-material (Hegge & Wortmann, 1991), production engineering as a set of processes such as in routing sheets (Swamidass, 2000) or controlling via their resource consumption (V. Anand et al., 2019).

Compared to the classical AD, the EAD is an integrative conceptual model, allowing for a common understanding across disciplines such as product management, engineering, and cost accounting. Since the EAD holds information on the product design and its costs, it supports multi-attribute decision-making during product development. Multi-attribute decision-making (MADM) problems describe the aim to maximize or minimize confronting objectives (Gonçalves-Coelho & Mourão, 2007; Kahraman & Cebi, 2009; Kulak et al., 2010) and occur all along the development process (Chen, Pan, Ma, Hou, & Zhao, 2022). Extending AD's pure design view to a product family design view, the EAD defines products via a combination of domain elements. Elements combination, however, is defined by the underlying design. It thus further supports the product planning process, which belongs to one of the information-intensive decisions in firms (Balakrishnan et al., 2012). Besides these advantages, the EAD is a relatively new model and, therefore, has unsolved issues.

Even though the EAD combines two widely accepted and empirically validated models, it requires empirical validation. The model, for example, assumes that a design matrix directly links source and target elements. Empirical literature and theory, however, note the presence of inter- and intra-domain dependencies (Bonjour & Micaëlli, 2010; Lindemann et al., 2009; Sawai et al., 2017), raising the need for an extension of the one-stage domain mapping. Another caveat is the incomplete inclusion of the product view. Although products are sub-sets consisting of domain elements, the model does not sufficiently explain how the design and the product view (product demand and configuration) are linked. For example, product demand and its distribution across products have several implications. While the first directly impacts the firm's microeconomic objective, the latter is essential for questions of design

[9] For a more detailed explanation of how resource consumption leads to costs see section 2.6.1.

optimization. Increasing the degree of component commonality through an overdesign of components is such a technique. However, literature (see Labro, 2004 for an introduction) notes that overdesign is only beneficial if increasing economies of scale outweigh higher material costs. Thus, information on product demand is necessary to decide whether an overdesign gains advantages. Finally, the mapping between customer needs and functional requirements is not fully included yet. Although this mapping is complex, using the discrete choice theory would offer an opportunity to formalize this step. Other caveats are EAD's limitation to linear dependencies (e.g., see Figure 2.7) and the time invariance.

Besides the open issues, the EAD is a first step towards standardizing product family design models and, thus, following the call of Jung et al. (2022). Hennig et al. (2022), for example, note the variety of such models and measures where a formalized model allows for joint and faster theory development (Mertens, 2020) as well as easier replication in future research.

2.5 Complexity within the Firm System

The previous sections introduced the concept of complexity (2.2) and proposed an interdisciplinary product family model (2.4). This section aims to connect these two parts. In doing so, the first section (2.5.1) defines the firm as a system and argues that firms represent complex systems as they show all major characteristics (2.5.2). Section 2.5.3 answers why firms face a certain level of complexity and how it manifests within firms (2.5.4).

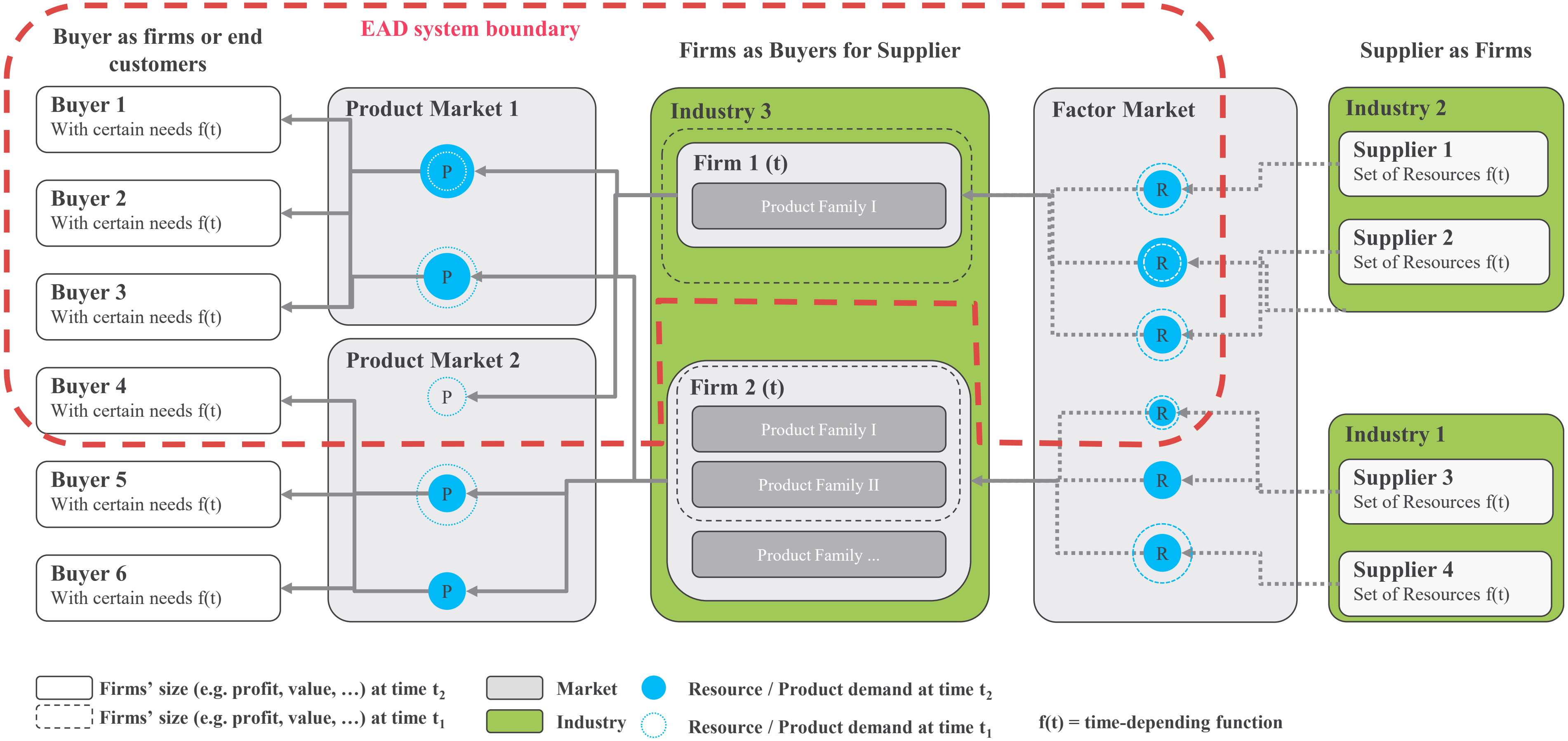
2.5.1 Boundaries of the Firm System

The conceptual EAD defines a multiproduct firm from a joined engineering and production economics perspective. This chapter puts the EAD into a broader perspective by discussing its interfaces and hierarchical levels. Understanding EAD's boundaries and environment is necessary since later sections show that a large proportion of a firm's (internal) complexity is induced by external factors[10]. Figure 2.9 provides an overview of EADs' boundaries (red dotted line).

[10] Section 2.5.3 takes a deeper look at the different types of complexity.

Figure 2.9

Overview of the firm environment, consisting of factor markets, suppliers, product markets, industries and buyers. The red line defines the boundaries of the firm system used in this work.



On the most granular level, the EAD represents the design of one product family. Firms, however, can offer more than one product family at once (e.g., firm 2). Product families are part of a product line, and product lines are part of a production program (D. Krause & Gebhardt, 2018)[11]. The EAD, therefore, represents not an entire firm but only a single product family of the firm's production program. For simplicity, the following discussion focuses on a single-product family but a multi-product firm such as firm. According to the RBV, firms hold certain resources at a specific time (Caves, 1980; Wernerfelt, 1984). To be clear, resources, as mentioned by the firm theory, contain more than resources defined by the microeconomic production theory. While the EAD defines resources as inputs directly required for the output, economic firm theory has a broader understanding. Wernerfelt (1984), for example, notes brand names, in-house knowledge of technology, employment of skilled personnel, trade contacts, machinery, efficient procedures, and capital as examples of resources. Fixed in the short run, firms must deal with existing resources. However, they can decide which resources to hold at which quantity in the long run (Balakrishnan & Sivaramakrishnan, 2002). Differences across firms occur as they have different sets of resources and different internal resource accumulation (Kor & Mahoney, 2004; Nelson, 1991). These differences explain why some firms gain a competitive advantage over others. For example, if firms' capabilities allow for more efficient production, the total resource consumption decreases and, therefore, costs as well, leading to increased revenues. Technological achievements because of internal R&D allow firms to offer new products, and better market insights enable firms to offer a market-fitted product mix. These advantages allow firms to grow while others shrink (Hounshell, 1989; Nelson & Winter, 1982; Reich, 1985). As noted, the EAD assumes a multi-product firm with one product family. Effects between different product families, such as shared resources or components, and interactions between firms, such as the adaption of competitors' strategy (Barney, 1991), are neglected in this work and may indicate a current limitation of the EAD.

Production theory links product demand and resource consumption, where increased product quantities lead to higher resource consumption. Under the assumption of the Leontief production function, consumption increases linearly with the quantity. For example, if firms double the output, the total resource consumption also doubles. This may indicate another limitation of the EAD since, in practice (dis-), economies of scale and scope exist, such as assumed by the Cobb-Douglas production function. Resources are sourced from factor markets where firms search for vendors offering the lowest price (Labro, 2004). Under perfect markets, input prices are the only selection criteria since quality and availability are neglected. Nevertheless, supplier selection as part of the resource allocation process is a complex decision process in practice (Boer, Labro, & Morlacchi, 2001). Some suppliers, for example, are more flexible in reacting to demand shocks as they guarantee delivery reliability over others. On the other hand, it can be beneficial to source different resources from the same supplier to gain economies of scale. The supplier domain is not part of the EAD as it contains a strategic component. However, costs caused by supplier acquisition and management (e.g., S. W. Anderson &

[11] There are different hierarchical levels across different authors. For example, McKay, Erens, and Bloor (1996) define another layer between products and product families called product range. Meffert, Burmann, and Kirchgeorg (2015) arrange products directly into product lines as part of the firm's product program.

Dekker, 2009) are considered in section 4. For reasons of simplicity, this work assumes that each resource is sourced from one and only one (no double sourcing) supplier.

A firm offers products in different markets (Figure 2.9, left). Individual markets are characterized by different customer preferences, willingness-to-pay, and, therefore, target prices, as well as different governmental restrictions and more. The EAD contains the customer needs, and via a customer selection process, these needs are transferred into a requested product demand $\hat{q}$, fulfilled by a produced demand q. The EAD assumes that $\hat{q} = q$, which implies that no stock is required, following a common simplification (V. Anand et al., 2019)[12]. Interactions with other products from the same firm (cannibalization) and interactions with products from competitors still need to be defined and indicate additional unsolved issues. Product demand varies over time due to endogenous and exogenous influences. Based on internal decision-making, firms drop or introduce certain products from their product mix. Exogenous effects, for example, are changing prices on factor markets due to market shocks or changing customer needs. Since the EAD is a single-period model in its current formulation, all dynamic aspects are neglected. Those aspects, however, become important if the EAD is extended beyond the steady state in the future.

Although the EAD is a single-period model, problems that arise during a multi-period extension are briefly discussed. In real (imperfect) markets, demand varies over time due to changing customer preferences. Firms' reaction is not instantaneous since the internal resource accumulation and resources must be adapted based on the new demand. Therefore, they face the challenge of solving the capacity planning problem under uncertainty (Balakrishnan, Pugely, & Shah, 2017; Balakrishnan & Sivaramakrishnan, 2002; van Mieghem, 2003). Too optimistic capacity planning ($\hat{q} \ll q$), for example, results in more resources being allocated as needed. The result is an overcapacity where the currently existing capacity increases the firm's total costs, impacting its microeconomic objective. Too pessimistic planning ($\hat{q} \gg q$), on the other side, results in unused capacity, again affecting the microeconomic objective. Therefore, firms must decide which resources to allocate for a certain period. Balakrishnan et al. (2012) note that these decisions are the most information-extensive and complex decisions firms make. Several studies investigate the effect of uncertainty on capacity planning. For example, S. Hansen and Magee (1993, p. 636) discuss the *„relationship between the optimal opportunity cost of capacity and the sunk costs of existing capacity“.* In the context of cost-based decision-making, V. Anand et al. (2017) show the impact of uncertainty on product costs. From a multi-period perspective, these effects would add feedback loops to the firm system, resulting in a discrete dynamic system. Balakrishnan and Sivaramakrishnan (2002) summarize these feedback loops as the 'grand program'. According to control system theory, such systems can oscillate under certain conditions (Rabbath & Léchevin, 2014). Improved capacity planning (e.g., through more available information) and established cost management systems work as damping elements. However, V. Anand et al. (2019) note that including the grand program dramatically increases the model complexity and assumes that resources are acquired

[12] This assumption is relaxed in later sections, where a simple inventory model is assumed (section 4.3.3.3).

instantaneously. Following this simplification for the EAD, the exclusion indicates another limitation that needs to be addressed in future research.

Beyond the grand program, firms have more feedback loops. According to Nelson and Winter's (1982) evolutionary theory, improving routines (e.g., through better resource recombination) is another example of a feedback loop as it influences decision-making in the next period. However, such feedback loops are limited to the multi-period view. Since the EAD is a single-period model, these feedback loops are neglected. Feedback loops, however, represent the first property of complex systems, where the following section takes a deeper look.

2.5.2 The Firm as a Complex System

Based on the defined firm system, this section argues that the firm is a complex system as it shows all characteristics introduced in section 2.2, where Table 2.1 provides a summary. Feedback loops such as the grand program or learning to improve routines are examples of a time dependency. Thus, firm’s current state is the result of its history since decisions made in previous periods may still oscillate within several feedback loops and influence today's state. Therefore, the state of a firm may change, even if external and internal conditions remain constant since previous effects have not fully decayed yet.

Table 2.1:
Characteristics of complex systems manifested in the firm.

Characteristics	Example
Hierarchical order	A product as a set of features, components, processes or resources and a firm as a set of different product families or elements as entities of domains.
Open character	Many interfaces such as customer preferences, resource prices, competitor interactions, or product cannibalization.
Fuzzy boundaries	Interaction effects of product mix and customer preferences on the demand vector because of the customer choice process.
Non-trivial connections	Mapping between customer needs and functional requirements (customer choice process) affecting the product mix.
Time dependency	Feedback loops within the firm system (e.g., grand program, learning)

The remaining characteristics refer to the static view of complexity. The customer choice process is an example of a non-trivial connection and fuzzy boundaries. According to classical discrete choice theory (e.g., S. P. Anderson, Palma, & Thisse, 1992; Ben-Akiva & Lerman, 1985), customers aim to maximize their utility value. However, there is also the possibility that they choose no product at all. In imperfect markets, customers do not have full transparency on product alternatives, resulting in search costs to find the best product alternative (Dellaert & Häubl, 2012; Tsafarakis, Grigoroudis, & Matsatsinis, 2011). If these costs become too high, they abort the purchasing process or select a product that does not represent the maximum utility. Due to the complexity of the customer choice process and detailed discussion in marketing literature, it is excluded by cost accounting models, indicating a fuzzy boundary of the firm system. Cost accounting studies (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011) assume an exogenously given demand instead of demand being an endogenous outcome of the customer choice process[13]. The open character is another property present in firms. Many interfaces exist at different ends, such as the input of resources or market information (e.g., customer requirements, competitor behavior) and the output of products. Regarding the hierarchical order, products are defined as a set of domain elements, and a product family is defined as the set of all products with certain similarities. These similarities exist since products are derived from a set of functional requirements, components, processes, or resources (D. Krause & Gebhardt, 2018; Marc Meyer, 1997). While domain elements represent the model's most granular (atomic) layer, the product family is the most aggregated. Both ends are not immovable, as there is always a broader or more detailed context.

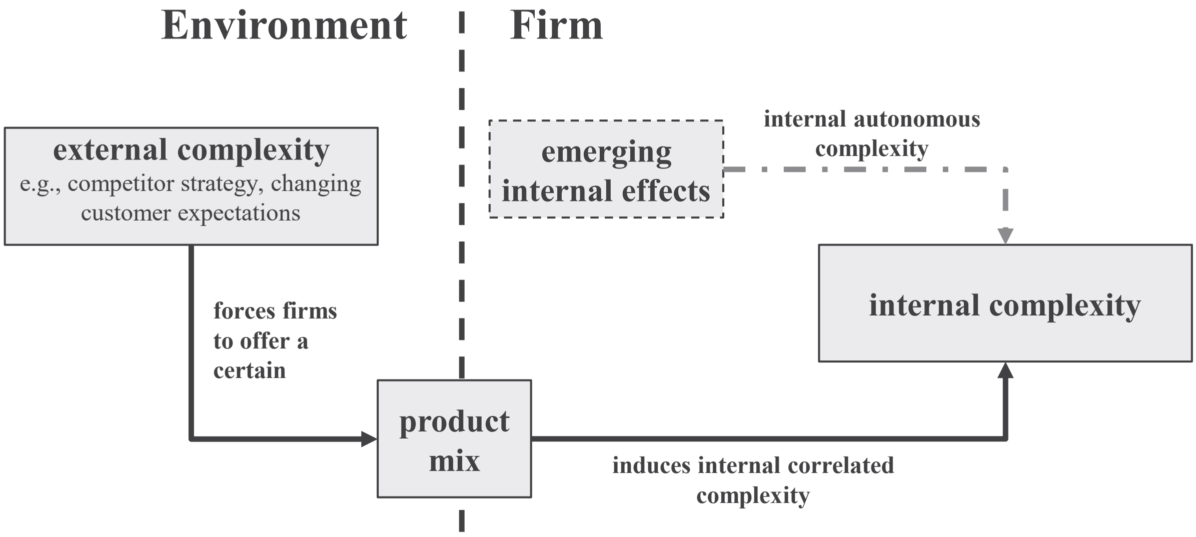
2.5.3 The Causes of Complexity

Not all firms are equally complex. While some firms have a moderate number of products in their product mix, like Pepsi with around 3500 individual products (Meinrenken et al., 2014), others have nearly no restriction in the combination of functional requirements and, therefore, many distinct product variants such as car manufacturer with 10^{17} up to 10^{38} individual product variants for their cars (S. J. Hu et al., 2008; Neff, 2016). The economic firm theory explains differences across firms via external effects such as different market environments (Nelson, 1991) or competitors' strategies (Barney, 1991), stimulating the firm system from the outside and triggering internal effects. These effects are also noted as internally correlated. The adaption of competitive strategy is an example of such an internal effect induced by external causes. To compete with other firms, they can copy competitor's strategies (external), inducing internal effects such as routine changes and new capabilities (internally correlated). Additionally, firm theory notes internal effects that emerge within firms and are autonomous from external sources. Lippman and Rumelt (2003) note that firms' home-grown resources are an example for such autonomous effects. As part of firms' internal resource accumulation, they result from firms' technological evolution in capabilities, such as through technological achievements or learning. Therefore, differences across firms are caused by external effects that trigger internal (correlated) effects and internal autonomous effects without external causes.

Hence, it is unsurprising that a structured literature review by Vogel and Lasch (2016) uses the same classification for the complexity firms face. Based on 212 publications across different fields, they define complexity as a result of internally correlated complexity induced by external complexity and internal autonomous complexity that emerges within firms. Figure 2.10 shows the dependencies between external and internal complexity, where the product mix is the connecting element as it defines the extent to which a firm responds to external complexity (Marti, 2007; Maurer, 2017). While firms are limited in influencing external complexity, as it lies outside their control (Sinha & Suh, 2018), they can decide how to respond and, therefore, have control over the level of internal complexity (Child et al., 1992; Collinson & Jay, 2012; Götzfried, 2013; Grimm et al., 2014; Maurer, 2017; Piller, 2003).

[13] This work is no exception. However, it provides a blueprint for the integration of the customer choice process in future research (section 3.2.1)

Figure 2.10:
Sources of internal complexity



Following the review by Vogel and Lasch (2016), external complexity consists of society and market complexity. Society complexity summarizes cultural, ecological, legal, and political factors and standards, while market complexity contains demand, competitive, supply, and technological complexity. Table 2.2 details these drivers of external complexity and provides empirical examples. Responding to these external drivers, firms offer a product mix instead of a single standardized product. The extent of product variety ranges from some product variants, such as the Apple iPhone, to 10^{38} variants in the car industry.

Table 2.2:
External complexity consists of society and market complexity, according to Vogel and Lasch (2016).

Type	Driver	Description	Example
Society Complexity	Cultural factors	Different languages across the market, education level, and willingness to pay	In 2004, Windows offered "Windows XP Starter Edition" to address the markets in Thailand, Malaysia, and Indonesia. This version had fewer features than the standard XP as it was designed to get in touch with personal computers (Microsoft, 2004).
	Legal factors	Governmental restrictions manifested in laws or technical norms must be fulfilled to sell a product on that specific market (must-have factors)	The US and EU have different legal restrictions for car headlights. For European cars, an 'ECE approval' is required. The US requires 'SCA' or 'DOT' lights. This results in different versions of car headlights where the turn signal is orange in the US and transparent in Europe (Toma, 2016).
	Standards	Guidelines, but without legal restrictions, should - have factors to sell a product on the market	Until 2021, the European Commission voluntarily tried standardizing charging plugs for cell phones across manufacturers (European Commission, 2021).
Market Complexity	Supply	Number of suppliers, supply reliability, heterogeneity of supplier objects, …	While a high number of available suppliers increases the decision complexity, a low number of supplier increase the complexity in supply reliability, such as in the electrical power market, where five companies produce 65% of German electricity (Bundesnetzagentur, 2022).
	Technological (external)	Technological progress, innovation, technology, and materials	The development of voice-over-IP (VOIP) leads to new phone types based on different underlying technology. While traditional telephones convert a continuous electrical signal into voice and vice versa, VOIP phones reassemble data packages (Fuchs & Golenhofen, 2019).
	Competitive	Number of competitors, competitive demand, and activities	A case study by Salvador et al. (2002) found that the number of competitors for a major European moped manufacturer increased from three in 1996 to 20 by 2000.
	Demand	Globalization of demand, the individuality of customer demand and demand fluctuation	The semiconductor industry is characterized by *"high demand volatility, rapidly changing environments [and] manufacturing lead-times (months) are usually longer than customer order lead-times"* (Govindaraju, Achter, Ponsignon, Ehm, & Meyer, 2018, p. 148). Lead times of 100 days or more are typical (Lapedus, 2017).
	General market related	Market requirements, size, uncertainty, saturation of the market	Dräger Medical offers workplaces for intensive care and surgical personnel. Due to the high complexity of such a workplace and the existing infrastructure on the customer side (location of gas outlets, ceiling bearings, available space) nearly every workplace is unique (Blecker & Kersten, 2006).

A second source of complexity, autonomous internal complexity, emerges from within the firm. Vogel and Lasch (2016) identify organizational, process, production, planning, control, and information complexity, as well as resource, logistics, sales, and distribution complexity, as drivers of internal autonomous complexity. Organizational complexity, for example, increases with the number of employees, their nationalities and distribution worldwide, or firms' hierarchical structure (Vogel & Lasch, 2016). Logistical complexity arises when companies decide on a multi-sourcing strategy, where the same resource is sourced from different suppliers to mitigate delivery delays (Calvo & Martínez-de-Albéniz, 2016). Resource complexity describes the availability of internal resources or the sequence of resource's application. The size and structure of a firm's internal distributor network is an example of sales and distribution complexity. All these drivers have in common that they do not depend on external factors or, at least, are only weakly influenced by those. Internal decision-making or historical reasons are causes of autonomous complexity. Multi-sourcing of components, for example, can result from a strategic decision due to external causes such as delay in supply or bad decision-making, as Abdelkafi (2008) notes. For example, Airbus, founded as a European joint venture, transferred three of its eight final assembly lines for the A320 product family to Hamburg, Germany (Muller, 1990). This increases supplier complexity since parts are shipped within Europe, while a centralized assembly of all Airbus products would be more efficient [14].

While literature describes internal complexity as a result of external or internal autonomous complexity, in practice, causes are fuzzy. For example, multi-sourcing might result from external (product mix) and internal autonomous causes (bad decision-making) simultaneously. Against this background, treating internal complexity as a construct of induced and autonomous drivers might be impractical. Instead, another stream of literature describes internal complexity by its manifestation (effects) within the firm instead of its causes. This allows the classification of complexity drivers based on an observation (e.g., increased development efforts) instead of searching for potential causes. The following section takes this perspective as it discusses the manifestation of complexity within firms.

2.5.4 Manifestation of Complexity

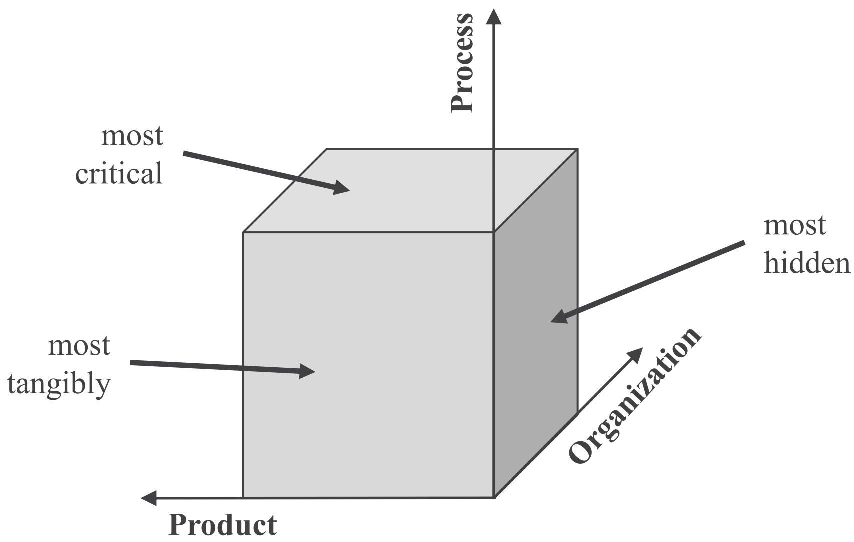
Even if two firms face the same external complexity, offer the same product mix, and have an absence of internal autonomous complexity, their level of internal complexity can vary. Firms' capabilities decide which degree of internal complexity they face (Child, Diederichs, Sanders, Wisniowski, & Cummings, 1991; Collinson & Jay, 2012). For example, firm (A) s a specialist in producing standardized products using highly automated assembly lines, allowing the production of products in a high quantity but with none or only minor variations. A second firm (B) focuses on fulfilling customer needs by offering customized solutions. Due to additional customizing steps and more job-shop-oriented production, production quantities are lower compared to the first firm. If firm A now decides to compete with firm B by offering the same product variants, it will be confronted with a much higher degree of internal complexity than firm B. The reason is that additional and manual workarounds might be necessary since the automated assembly line cannot produce products with such large variations among each other. On the other hand, if firm B produced highly standardized products, it would face a much higher degree of internal complexity than firm A. While a flexible (e.g., modular) product design and job shop-oriented production allows the creation of products with minor differences more efficiently, it is characterized by a small degree of specialization, resulting in performance trade-offs (Fuchs & Golenhofen, 2019). The example demonstrates that firms' degree of complexity depends on several factors, such as the mapping between products, processes, and organization (Bonjour & Micaëlli, 2010; Wilson & Perumal, 2010). While the

[14] To be precise, this applies only for the aspect of supply complexity. There are many other reasons to separate production lines.

economic consequences of complexity are discussed later in this work (section 2.6), this chapter explores the relationship between the different levels within the company in more detail.

In doing so, Wilson and Perumal's (2010) conceptual complexity cube model is used (Figure 2.11), which is characterized by the dimensions of product, process, and organizational complexity. Product complexity is described by the number and heterogeneity of products or services a firm offers and refers to what other authors note as product and product portfolio complexity (Marti, 2007; Vogel & Lasch, 2016). It further contains the structural complexity in terms of component variety, interfaces, and their dependency on function requirements (Jung et al., 2022; Sinha & Suh, 2018). El Maraghy and Urbanic (2003) define product complexity as the variations in materials, design, or specifications for product families' components. Process complexity is characterized by all activities caused by product variety, such as rework, additional set-ups, or coordination efforts (Wilson & Perumal, 2010). Vogel and Lasch (2016) further add that process complexity is defined by the number of processes, the degree of process optimization, stability, standardization or interfaces. Organizational complexity is defined as organizational structures (Fixson, 2007) as well as *"the demands placed on (…) [them such as] staffing, assets, policies, metrics"* (Wilson & Perumal, 2010, p. 13) to offer a certain product mix. Marti (2007) notes the interfaces between different resources or the degree of fragmentation within the organizations as another driver of organizational complexity.

Figure 2.11
Complexity cube as introduced by Wilson and Perumal (2010)



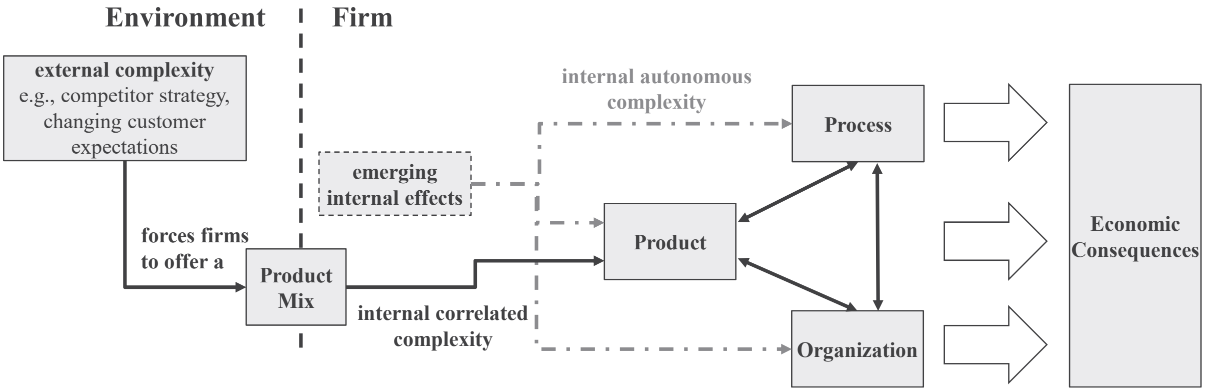
Cubes' faces represent the interactions between the three dimensions. First and most prominent is the product-process face, which describes interdependencies between product and process complexity and is noted by several authors (e.g., Chaudhuri & Boer, 2016; Danese & Romano, 2004; El Maraghy & Urbanic, 2003; Hackl et al., 2020; Hvam, Hansen, Forza, Mortensen, & Haug, 2020; Trattner et al., 2019). These studies note that more distinct components lead to more setup changes and require new tools or manufacturing machines to handle the variety. Other authors note the impact of product variety on processes along the supply chain, such as engineering, manufacturing, procurement, logistics, and sales processes (Lyons et al., 2020; L. L. Zhang, Lee, & Akhtar, 2020). This layer is seen as the most tangibly as their effects are directly related to the increasing number and heterogeneity of products, components, and dependencies.

The product-organization face is discussed in the context of the mirroring hypothesis in the literature (Henderson & Clark, 1990; MacCormack, Baldwin, & Rusnak, 2012; Sinha & Suh, 2018; Sosa, Eppinger, & Rowles, 2004). It states that the resulting product structure reflects the organizational structure and vice-versa (MacCormack et al., 2012; Sosa et al., 2004). While some authors note the importance of aligning organizational structure to product structure, such as team composition and communication, others find that perfect mirroring is only beneficial under certain conditions (Querbes & Frenken, 2018). A literature review by Sorkun and Furlan (2017) shows that most studies criticize the mirroring hypotheses as additional factors such as fast technological change, complex product architecture, or cost decisions further influence the alignment between product and organizational structure. Additionally, Fixson (2007) notes that an organization's capabilities are another driver of the structure and, therefore, the alignment between product and creating organization. For example, a firm with competencies scattered across employees may require a different structure than a firm with dedicated experts, although both create the same products. Besides the criticism, Wilson and Perumal (2010) argue that the product-organization layer is the most critical in terms of cost and capital, deeply rooted and most difficult to change since resources are tied to firms in the short run. Decisions on making or buying components are an example of challenges on this layer. These decisions define whether a component is produced in-house with potential investments in new resources such as machines and knowledge or by external suppliers. This results in higher component costs since complexity is shifted to the supplier side. As another example in product development, Hackl et al. (2020) note that the product-organizational complexity decreases when a modular organization develops a modular product since it allows a better match between organization and tasks, which is an enabler for parallel development of modules as they are more decoupled than in integral product design.

While the previous faces are often visible, the process-organization layer is the most hidden due to conflicting objectives (Wilson & Perumal, 2010). Two examples demonstrate such conflicts. A strong alignment between firms' structure and processes allows simplified administration. However, it hides the holistic picture necessary for efficient resource allocation (Wilson & Perumal, 2010). Conducting a survey among 155 U.S. firms, Visser et al. (2010) investigate how organizational structure affects the product development process. They find that firms with *"cross-functional structure […] perform better in terms of breakthrough innovation performance"* (Visser et al., 2010, p. 291) while firms with functional structure have a better performance in terms of derivative innovations. This case study demonstrates that there is no good or bad structure for firms per se. Instead, the mapping between processes and organization must fit the firm's strategy. Symptomatic for a bad structure is when *"exceptions become the rule"* (Wilson & Perumal, 2010, p. 135) or when departments *"work against each other"* (Wilson & Perumal, 2010, p. 132), although they follow the same goals.

This section highlights that complexity manifests at different points within firms. External complexity induces internal complexity via the product mix (see Figure 2.12). Induced complexity manifests in the product complexity as a direct entry point and then spreads to processes and the organization. A new product variant, for example, requires a new component, which requires new processes and suppliers. Internal autonomous complexity can affect each axis of the complexity cube directly. Bad decision-making can induce complexity as the processes do not fit well with the organization or the products. Unnecessary interfaces between two components are an example of bad decision-making, leading to additional assembly steps and higher costs. A better solution would be to unify both components into one single component. Based on the examples discussed in this section, it becomes clear that internal complexity is associated with economic consequences. Therefore, the following section takes a closer look at these consequences.

Figure 2.12
Manifestation of internal complexity on product, process, and organizational level having certain consequences on costs, quality, or time efforts.



2.6 Economic Consequences of Internal Complexity

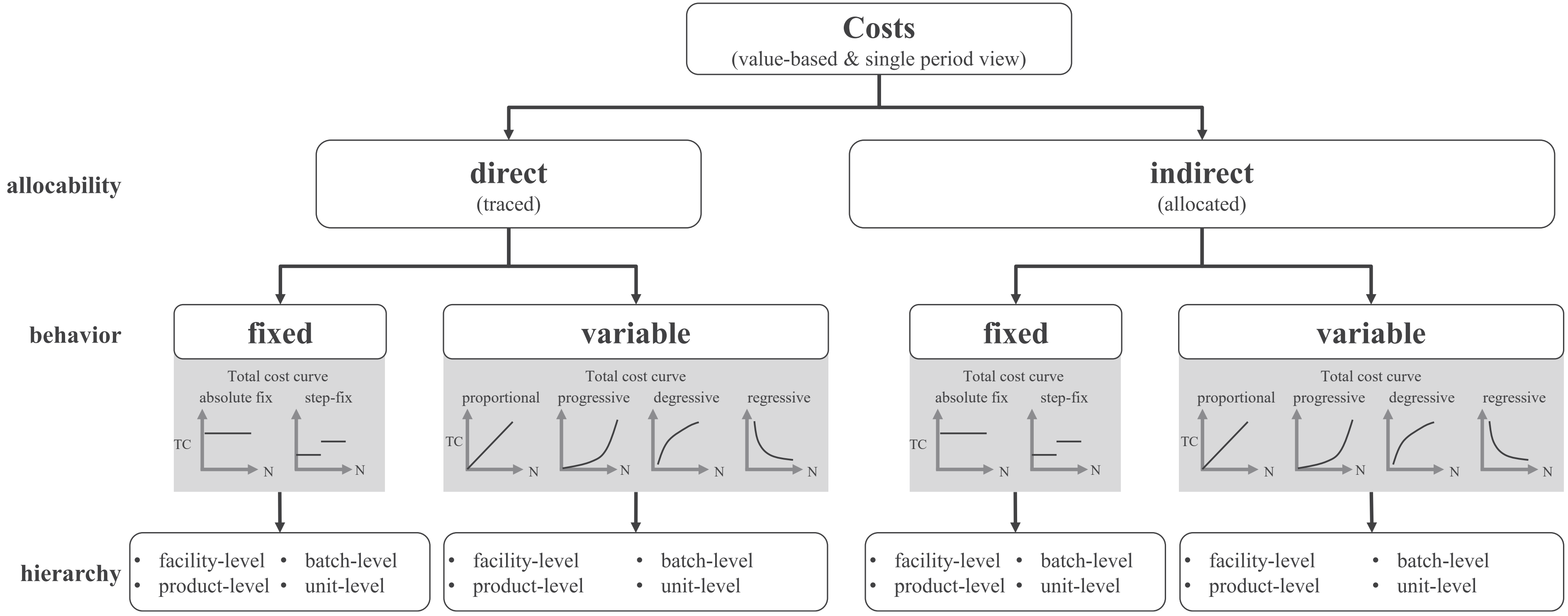
Internal complexity has several economic consequences, such as costs, quality, and lead times (Hackl, 2022; Trattner et al., 2019). This work focuses solely on the cost aspect and is separated into the total cost associated with internal complexity and the effects on product costing system accuracy. Focusing on the cost aspect is vital as it directly influences a firm's microeconomic objective of profit maximization. Based on the microeconomic view (section 2.4.2), profit is maximized by reducing costs or increasing revenue. However, both levers are not independent of each other. An increase in revenue by increasing the number of products (selling opportunities) to meet customer requirements more precisely is associated with increased costs (Hackl et al., 2020; Lancaster, 1990; Lyons et al., 2020; Trattner et al., 2019). Thus, firms face a tradeoff between the optimal level of product variety and how to deliver it at minimal costs (Ripperda & Krause, 2017). The first part of this section (2.6.1-2.6.2) focuses on this aspect. In doing so, the cost effects of internal complexities are discussed.

Speaking of costs, but rarely discussed, is the impact of internal complexity on the accuracy of product costing systems (PCS). PCSs are a set of rules on measuring and allocating costs on cost objects such as products (Friedl, 2010). While some costs are traced directly onto products and are error-free, others, so-called indirect costs, are allocated using specific heuristics, leading to errors in product costs. Indirect costs exist since firms face the trade-off between the costs of gathering cost information and the expected disadvantages of errors in product costing (Labro & Vanhoucke, 2007). Errors in product costs are critical as they also affect a firm's microeconomic objective. Products for which the PCSs report more costs than the true but hidden costs seem unprofitable (M. Gupta & Galloway, 2003; D. R. Hansen & Mowen, 2006). Aiming to optimize profits, managers may drop such products, although their contribution margins are sufficiently high. Products reported with lower costs are even more dangerous for firms. Such products seem profitable, but their true contribution margins are much lower or negative (Horngren et al., 2012). Since firms believe that such products are still profitable as the product costing system gives the illusion of positive contribution margins (Cooper & Kaplan, 1988a), they are not dropped. The second part of this section takes a closer look at this topic as it provides the theoretical foundations of product costing systems and their role in firms' decision-making (section 2.6.3). Errors in product costing systems caused by the presence of indirect costs are discussed in Section 2.6.4.

2.6.1 Costs - A Brief Introduction

The value-based perspective defines 'costs' as a measure of resource consumption by a specific cost object (Friedl, 2010). Therefore, they are seen as effect indicators (Cokins, 2001). Generally, a cost object is every entity for which costs should be reported. Examples of cost objects are cost centers, products, or periods (Friedl, 2010). Costs are classified into different categories, as Figure 2.13 summarizes. A single-period view differs between allocability, behavior, and hierarchy. While the first two dimensions refer to a traditional cost accounting view, the hierarchy is introduced by the activity-based view of costs (Cooper & Kaplan, 1991). Allocability differs between direct and indirect costs. Direct costs are assigned directly to a cost object by tracing such costs via PCSs (Horngren et al., 2012). Material costs are an example of such costs since firms can trace the assembly of components into final products. Indirect costs cannot be assigned directly to a cost object (Friedl, 2010). They result from consumed resources shared by at least two products (Balakrishnan et al., 2012). Such costs, therefore, need to be allocated by PCSs. Horngren et al. (2012) note that tracing such costs is either impossible or not economically feasible. Firms' R&D efforts are an example of indirect costs which are difficult to trace. When designing an interface between two components, the exact time spent constructing each side of the interface is required for error-free allocation.

Figure 2.13:
Categorization of costs



The following classification criterion refers to the behavior of costs. Variable costs depend on an input measure, while fixed costs are independent (Heinen, 1985). The term 'input measure' is rather abstract as several drivers for variable costs exist, such as the number of units, batches, or repetitions (Cooper & Kaplan, 1988a; Labro, 2004, 2018). Figure 2.13 shows four different types of variable costs where the x-axis 'N' represents a generic cost driver, and the y-axis represents the resulting total costs[15]. Proportional costs increase linearly with the input quantity, such as material costs. If firms can get volume discounts, these costs become degressive. Learning effects are another example of degressive costs since the duration of single process time is decreased through increased efficiency due to repetition of the same task. Progressive costs show a slow increase for small driver values but a substantial increase for large values. This behavior is observed when firms acquire additional resources in the short run as a reaction to demand shocks. It is the foundation of capacity planning models (Balakrishnan & Sivaramakrishnan, 2002). Regressive costs decrease with increasing input and are seldom observed in practice. An example is lower energy costs for a fully loaded freezer than an empty one (J. Weber, 2022). Costs that do not depend on an input measure are called fixed costs. Often, they are noted as the costs for holding a specific capacity where capacity adjustments lead to a change in fixed costs. They are called absolute fixed costs if the capacity stays constant within a certain period (Friedl, 2010). Stepwise fixed costs are a special case of fixed costs, although they depend on an input measure. Such costs, for example, occur when capacity changes within a certain period and not between two periods (Friedl, 2010). The distinction between fixed and variable costs results from the short- and long-run view of costs. According to the RBV, resources are tied to a firm in the short run (Caves, 1980; Wernerfelt, 1984) and cannot be changed quickly, resulting in fixed costs as they occur independently from an input measure (Horngren et al., 2012). In the long run, however, firms can control the type and quantity of resources they acquire. Therefore, the fixed costs for holding specific resources turn into variable costs in the long run as they now depend on an input measure (Cooper & Kaplan, 1988a; Kaplan & Cooper, 1992). The third level classifies costs based on their hierarchy, extending the traditional view of fixed and variable costs (Labro, 2018). The hierarchy introduces a structure to a firm's resource consumption. Four hierarchical levels (unit, batch, product, facility) are noted in activity-based costing literature (Balakrishnan et al., 2012; Cooper & Kaplan, 1991). Material costs of machine hours are examples of unit-level costs. Batch-level costs arise for material setups or handling of lots, such as in the production, logistics, or purchasing department. Product-level costs, for example, result from product management activities or the product family design. Facility-level costs aggregate all costs that arise on a global firm level, such as the required energy to run production or all plant management costs[16].

---

[15] It is assumed that the cost rate is kept constant.

[16] Labro (2018, p. 281) further notes that *„the concept of the ABC hierarchy provides more fineness to the notions of fixed and variable used in traditional costing systems, where such fixity or variability is defined w.r.t. changes in volume. ABC terms such variable costs unit level costs, as the idea is that they are incurred with each unit produced and/or sold. Fixed costs are facility level costs. Traditional costing systems typically misclassify the intermediate levels in the hierarchy. Product-sustaining costs are lumped in with the fixed costs, disregarding the fact that they are variable with higher-level decisions in the firm,*

2.6.2 Cost Effects of Internal Complexity

The literature describes various cost effects of internal complexity, where a complete overview of all effects is more of a utopia. Luckily, there exist some substantial review studies summarizing the main effects. Labro (2004) analyzes the economic consequences of component commonality. A review by Brun and Pero (2012) shows the effects of product variety along the entire supply chain. Lyons et al. (2020) extend this study and provide evidence from a case study among 162 manufacturing firms. Internal complexity effects on costs and operational performance are reported by Trattner et al. (2019). Finally, an impact model introduced by Hackl et al. (2020) and later extended (Hackl, 2022) represents the cause-effect relationships between structural design patterns, such as component commonality and costs. Analyzing these studies indicates an overlap in reported effects. These studies mainly differ in their selection of sources as they refer to different research communities. The impact model, for example, is more related to the engineering community, whereas the other reviews are more related to the community of operational research. This work uses the impact model as a foundation since it visually represents the conceptual cause-effect relationships and highlights potential interactions. As mentioned above, the effects noted by the other reviews are integrated into the impact model. The integration of the review by Lyons et al. (2020) is highlighted. Integrating their study allows for extending the impact model beyond the product family design by adding product variety as another input.

The impact model, as introduced by Hackl et al. (2020), notes the economic consequences of internal complexity. It is based on literature discussing the effects of internal complexity and nine cross-industrial case studies. Figure 2.14 shows the impact model, already extended by product variety. It is separated into three columns and rows. Each row represents a cause-effect relationship in a specific department, such as product development, procurement, and production. Although more departments are involved during the product family life cycle[17], the early phases, such as product design, purchasing, and manufacturing, are the most cost-critical (Asiedu & Gu, 1998; Fixson, 2006; Rezaie, Ostadi, & Torabi, 2008; Skirde et al., 2016). The impact model is further separated into columns. The first column describes the causes of complexity-related costs. Oversizing or -design describes substituting two or more components with an oversized one. By doing so, the reuse of components across product variants is increased, further increasing the degree of component commonality (Hackl et al., 2020). According to Mertens (2020), overdesign is related to what Marc Meyer and Lehnerd (1997) call vertical leveraging. Product variety refers to the number of product variants in a firm's product mix (Trattner et al., 2019). The middle column shows the individual effects and their interactions as described in the literature. The following paragraphs discuss these effects for each department in more detail. Finally, the economic impact is shown in the last column, where Hackl (2022) differs between costs, time, quality, and risk as economic objectives. As argued, this work focuses on costs and product costing accuracy as they directly impact the

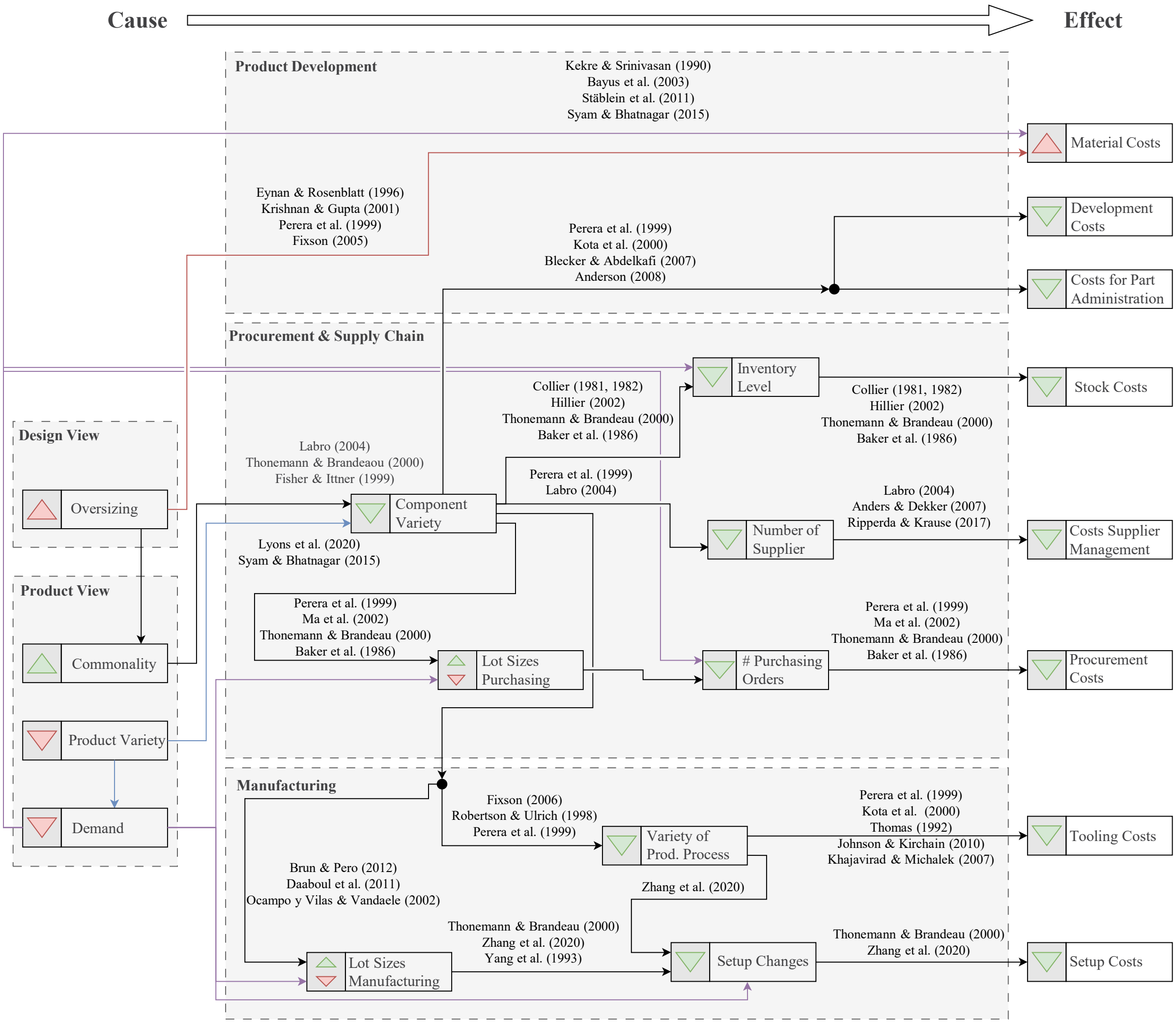
such as the expansion of the firm's product offering. Batch-level costs are lumped in with variable costs, ignoring that they are incurred no matter what the size of the batch (its volume) is."

[17] For an overview, see Terzi, Bouras, Dutta, Garetti, and Kiritsis (2010)

microeconomic objective. Therefore, indirect effects on costs such as decreased lead times (e.g., Jacobs, Droge, Vickery, & Calantone, 2011), decreased queuing delays, or higher forecast accuracy (e.g., Dogramaci, 1979) through an increasing level of component commonality (e.g., Kekre, 1987) are not part of this conceptual model.

Figure 2.14:

Direct cost effects, as reported by Hackl et al.'s (2020) impact model, extended by accounting literature (Labro, 2004) and production economics reviews (Lyons et al., 2020; Trattner et al., 2019)



During product family development, engineering decides on the right degree of overdesign. On the one hand, increased overdesign increases component commonality, which allows the reuse of components across products within as well as across a product family (Abdelkafi, 2008; Harland & Uddin, 2014; Otto & Wood, 2001; Robertson & Ulrich, 1998). The result of an increased overdesign is fewer distinct components (Fisher & Ittner, 1999; Labro, 2004), leading to a reduction in part management costs along the lifecycle as well as development costs (D. M. Anderson, 2008; Blecker & Abdelkafi, 2007; Kota et al., 2000; Perera, Nagarur, & Tabucanon, 1999). While component overdesign allows the replacement of individual components by a single, oversized one, often, the overdesigned component has a higher volume and mass or fulfills more functions than required (Fixson, 2005; Hackl et al., 2020), resulting in higher material costs (Krishnan & Gupta, 2001; Perera et al., 1999). Therefore, Eynan and Rosenblatt

(1996) criticize early studies on component commonality, which assumed cost equality for the oversized component with the ones it replaces.

In the procurement and supply chain department, fewer material numbers reduce the safety stock due to risk pooling, the cycle stock (Collier, 1981, 1982; Hillier, 2002; Thonemann & Brandeau, 2000), as well as the number of suppliers (Labro, 2004; Ripperda & Krause, 2017). For example, S. W. Anderson and Dekker (2009) note that the supplier selection process is cost-intensive for firms. When component variety is reduced, firms can reduce the number of suppliers, resulting in lower costs for maintaining the supplier relationship. Increased commonality increases the order volume for each component, increasing order lot sizes and leading to a reduction of purchasing orders and, therefore, order costs (Baker, Magazine, & Nuttle, 1986; Ma, Wang, & Liu, 2002; Thonemann & Brandeau, 2000).

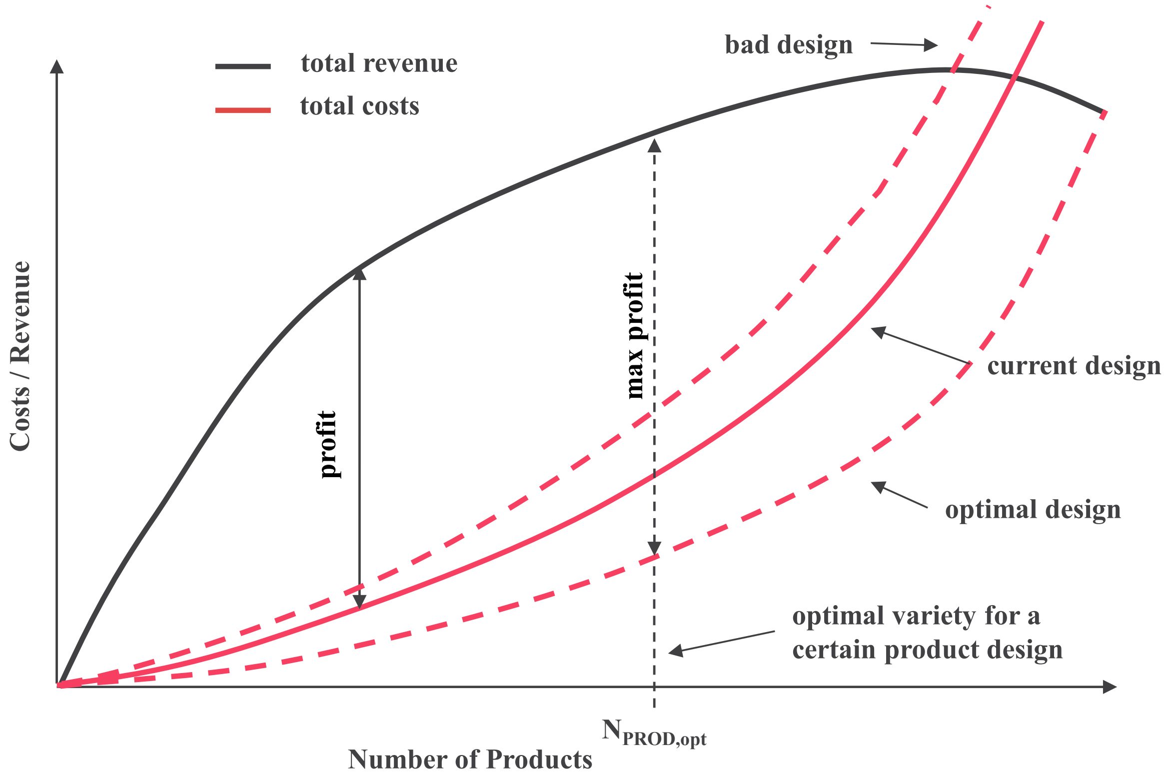
In manufacturing, the reduced component variety positively affects the setup costs. Decreased variety results in larger manufacturing lot sizes (Brun & Pero, 2012; Daaboul, Da Cunha, Bernard, & Laroche, 2011; Ocampo y Vilas & Vandaele, 2002) and reduces process variety (Fixson, 2006; Perera et al., 1999; Robertson & Ulrich, 1998; Thonemann & Brandeau, 2000; Yang & Deane, 1993). L. L. Zhang et al. (2020) define a process variant as a change in machines, setups, or tools. They argue that producing fewer components in higher lots decreases the probability of setup and tool changes. Decreasing tooling costs are noted as a second cost effect associated with increased component commonality (M. D. Johnson & Kirchain, 2010; Khajavirad & Michalek, 2007; Kota et al., 2000; Perera et al., 1999; Thomas, 1992). Tooling equipment is necessary to place components into manufacturing machines, transport components within the shop floor, or hold components during assembly.

Besides the product family design, product variety is another cause of costs. While commonality and oversizing describe the product design, product variety covers the aspect of component combinability[18]. An overview of the effects of product variety on costs is provided by the work of Lyons et al. (2020). In contrast to component commonality, the literature reports only cost-increasing (negative) effects caused by an increased product variety. Positive effects of product variety, such as increased customer satisfaction, increased sales and market share, and competitive advantages lay on the external side (Lyons et al., 2020). Product variety increases component variety if newly introduced products require new components (Lyons et al., 2020; Syam & Bhatnagar, 2015). A decrease in product variety reduces component variety only if all products using a specific component are excluded from the product mix. Fewer components trigger further effects that are already described in the previous paragraphs. The original impact model does not mention the positive association between product variety and total demand (Bayus, Erickson, & Jacobson, 2003; Kekre & Srinivasan, 1990; Syam & Bhatnagar, 2015). By introducing new products, firms increase the number of selling opportunities, aiming to increase revenue. However, the dependency of product variety and demand is not linear (see Figure 2.15). Using empirical data from a car manufacturer, Stäblein et al. (2011) show that 20% of the offered product variety accounts for 40 to 70% of the overall sales volume, and the top 10% of the revenue is generated by 30 to 55% of the product variety.

[18] Remember, products are defined as a set of functional requirements, components, processes, and resources.

It indicates that firms must deal with some high-volume product variants and many low-volume variants. An empirical analysis in section 3.5.3 confirms this picture. Since product managers prefer to introduce high-volume products before entering a niche with many low-volume products, the revenue curve shows a steeper increase under low levels of product variety. In a case study by Tan, Netessine, and Hitt (2017), the authors show that increasing product variety further leads to demand diversification. Under extremely high levels of product variety, a decrease in revenue is investigated (e.g., Gourville & Soman, 2005; Iyengar & Lepper, 2000; Kuksov & Villas-Boas, 2010; Villas-Boas, 2009; Wan, Evers, & Dresner, 2012). This phenomenon occurs as customers face an 'overchoice' so-called feature fatigue (Thompson, Hamilton, & Rust, 2005) or when variety overwhelms firms so they cannot deliver the appropriate quality anymore (Hackl et al., 2020).

Figure 2.15:
The damping effect of the product design (e.g., degrees of commonality) on total costs and profit.



The red curve in Figure 2.15 shows total costs as a function of product variety. Product design literature claims a progressive increase of costs with increasing product variety (e.g., Ripperda & Krause, 2017), where the principles of economies of scale and scope indicate a degressive curve. Later in this work (section 4.6), conditions leading to degressive, linear, or progressive costs under increasing product variety are discussed. The exact curve is not important for now since the following effects apply to both degressive and progressive curves. Demand diversification is neglectable under small levels of product variety; therefore, each additional variant increases revenue significantly. This leads to increased profits up to the point of optimal product variety ($N_{PROD,opt}$), where the maximum difference between total costs and revenue is reached. Every increase in product variety beyond this point will reduce profits since costs increase more than revenue. This is caused by increased demand diversification, resulting in many low-running products. In a recent case study, Santos et al. (2020, p. 1502) observe that *“the introduction of new products requires an in-depth analysis of the existing mix of products, while working on a simplified portfolio could bring more profitability”*.

Besides the ideal product variety, product design adjustments are another lever to maximize profitability as they reduce costs. The total costs are reduced if the monetary advantages of increased economies of scale due to increased component commonality outweigh the disadvantages of increased material costs due to components’ overdesign. Therefore, a good product design is seen as a lever to dampen the cost increase (Ripperda & Krause, 2017; Syam & Bhatnagar, 2015). However, by changing the product design, the optimal level of product variety is shifted, too. For example, a study by Thonemann and Brandeau (2000) shows that component commonality affects product variety. It highlights that questions of optimal product design are connected to product mix decisions.

Firms can maximize their profits by optimizing both the product design and the product mix. On a conceptual level, this section provides an overview of individual effects leading to economic advantages. By far, the model raises no claim to completeness but still shows the non-trivial interactions between product design, product variety, and costs. Overdesign, for example, has cost-increasing and cost-reducing effects, which further depend on product variety. Summarizing the existing literature investigating the cost effects reveals three types of studies. While review studies (e.g., Hackl et al., 2020; Labro, 2004; Perera et al., 1999; Trattner et al., 2019) provide an overview of existing cost effects, they are conceptual and, thus, limited in investigating effect interactions. Empirical studies (e.g., Lyons et al., 2020; Ripperda & Krause, 2017; Santos et al., 2020) report detailed cost effects but are limited to a single or handful of cases. Self-critically, Santos et al. (2020) highlight a lack of case studies’ generalizability. Finally, analytical studies allow for generalization. However, they are limited to a specific aspect, such as the effects of component commonality on inventory costs (Thonemann & Brandeau, 2000), setup costs (L. L. Zhang et al., 2020) or the assembly process (Ma et al., 2002). What seems to be missing is a study connecting these ends and combining the individual advantages. Analytical studies, for example, provide detailed models that can be used to operationalize many cause-effect relationships mentioned in review studies. Empirical studies provide fruitful insights on whether certain assumptions reflect practical settings. Following the call of recent studies (Hackl et al., 2020; Lyons et al., 2020; Trattner et al., 2019), section 4 operationalizes the conceptual impact model as presented in this section. By integrating empirical assumptions, this work provides more empirical evidence (Labro, 2004; Santos et al., 2020) and is a step toward a more general understanding (Stäblein et al., 2011). This work is not limited to the cost effects of internal complexity. As mentioned, internal complexity also affects the error in product costing systems. The following section, therefore, introduces the basic principles of product costing systems as well as errors in product costing.

2.6.3 Costing Systems and their Role for Decision-Making

Standard textbooks define costing systems as a set of rules for measuring and allocating costs on cost objects (Friedl, 2010) that are placeholders for any entity for which costs want to be measured (Horngren et al., 2012). Costing systems are seen as suppliers of information within firms[19]. Based on the information receiver, the type and design of the costing system can vary strongly (Friedl, 2010). Costing systems for long-run capacity planning require other cost objectives as costing systems for short- or medium-run decisions. Another decision layer is the desired level of costing system sophistication. Several authors (Al-Omiri & Drury, 2007; Alsayegh, 2020; Brierley et al., 2001; Hughes & Gjerde, 2003; Rezaie et al., 2008; Shields, 1995) identify factors influencing the choice, adoption, and design of product costing systems in practice.

Aiming to support decision-making within firms (Balakrishnan & Sivaramakrishnan, 2002), costing systems provide cost information used for *"capacity acquisition, planning and pricing decisions"* as well as *"customer portfolio management, inventory management, and competitive decisions"* (Labro, 2018, p. 307). Balakrishnan et al. (2012) note that capacity acquisition, allocation and product mix decisions are some of firm's information-extensive and complex decisions. Other aims of costing systems are inventory valuation or the identification of cost reduction opportunities (Innes, Mitchell, & Sinclair, 2000; Labro, 2018) [20]. Literature notes that more cost information (quantity or quality) increases the accuracy of reported costs, enabling better decision-making (Balakrishnan et al., 2011; Labro & Vanhoucke, 2007). However, it follows the legitimate question of why firms do not gather as much information as possible to maximize the amount of information for decision-making. The answer is the price tag on information. Increasing the amount of information results in higher costs as more data or data with higher granularity must be gathered and processed by the costing system. As a result, a particular costing systems design represents the trade-off between *"accuracy and the cost of accuracy"* (Labro & Vanhoucke, 2007, p. 940) or the assumed benefits of more accurate product costs versus the costs of information collection (Balakrishnan et al., 2011).

Caused by the fact that resources are semi-permanently tied to a firm (Caves, 1980; Wernerfelt, 1984), literature differs between short- and long-running aims of costing systems. In the short run, decision-makers cannot control resource capacity (Balakrishnan et al., 2012). Therefore, decisions on efficient resource allocation are primarily on interest. In the long run, decisions regarding resource acquisition (type of resources and quantity), strategic resource prices, or supply and product mix decisions become important (Labro, 2018). Costing systems support decision-making at both stages. In the short run, costs are not merely the result of pure resource consumption but also of the (partial) non-consumption of available resources already allocated by firms, resulting in unused capacity in a certain period. Kaplan and Cooper (1992) demonstrate this aspect with a simple example. Under full usage of available capacity, a firm with periodic costs of 25.000 $ can produce 1.250 output units (max), resulting in 20 $

[19] Other accounting systems may have different objectives and other recipients who lay outside the firm. An example is annual financial statements, which must be reported to shareholders.

[20] German textbooks such as Friedl (2010), for example, further differ between objective aims (German: "Sachziel ") and formal aims (German: "Formalziel "). Objective aims define the purpose of information, whereas formal aims describe the goal that should be achieved by collecting this information. In the context of customer profitability, costing systems also provide information to determine the cost-to-serve for individual customers, such as the periodical costs noted by Foster and Gupta (1994) or acquisition costs noted by Bjørnenak and Helgesen (2013). Boer et al. (2001) further note that supplier selection is also partially based on cost systems' information.

per unit. Suppose the firms now produce only 1.000 units in a period. In that case, the costs of used capacity are estimated at 20.000 $ as each unit costs 20 $ and the costs of unused capacity (250 units) are estimated at 5.000 $. These 5.000 $ are fixed costs and should be avoided by firms (e.g., using unused capacity for other activities) to increase efficiency. If capacity is already fully used, information provided by costing systems allows decisions on allocating available capacity best (Labro, 2018). In the long run, capacity allocation (e.g., which resources are required at which amount), the supplier selection for those resources on factor markets at specific prices and product mix decisions become important (Banker & Hughes, 1994; Kaplan & Cooper, 1992; Labro, 2018). Balakrishnan and Sivaramakrishnan (2002) summarize this long-run decision-making problem as the "Grand Program". Solving this problem minimizes opportunity costs by determining the optimal quantity of resources that need to be acquired under certain constraints. The solution to this problem is non-trivial, as cost-based decisions such as pricing or resource allocation in the current period $(t = n)$ are made based on cost information from the previous period $(t = n - 1)$, which are unlikely to be error-free[21]. A formulation of such a closed-loop discrete dynamic system is provided by V. Anand et al. (2017). These decisions are not only the most information-demanding, but they are also the most essential ones since resource allocation questions directly affect firms' microeconomic objectives.

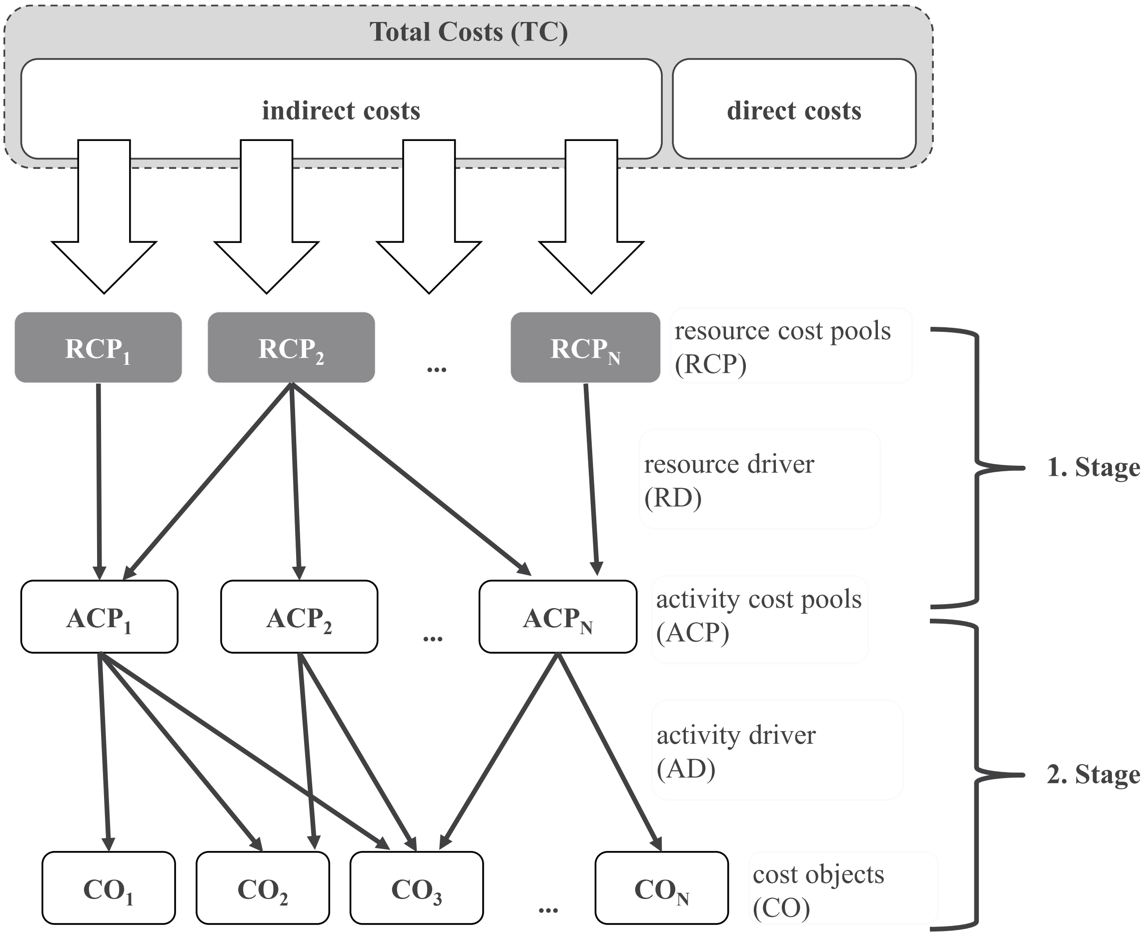
Costing systems have a wide field of applications. Product costing systems (PCS) are of primary interest for this work. PCSs report costs for each product (cost object), which firms can sell on product markets. These systems differ between direct and indirect costs. Direct costs are traced by costing systems as they can be directly assigned to costing objects. Indirect costs, however, need to be allocated by costing systems via an allocation heuristic as they cannot be traced directly to cost objects (Horngren et al., 2012; Labro & Vanhoucke, 2007)[22]. For example, tracing product family development costs onto individual products is challenging since engineering develops a product family rather than individual products. The process of cost tracing and allocating is called cost assignment (Horngren et al., 2012) and is formalized via a two-stage allocation process in literature (V. Anand et al., 2019; Balakrishnan et al., 2011; Labro & Vanhoucke, 2007, 2008; Mertens, 2020). The general structure of a two-stage PCS is shown in Figure 2.16, starting with resource cost pools (RCP), allocated via a resource driver (RD) onto activity cost pools (ACP) in the first stage. The second stage allocates ACPs via activity drivers on cost objects (CO).

---

[21] See section 2.6.4

[22] According to Ben-Arieh and Qian (2003), the following four allocation heuristics types exist: intuitive, analogical, parametric and analytical. Rezaie et al. (2008, p. 1048) note, *"The intuitive methods are based on the past experience of the cost estimator. The analogical methods estimate the cost of products using similarity to other products with known cost. The parametric methods estimate the total cost of a product from parameters that are usually used by the designers. These parameters influence the total cost in a known way, usually represented by a simple equation. Finally, analytical methods such as ABC allow evaluation of the total cost of a product from a decomposition of the work required into elementary tasks, operations or activities with known (or easily calculated) cost".*

Figure 2.16:
Formalization of a two-stage costing system. Syntax according to (Labro & Vanhoucke, 2007).



Product costing systems have various stakeholders within firms, making them an important instrument. However, product costs are unlikely to be error-free as their design is always a trade-off between accuracy and the resulting costs to achieve this accuracy — the next section, therefore, deep dives into errors in product costing systems.

2.6.4 Errors in Product Costing

Early and highly influential works[23] of Cooper and Kaplan (1988a) and Cooper and Kaplan (1988b) highlight the problems of product costing system accuracy in practice. They find that practice rejects the academic short-run perspective, which assumes only variable costs and that simple volume-based allocation systems over-cost high-volume and under-cost low-volume products[24]. As an implication, they suggest activity-based costing (ABC), which allows to define costs to be variable on unit-, batch-, product- or facility-level. Such systems can report more accurate product costs. However, they require more information compared to traditional costing methods. In the following years, many empirical studies investigated the

[23] Publications have more than 4,000 citations in sum, according to Google Scholar Search in February 2023.

[24] Other works (e.g., Rezaie et al. 2008) call volume-based PCSs also traditional costing systems.

challenges as well as the adoption of ABC in practice (e.g., Al-Sayed & Dugdale, 2016; Innes et al., 2000; Innes & Mitchell, 1995; Ittner, Lanen, & Larcker, 2002; Malmi, 1999; Schoute, 2011; Shields, 1995). Other studies discuss the selection of ABC cost drivers in analytical studies (e.g., Babad & Balachandran, 1993; Homburg, 2001). Latest studies use numerical experiments for a systematic investigation of PCS's robustness (Labro & Vanhoucke, 2008), the impact of different allocation heuristics (Balakrishnan et al., 2011), or the impact of measurement errors (Labro & Vanhoucke, 2007; Mertens, 2020; Mertens & Meyer, 2018) on its accuracy. As argued in section 2.3, numerical experiments have several advantages, making them a good choice for addressing such questions. Balakrishnan and Penno (2014) note that numerical experiments can analyze the complex interactions that occur at different stages of the allocation process. They allow the generation of different levels of available information, calculating the true benchmark costs under full information and reporting heuristic costs by reducing the amount of information. The latest achievement is the numerical framework of V. Anand et al. (2019), introducing a standardized model for future experiments (hereafter: ABL).

As mentioned, product costing systems collect information on costs, allocate (indirect costs) or trace (direct costs) them and report costs for each product. Those costs differ from the true benchmark costs since costing systems must deal with incomplete information and uncertainty or are restricted by their design choice (V. Anand et al., 2017; Balakrishnan et al., 2011). Therefore, a non-necessary but sufficient condition for a costing system to report the true costs is the existence of full information, which is difficult to obtain in practice [25]. Differences between the true benchmark costs (PC_B) and reported costs (PC_H) are defined as costing errors, whereas the work of Labro and Vanhoucke (2007) provides a summary. They define three different types of errors in PCS. Aggregation errors occur when non-identical resources are grouped into the same resource cost pools, or non-identical resource cost pools are aggregated into the same activity cost pool. Specification errors occur for resource and activity drivers, for example, when an allocation base that does not adequately reflect resource consumption is used. Measurement errors are induced at resource- (e.g., biased resource costs), on resource driver- (e.g., biased measurement of resource consumption), and on activity driver-level (e.g., biased measurement of activity consumption for products). Two widely used measures describe the error of product costing systems (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011; Labro & Vanhoucke, 2007, 2008). For the two product cost vectors, $PC_B \in \mathbb{R}^{N_{PROD}}$ and $PC_H \in \mathbb{R}^{N_{PROD}}$, costing systems' absolute error ($EUCD$) is defined as:

$$EUCD = \sqrt{\sum_{i=1}^{N_{PROD}} (PC_{B,i} - PC_{H,i})^2} \tag{2.15}$$

As a relative measure, the mean average percentage error ($MAPE$) is defined as:

[25] At this point, it is essential to note that the presence of full information is not necessary for a costing system to report the benchmark costs. Mertens and Meyer (2018) report that errors can compensate each other, which means that, by chance, true product costs are also possible under incomplete information. However, a costing system with complete information always reports the true benchmark costs.

$$MAPE = \frac{1}{N_{PROD}} \sum_{i=1}^{N_{PROD}} \frac{|PC_{B,i} - PC_{H,i}|}{PC_{B,i}} \quad (2.16)$$

Larger values of both $EUCD$ and $MAPE$ indicate costing systems that produce larger errors. Studies, however, argue that $EUCD$ and $MAPE$ are correlated (e.g., Balakrishnan et al., 2011).

$MAPE$ and $EUCD$ are system-level measures, reporting one value for the entire product mix. To explain the economic impact of product costing errors, under-costing (UC) and over-costing (OC), refering to the product level, are defined. Accounting defines under-costing as products having lower reported costs than their true costs and vice-versa for over-costed products (Horngren et al., 2012). A product is under-costed if benchmark costs are below 95% of its reported costs and is defined as over-costed if benchmark costs are above 105% of its benchmark costs, as shown in equations (2.17) and (2.18) (Labro & Vanhoucke, 2007). The two-sided 5% interval is chosen since minor errors are immaterial (Balakrishnan et al., 2011; Kaplan & Atkinson, 1998; Labro & Vanhoucke, 2007). For a given product mix with N_{PROD} individual products, the percentage of products being UC, OC or have no notable error ($< 5\%$) is always 100%.

$$UC = \begin{cases} TRUE, PC_H < PC_B * 0.95 \\ FALSE,\ else \end{cases} \quad (2.17)$$

$$OC = \begin{cases} TRUE, PC_H > PC_B * 1.05 \\ FALSE,\ else \end{cases} \quad (2.18)$$

Since some products are OC and others are UC, a cross-subsidization among products exists (Horngren et al., 2012). While OC products are reported to have higher costs, product prices might not reflect the proportional functional value of a product. Therefore, over-costed products are inefficient for firms since they indicate missed selling opportunities under the concept of price elasticity and perfect markets. If competitors with more accurate product costing systems enter the market, firms may drop OC products as they seem unprofitable (M. Gupta & Galloway, 2003; D. R. Hansen & Mowen, 2006). Under-costed products, on the other hand, are also dangerous for firms. PCSs report them as profitable; however, their true costs are higher and, in the worst case, even higher than their contribution margins. Horngren et al. (2012, p. 140) summarize: *"sales [of under-costed products] bring in less revenue than the cost of resources they use",* resulting in losses for the firm. Customers will notice the good price-to-functionality ratio (more bang for the buck) for such products as their prices are below or slightly above the mere resource consumption. Firms believe such products are still profitable as the product costing system gives the illusion of positive contribution margins (Cooper & Kaplan, 1988a). By conducting a numerical experiment, Homburg, Nasev, and Plank (2018) show that keeping under-costed products instead of erroneously dropping profitable products is much more dangerous for firms.

Dropping seemingly unprofitable products and being unaware of selling products with unnoticed negative contribution margins can lead to a vicious circle over multiple periods, as described by accounting and product management literature. Baxendale (2001), for example, notes that, in reality, profitable but over-costed products can suffer a quick death. Firms will increase product prices to make this product profitable again. However, as prices rise, demand will drop, making those products even more unprofitable due to decreased economies of scale. At the end of such circles, managers drop such products. However, it is important to note that such circles are not only caused by biased product costs[26]. Conducting numerical experiments with a multi-period firm model, V. Anand et al. (2017) show that firms with simple costing systems have a higher chance of entering a vicious circle where, at its end, a steady state with zero production is reached[27]. From a product management perspective, Schuh (2005) notes that firms may compensate for dropped and seemingly unprofitable products by increasing product variety, especially to address niches. As a result, firms' total costs increase due to decreasing economies of scale. From the perspective of biased product costs, single-product firms have a competitive advantage as their costs are allocated to only one cost object, reducing the potential for errors in product costing (Horngren et al., 2012). A case study among 234 U.K. firms conducted by Kennedy and Affleck-Graves (2001) shows the importance of accurate product costs. They find that firms using more accurate costing systems (such as ABC) make better investment and operating decisions, resulting in higher market value.

[26] For example, see Lemon, Zeithaml, and Rust (2001), Andrews, Cannon, Cannon, and Low (2009), and Cannon, Cannon, and Schwaiger (2012)

[27] See the discussion by V. Anand and Balakrishnan (2013)

3 On the Modelling of Product Families and their Complexity

3.1 Introduction

The previous chapter discussed the conceptual relationships between complexity, their variety-induced costs, and the potential impact on product costing systems' accuracy. This section goes one step further as it operationalizes internal complexity. Operationalizing internal complexity, however, is non-trivial since the literature notes problems in measures' validity and variety, leading to scattered literature on product family complexity measures (hereafter PFCM; Hennig et al., 2022). There exists a variety of measures that aim to measure complexity in the context of product family design (e.g., Ameri et al., 2008; Hennig et al., 2022; Sinha & Suh, 2018; Summers & Shah, 2010; Trattner et al., 2019) leading to challenges for both, practitioners and researchers. Practitioners face the challenge of choosing appropriate measures from a large pool of existing ones. For researchers, on the other hand, a variety of measures hamper the comparison of existing studies and replication (Hennig et al., 2022), which is a crucial aspect of fast theory development. Measure validity is another concern raised by different studies and is characterized by insufficient validation (Blecker & Abdelkafi, 2006; Sinha & Suh, 2018) and a large variety of existing validation approaches (Hennig et al., 2022). Valid measures are essential as they enable effective complexity management (Hennig et al., 2022) or support decision-making, such as allocating development resources (Jung et al., 2022).

Addressing PFCM's variety and validity issues, this section introduces a reduced set of measures to operationalize internal complexity. Operationalizing internal complexity is necessary to analyze its economic consequences and impact on product costing systems in line with Libby's predictive validity framework (see Figure 1.1). In doing so, measures reported in five substantial review studies are summarized, and their relation to complexities' dimensions (interrelatedness, diversity, multiplicity) is discussed. Integrating these measures in the numerical EAD model, derived from the conceptual EAD (see section 2.4.4), is the foundation for conducting numerical experiments. In these experiments, various unique EAD designs are created, and PFCMs' values are calculated - correlation analyses allow identifying similarities among measures. When two measures show a high correlation, they are proxies, allowing one to drop since the amount of additional information provided by a second highly correlated measure is negligible. As a result, a reduced set of PFCMs is reported, which minimizes the informational loss caused by dropped measures. By conducting these steps, this section provides a blueprint for addressing the issue measures' variety (Hennig et al., 2022). The proposed set of measures further reflects that no 'one-fits-all' measure adequately captures all facets of complexity (Jung et al., 2022). Second, according to the practice of good modeling (see Robinson, 2008), a good model should provide a simple explanation, which means that keeping highly correlated PFCMs increases model complexity but does not generate new insights due to the low amount of added information provided by those measures. Besides measures' variety and minimizing model complexity, dropping such measures also has a statistical motivation. Highly correlated measures lead to collinearity, a critical point for several statistical analyses, such as regression, as it results in unreliable effect sizes (Hair, Black, Babin, & Anderson, 2014). In contrast to many existing studies and following the work of Hennig et al. (2022), this section discusses PFCMs using artificially generated product family designs. Research on internal complexity is strongly driven by case studies (e.g., Bonjour & Micaëlli, 2010; Kashkoush & ElMaraghy, 2017; G. Kim et al., 2016; Li, Ni, Zhang, & Liu, 2021; Min, Suh, & Hölttä-Otto, 2016). Although such examples support readers' problem understanding, they do not allow them to compare measures among each other on a larger scale (see Balakrishnan & Penno, 2014). While case studies represent only a small proportion of potential product designs, large-scale numerical experiments can generate thousands of observations, leading to a denser discretization of the design space and more reliable results. This section, however, is not just a replication of Hennig et al.'s (2022) work as it differs in several points. First, PFCMs beyond the design structure matrices (DSM) are investigated, and second, more diverse designs are created, leading to better discretization. For example, Hennig et al. (2022) differ only between strictly random or modular designs, while this study generates mixed patterns.

After dropping measures based on correlation results, their construct validity is checked. Construct validity describes the ability of a construct (e.g., a complexity measure) to reflect the underlying objective (O'Leary-Kelly & Vokurka, 1998). According to the motto 'measure it to manage it' [28], valid complexity measures become essential as they allow for effective management of internal complexity and support decision-making such as the allocation of development resources (Hennig et al., 2022; Jung et al., 2022; Lindemann et al., 2009). In the context of PFCMs, the highly cited works of Weyuker (1988), as well as Briand, Morasca, and Basili (1996) provide criteria for valid measures. While selecting a reduced set of PFCMs, their validity against those criteria is checked, leading to a consolidated set of measures in terms of independence among measures.

The remainder of this chapter is structured as follows. Section 3.2 discusses the product family design modeling based on the conceptual EAD introduced earlier in this work and empirical evidence. A case study demonstrates the ability of the further extended EAD to represent product family design. Section 3.3 then connects the EAD model with the concept of complexity and discusses the validation criteria. Based on the existing literature discussing PFCMs, section 3.4 introduces the identified measures. Section 3.5 introduces a computational model to conduct numerical experiments. Numerical experiments are conducted to analyze the similarities among PFCMs using the computational EAD framework in section 3.6. This section further

[28] Literature does not agree whether this notion was made by Peter Drucker, Robert Kaplan, or other famous economists. On the other side, Edwards Deming notes this as a costly myth.

proposes a reduced set of validated PFCMs, which allows the description of firms' internal complexity (section 3.6.3). Finally, implications for practice and research are derived from the results of this analysis in section 3.7.

3.2 Modelling Product Family Design

Introduced in section 2.4.4, Mertens's (2020) conceptual EAD indicates the starting point for the section. Separated into three main parts, the conceptual EAD consists of the product portfolio definition (3.2.1), the product family design (3.2.2), and the production technology (3.2.3). Differences between the Axiomatic Design and the EAD, as well as its limitations, are discussed in section 3.2.4. Since the AD is widely seen as a framework supporting decision-making, a case study highlights a typical multi-attribute decision-making problem during product development (3.2.5).

3.2.1 Product Portfolio Definition

Defined as the mapping between customer needs and functional requirements, the EAD starts with the product portfolio definition (Mertens, 2020)[29]. Firms aim to offer an optimal product variety to maximize their profit by increasing revenue (a function of product mix and demand), decreasing costs (a function of resources and prices), or both. Assuming perfect markets, firms can increase demand by increasing product variety. However, several authors note that an increased product mix increases costs (and therefore product prices), where, as a result, products are less attractive to customers (ElMaraghy et al., 2013; Jiao, Tseng, Duffy, & Lin, 1998; Tolonen, Shahmarichatghieh, Harkonen, & Haapasalo, 2015). Therefore, firms' product (portfolio) management[30] search for the right product mix (Tolonen et al., 2015).For example, Jiao et al. (2007) highlight the need to detail the mapping between customer needs and functional requirements. The discrete choice theory (DCT) solves this problem of how individuals choose between alternatives (S. P. Anderson et al., 1992; Ben-Akiva & Lerman, 1985). DCT, therefore, provides the theory to detail the link between EAD's customer and functional domain.

In doing so, the free product mix ($P_{FD,free}$) is defined as a matrix including all unique products (rows), which are described by the combination of functional requirements (columns) under the restriction that a product with no functional requirements (empty product) is not allowed. With $E = FR_1, \dots, FR_N$ being the set of functional requirements, the free product portfolio is given as:

$$P_{FD,free}(X) = \{E \subseteq X : E \neq \emptyset\} \tag{3.1}$$

$P_{FD}(X)$ defines the power set of the E, while the product without any functional requirements is excluded ($E \neq \emptyset$). With three functional requirements ($E = FR_1, FR_2, FR_3$), the free product matrix is given as:

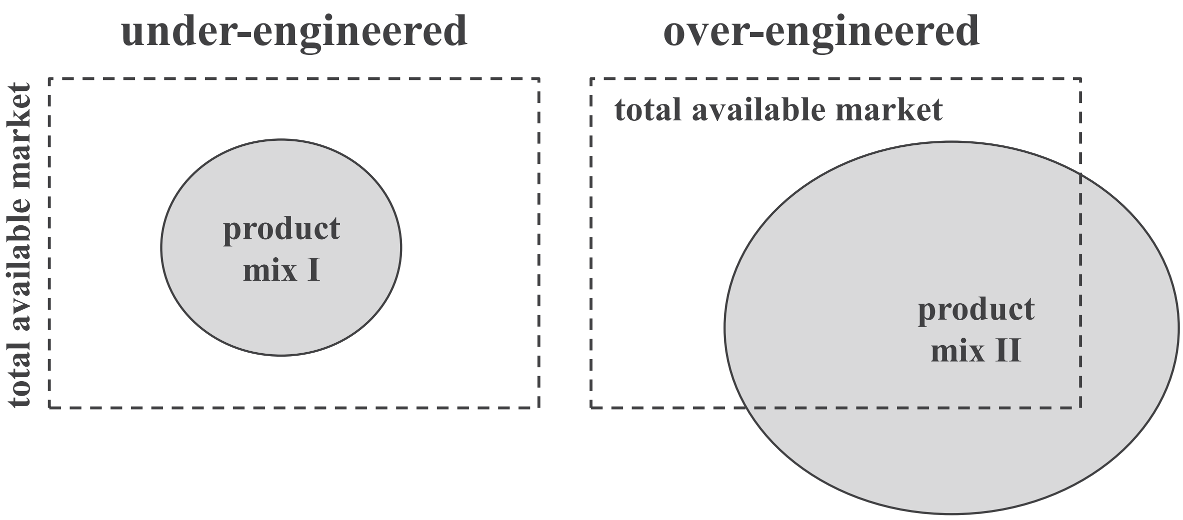
[29] Also known as product family positioning in Jiao and Zhang (2006) or Jiao et al. (2007)

[30] Product portfolio management is responsible for the management above the product family level. However, this work assumes a firm with a single product family. Therefore, these terms are equivalent.

$$P_{FD,free} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad (3.2)$$

Excluding the empty product, seven variants remain in the product mix, representing the so-called free or unconstraint product mix. However, as argued, the free combination of features is only sometimes the best economical choice as it does not necessarily represent the point of maximum profit under limited resources. For example, ElMaraghy et al. (2013) state that car manufacturers have more than 10^{17} individual product variants for a single product line. Since there are only eight billion ($8 * 10^9$) humans, there will always be variants that will never be sold. In practice, however, firms offer such an immense product variety due to uncertainty on customer needs. Figure 3.1 provides a graphical representation of this problem, which has been reported in different studies (Marti, 2007; Teboul, 1991). The rectangle represents the total available market for a random product (e.g., all car users). The dashed line indicates that the boundaries are fuzzy since firms have limited information on customer needs. Product mix I is under-engineered as it only partially covers the available market. Product mix II, however, is partially over-engineered as non-perceived products are included (Marti, 2007). For product mix one, each product is sold at least once. Product mix two contains products that are not requested by the market. However, none of the product mixes indicate an optimum. Under product mix one, firms miss selling opportunities, and under product mix two, firms have efforts such as for the development of products that are never sold.

Figure 3.1:
Visual representation of an under- and over-engineered product mix according to Teboul (1991) and (Marti, 2007)



To prevent over-engineering, firms can restrict their product mix. An unconstrained product mix $P_{FD,free} \in \mathbb{N}_{0,1}{}^{N_{PROD}\ x\ N_{FR}}$ may include non-profitable products (V. Anand et al., 2017), or firms' available capacity is used more efficiently to produce other products (Labro, 2018). The restricted product mix ($P_{FD,const}$), therefore, represents a subset of the free product mix.

Under small product mixes, the inclusion or exclusion of products can be done manually. For large product mixes, however, it is not possible to include or exclude 10^{17} products. Instead, rules for the combination of functional requirements are defined. The EAD stores these rules in $DSM_{FD} \in Z_{-1,1,\emptyset}{}^{FRn\ x\ FRn}$. Equation (3.3) shows a DSM_{FD} with three functional requirements, whereas equation (3.4) represents the resulting restricted product mix.

$$DSM_{FD} = \begin{pmatrix} \emptyset & \emptyset & 1;1 \\ -1;1 & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset \end{pmatrix} \tag{3.3}$$

$$P_{FD,const} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{3.4}$$

Function requirements' combinations are stored in DSM_{FD} in the conjunctive normal form. The matrix in equation (3.3) is equivalent to the following two clauses: if FR_2, then FR_1 and if NOT FR_1, then FR_3. Therefore, $P_{FD,const}$ does not contain products that have FR_2 without FR_1. Since the first clause is defined as $FR_2 \Rightarrow FR_1$ it does not forbit the presence of FR_1 while FR_2 is absent. In the current format, DSM_{FD} holds the two Boolean operators NOT and IMPLIES.

Customers respond to a given product mix P_{FD} as they have individual requirements. This customer choice process results in a demand vector ($DMD \in \mathbb{N}^{N_{PROD}}$), which holds the quantities for each product defined in P_{FD}. A theory that describes the customer choice process is the DCT. According to the theory, customers are defined by so-called part-worth utilities (V. R. Rao, 2014). These utilities are stored in a customer utility matrix ($C \in \mathbb{R}^{N_{CUST}\ x\ N_{FR}}$) where each row represents one customer and each column the utility value for each functional requirement. According to the DCT, utility values are a proxy for the attractiveness of a possible choice. Customers aim to maximize their utility values by choosing the product that fits their preferences best (S. P. Anderson et al., 1992, 1989; McFadden, 1974). According to Lattin (1987), overall utilities for a product mix P_{FD} are then given as :

$$U = C * P_{FD}{}^{T}, C \in \mathbb{R}^{N_{CUST} x\ N_{FR}}, P_{FD} \in \mathbb{N}_{0,1}{}^{N_{PROD}\ x\ N_{FR}} \tag{3.5}$$

The result is a utility matrix $U \in \mathbb{R}^{N_{CUST} x\ N_{PROD}}$ holding information on the attractiveness of products for each customer. Equation (3.6) shows a utility matrix for a product mix with seven alternatives and two customers.

$$U = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}^{T} = \begin{pmatrix} 1 & 0 & 1 & 1 & 2 & 1 & 2 \\ 1 & 1 & 2 & 0 & 1 & 1 & 2 \end{pmatrix} \tag{3.6}$$

The product utility matrix U shows that customer one has the same utility values for products five and seven and customer two for products three and seven. Since the DCT assumes that customers aim to maximize the utility, customer one will choose either product five or seven, and customer two will choose product three or seven. The example demonstrates that the current model allows customers to pick a product that overfulfils their needs (product seven). In theory, this implies that firms need to offer only a one-fits-all product that fulfills all function requirements at once. Such a product would maximize all customer's utility. Practice, however, tells us that firms do not offer such a product because of technical challenges and customers' willingness to pay, as it comes with a significantly larger price tag.

Therefore, product prices are added to the model. The concept of reference prices based on Helson's (1964) adaption-level theory captures this problem by adding a second term to equation (3.5) as done in the model of Han, Gupta, and Lehmann (2001). The utility values are determined as:

$$U = C * {P_{FD}}^{T} + \beta_{gain}(RP - PR)I_{gain} + \beta_{loss}(RP - PR)I_{loss} \tag{3.7}$$

with, β_{gain} and β_{loss} being scaling factors for how much a difference between product prices (PR) and customer's individual reference prices (RP) affect the utility. Two scaling factors are necessary as customers are more sensitive to losses than gains (Han et al., 2001). I_{gain} and I_{loss} are switches for the activation of the loss or gain term. They are defined as

$$I_{gain} = \begin{cases} 1, & RP - PR > \tau_{gain} \\ 0, & otherwise \end{cases} \tag{3.8}$$

$$I_{loss} = \begin{cases} 1, & PR - RP > \tau_{loss} \\ 0, & otherwise \end{cases} \tag{3.9}$$

where τ_{gain} and τ_{loss} are thresholds since small price changes *"would be neither noticed nor have an effect on consumers' choices"* (Han et al., 2001, p. 439).

Prior considerations on the customer choice process show the non-trivial character of this step, where the DCT and reference price are only two concepts in this context. Nevertheless, in order to reduce modeling complexity, an exogenous demand vector (DMD) is assumed. In doing so, this work follows common simplifications (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011)[31]. However, detailing the customer choice process in future research allows modeling firms' profitability as a function of costs to offer a product mix and its response under a certain market. First thoughts on this topic are reported by V. Anand et al. (2017).

3.2.2 Product Family Design

The EAD defines the product family design as the mapping between functional requirements and design elements (Mertens, 2020). Suh (1995) argues that design elements are an abstract placeholder for fulfilling the functional requirements, whereas this work uses the system view, which interprets design elements as components. This definition is strongly related to Ulrich's (1995) product architecture (PA), defined as the mapping between product functions, seen by the customer as products' behavior, and components as functional carriers. Widely accepted

[31] The study of V. Anand et al. (2017) is an exception here.

in research and practice, the PA provides a strong foundation for mapping functional and physical domains (Mertens et al., 2022)[32].

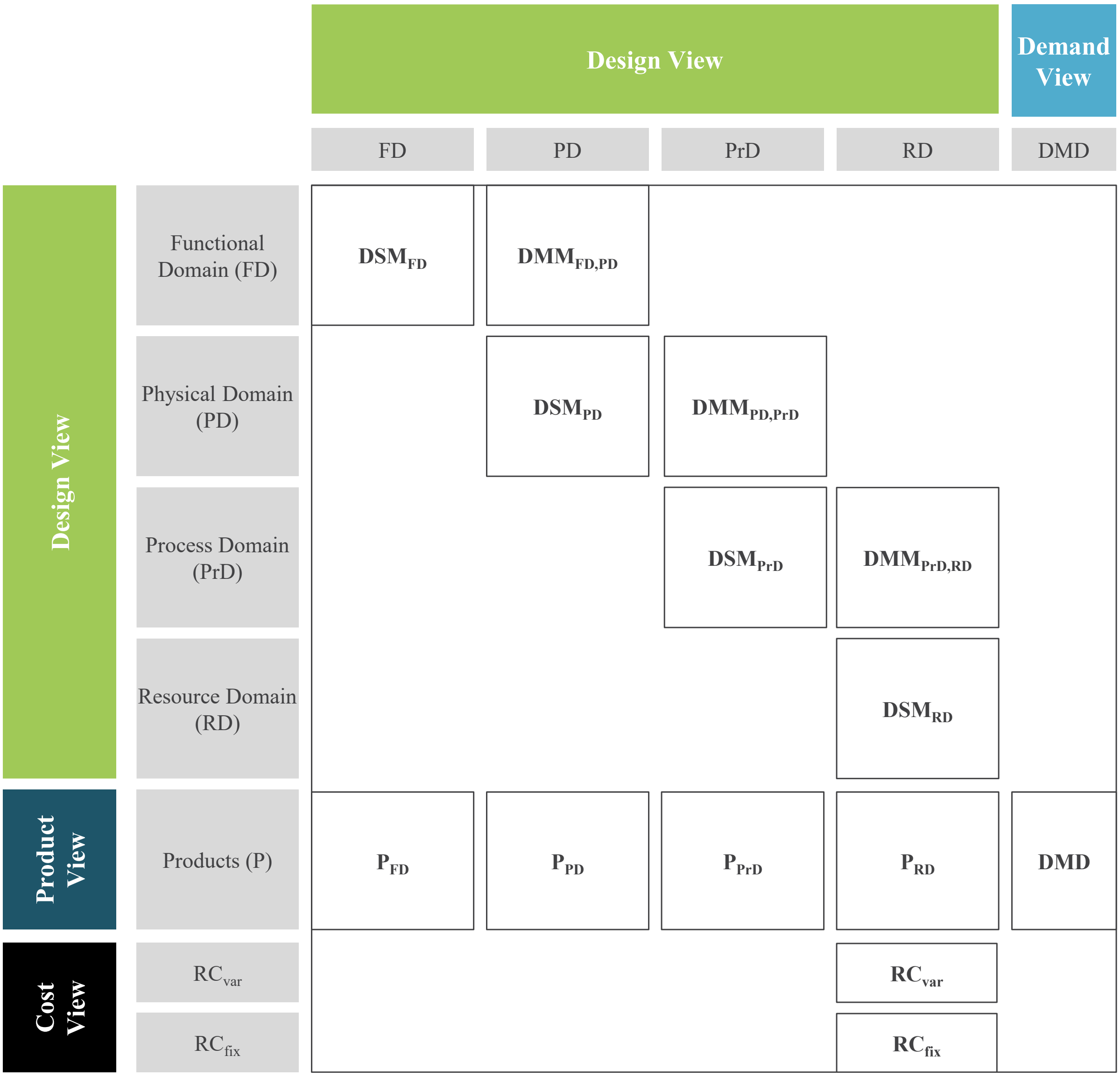
While the AD, as well as the conceptual EAD, describe the mapping between two domains by a single design matrix according to equation (2.10), this work further unifies the EAD with the concept of Dependency Structure Matrices (DSM)[33] and Domain Mapping Matrices (DMM). Several studies use these matrices to model product family design (Li et al., 2021; Morkos, Shankar, & Summers, 2012; Sawai et al., 2017; Sinha & Suh, 2018). An early study by Guenov and Barker (2005) further discussed adding DSMs to the AD. Introduced by Steward (1981), the DSM is a square matrix representing dependency information of elements within the same domain. Therefore, they are also called intra-domain matrices (Maurer, 2007). Browning (2001) differs between static information stored in DSMs, such as component interfaces used for clustering (Pimmler & Eppinger, 1994; Whitfield, Smith, & Duffy, 2002) or classification of product designs (Sharman & Yassine, 2004), and time-based DSMs used for sequencing, such as project activities (Maheswari & Varghese, 2005; Shi & Blomquist, 2012). As an inter-domain matrix, the domain mapping matrix (DMM) represents the mapping between domains (Maurer, 2007). In this case, the mapping between the functional domain (FD, functional requirements) and physical domain (PD, components). $DMM_{FD,PD}$ is equivalent to the design matrix as defined in Suh's (1995) AD. It contains information on the product architecture as defined by Ulrich (1995).

DMMs and DSMs hold different kinds of information. While DMMs hold the inter-domain knowledge representing the 'how' (Suh, 1995), like the required components to enable product functions, DSMs hold the intra-domain knowledge such as the compatibility of elements (Bongulielmi, Henseler, Puls, & Meier, 2001; Puls, 2003). Several authors highlight the need for combining these concepts (Danilovic & Sandkull, 2005; Guenov & Barker, 2005; Maurer, 2007; Matthias Meyer et al., 2019; Puls, 2003). Single-domain approaches are suitable for structural investigation and computational methods. Complexity, however, arises from the interaction of inter- and intra-domain dependencies (Lindemann et al., 2009). Figure 3.2 shows the EAD with additional DSMs in its design view. A case study by Sawai et al. (2017) uses such an approach to model the product family design. However, their study is limited to the functional and physical domains, whereas the EAD further includes the process and resource domains. Other case studies reduce product family design to the DSMs (e.g., Sinha & Suh, 2018; Kashkoush & ElMaraghy, 2017; Li et al., 2021) or DMMs (Jung et al., 2022).

[32] Several case studies use the concept of the product architecture. For example, see: John Paul Macduffie (2013), Sawai et al. (2017) or Robertson and Ulrich (1998)

[33] In literature also noted as Design Structure Matrices.

Figure 3.2:
Matrix representation of the EAD used in this work. The design view contains additional Dependency Structure Matrices holding intra-domain information, while the Domain Mapping Matrices (DMM) are equivalent to design matrices of the Axiomatic Design.



According to the matrix representation of the EAD, the entire model is separated into a design view holding the product family design and production technology, according to Mertens (2020). The design view contains all element dependencies within a domain (DSM) and across domains (DMM). The latter is equivalent to design matrices defined in Suh's (1995) AD. They contain information on the '*hows*'. In total, the EAD consists of four DSMs that define the dependencies among functional requirements in terms of configurational constraints (DSM_{FD}), the dependency among components (DSM_{PD}), processes (DSM_{PrD}), and resources (DSM_{RD}). DMMs and DSMs are directed matrices since they store information on elements dependencies. The component-component dependency matrix (DSM_{PD}) holds only natural numbers, including zero ($DSM_{PD} \in \mathbb{N}_0{}^{N_{PD}\ x\ N_{PD}}$), since physical components require positive integer proportions[34]. Non-integer entries occur in the process (DSM_{PrD}) and resource mapping (DSM_{RD}). For example, a process can require 1.5 units of another process. Equation (3.10) shows an example component-component mapping matrix (DSM_{PD}), which describes the dependencies of three components $DP = DP_1, DP_2, DP_3$ given as:

$$DSM_{PD} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 2 & 1 & 0 \end{pmatrix} \tag{3.10}$$

DSM_{PD} defines that DP_1 requires one unit of DP_2, DP_3 requires DP_1 twice and DP_2 once. Integrating DSMs to the EAD allows for modeling indirect element dependencies. For example, let $DMM_{FD,PD}$ be the product architecture for two functional requirements given as:

$$DMM_{FD,PD} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.11}$$

then, component DP_2 does not depend on any functional requirement, however it is necessary for the product to work as its usage across products is defined by DSM_{PD}. A practical example would be a standardized frame for a machine. A customer does not choose the frame directly. Instead, the frame results from the chosen functional requirements, leading to a specific set of components. A second reason for integrating DSMs is their practical relevance. Several case studies report a product design using DSMs (Bonjour & Micaëlli, 2010; G. Kim et al., 2016; Li et al., 2021; Morkos et al., 2012; Pandremenos & Chryssolouris, 2011; Sawai et al., 2017; Sinha & Suh, 2018). By including DSMs, the mapping between two domains extends the domain mapping to a two-stage approach compared to the classical AD theory, as noted in equation (2.10). The mapping between functional requirements and components is then defined as follows. Let $P_{FD} \in \mathbb{N}_{0,1}{}^{N_{PROD}\ x\ N_{FD}}$ be the product matrix representing the products in rows and functional requirements in columns. Multiplication with the inter-domain matrix $DMM_{FD,PD}$ leads to a physical product matrix ($P_{PD,1} \in \mathbb{N}_{0,1}{}^{N_{PROD}\ x\ N_{PD}}$) representing the direct dependencies between functional requirements and components.

$$P_{PD,1} = \begin{cases} 1, & P_{FD} * DMM_{FD,PD} > 0 \\ 0, & else \end{cases} \tag{3.12}$$

The resulting product matrix $P_{PD,1}$ is a binary matrix reflecting the fact that components can hold several functions simultaneously, such as in integral designs (Ulrich & Eppinger, 2016). Therefore, the functional and physical domain mapping does not follow an additive rule. This applies only to functional and physical domain mapping ($DMM_{FD,PD}$) since abstract functional requirements are manifested as physical elements at this stage. In the second step, $P_{PD,2} \in \mathbb{N}_0{}^{N_{PROD}\ x\ N_{PD}}$ is derived as:

$$P_{PD,2} = P_{PD,1} * DSM_{PD} \tag{3.13}$$

For completion of the FD-PD mapping, the final product matrix is assembled by choosing the element-wise maximum of both product matrices as:

[34] This assumption might change if physical elements are interpreted as service building blocks. Nevertheless, what remains is the assumption of positive numbers.

$$P_{PD} = \max\left(P_{PD,1,ij}, P_{PD,2,ij}\right), for\ i = 1 \ldots N_{PROD}\ and\ j = 1 \ldots N_{PD} \qquad (3.14)$$

Again, the elementwise maximum covers that components do not sum up.

3.2.3 Production Technology

EADs' production technology (PT) describes the mapping of physical elements to processes and further to resources (Mertens, 2020). Christensen and Hemmer (2006) define the PT as the set of all products and the production process, which describes the mapping between inputs and outputs. Starting with the physical product mix (P_{PD}) as an input for the production technology, the EAD formalizes the mapping between the physical and process domains. Typically, firms store information on the mapping between components and processes in routing sheets. Let $P_{PrD} \in \mathbb{R}_{\geq 0}{}^{N_{PROD}\ x\ N_{PrD}}$ be the product matrix in the physical domain, then the transition under consideration of process-process dependencies (DSM_{PrD}) is defined as:

$$P_{PrD} = P_{PD} * \left[\left(DMM_{PD,PrD} * DSM_{PrD}\right) + DMM_{PD,PrD}\right] \qquad (3.15)$$

Due to the introduction of intra-domain matrices, the mapping, as defined in equation (3.15), is extended by an additional interaction term. This term includes the dependency structure and domain mapping matrix ($DMM_{PD,PrD} * DSM_{PrD}$). The same principle is used for the mapping of P_{PrD} into the resource domain defined as:

$$P_{RD} = P_{PrD} * \left[\left(DMM_{PrD,RD} * DSM_{RD}\right) + DMM_{PrD,RD}\right] \qquad (3.16)$$

The matrix $P_{RD} \in \mathbb{R}_{\geq 0}{}^{N_{PROD}\ x\ N_{RD}}$ now represents usage of resources across products. It is equivalent to the resource consumption matrix as used in cost accounting models (e.g., V. Anand et al., 2019). While P_{FD} is defined as a binary matrix, P_{PD} as a matrix containing positive natural numbers (including zero), P_{PrD} and P_{RD} both contain positive real numbers.

According to the definition, the model assumes a Leontief model, widely used in cost accounting (V. Anand et al., 2019; Balakrishnan et al., 2011; Christensen & Hemmer, 2006). The Leontief function assumes linearity, additivity in production, no interaction among products (V. Anand et al., 2017; Christensen & Hemmer, 2006), and disallows substitution of inputs (Labro, 2018). However, it neglects the effect of labor, capital, and productivity as covered by other structures such as Cobb-Douglas or Gutenberg. However, the EAD assumes the Leontief structure for several reasons. Sickles and Zelenyuk (2019) note that the Leontief function dominates productivity literature, and several publications show its application in cost accounting (V. Anand et al., 2019; Schmidt et al., 2023). In distinction from other models such as Cobb-Douglas or Gutenberg, the Leontief model depends only on input quantity, which makes it easy to use (Diewert, 1971). While a Cobb-Douglas structure may reflect practical problems in more detail as it allows substitution, it also leads to more complex models, as Labro (2018) notes.

In line with the existing models, the PT contains a cost and demand view (V. Anand et al., 2019; Mertens, 2020). The resource product matrix (P_{RD}) is equivalent to the resource consumption matrix RES_CONS_PAT from equation (2.6), which allows the integration of their variable cost model. In the EAD, total costs (TC) are defined as the sum of variable costs (RC_{var}) and fixed costs (RC_{fix}) as:

$$TC = RC_{var} + RC_{fix} \quad (3.17)$$

While variable costs depend on an input measure, such as the amount of resources used or the number of output units, fixed costs (RC_{fix}) do not (Heinen, 1985). Modeling variable and fixed costs allow a distinction between costs that depend on the output, such as material costs, and those that do not, such as development costs. It is important to note that this distinction is only relevant in the short- and medium-run since, in the long-run, all costs are variable. Under a single-period view, total costs are written as the sum of all product costs (PC) multiplied by the demand vector $DMD \in \mathbb{Z}^{N_{PROD}}$, holding the product quantities.

$$TC = PC * DMD \quad (3.18)$$

Together with the total resource consumption TRC,

$$TRC = \sum_{i=1}^{N_{PROD}} P_{RD\,ij} * DMD_i \quad (3.19)$$

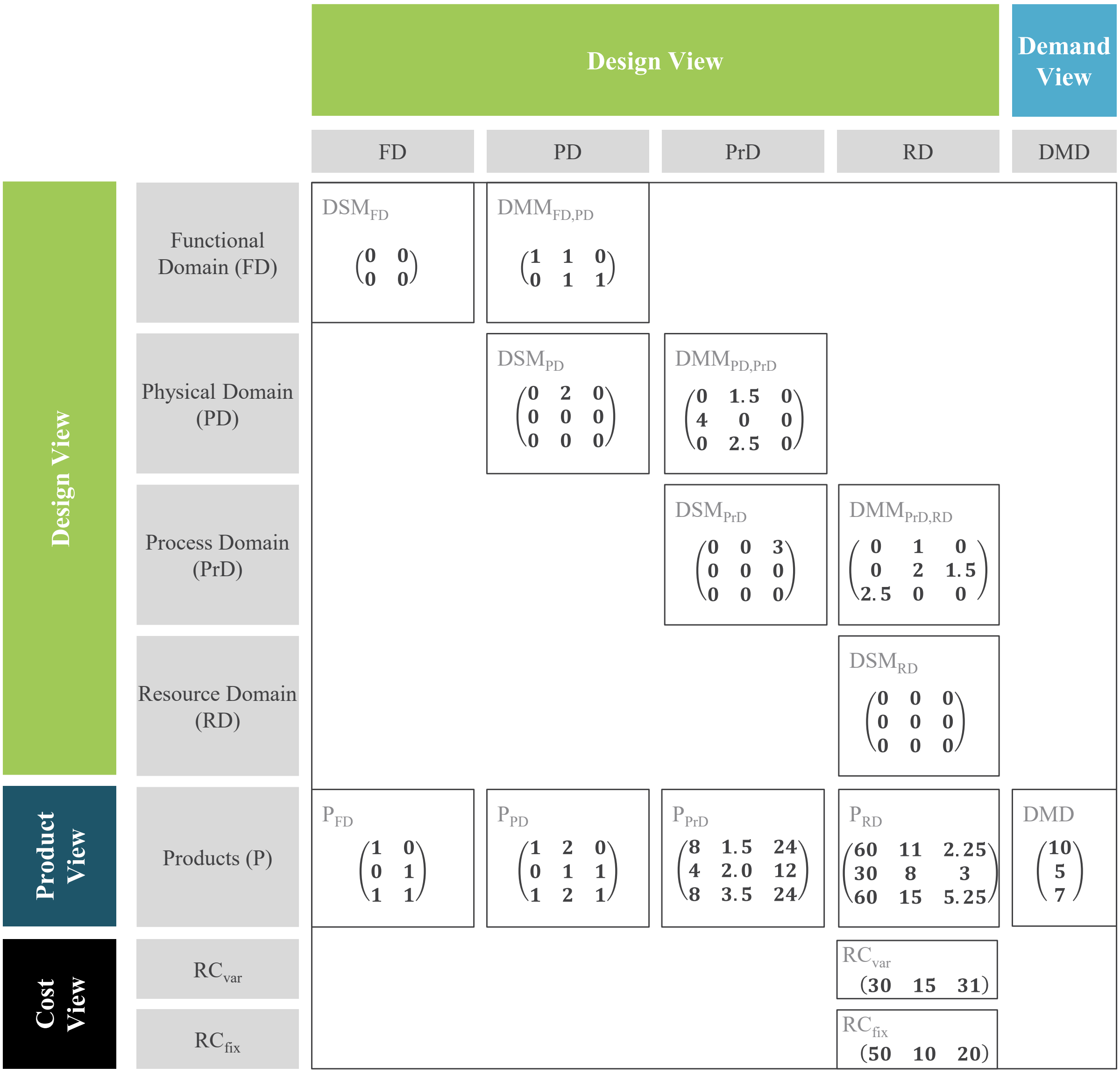
product costs ($PC \in \mathbb{N}^{N_{PROD}}$) are than defined as:

$$PC = P_{RD} * \frac{RC_{var} + RC_{fix}}{TRC} \quad (3.20)$$

Following the ABL model, it is assumed that capacity can be acquired instantaneously and, therefore, a zero-demand product does not induce costs. Demand vectors are called homogenous if all products have the same quantity and heterogenous if some products account for a large proportion of sales. For a better understanding of the model, Figure 3.3 shows an example design with two functional requirements under free combination ($DSM_{FD} = 0$). The product design further contains three components, processes, and resources. Total costs are given as 156 monetary units, and the total demand is 23 units. Under the given design, product costs are calculated as:

$$PC = \begin{pmatrix} 60 & 11 & 2.25 \\ 30 & 8 & 3 \\ 60 & 15 & 5.25 \end{pmatrix} * \begin{pmatrix} \frac{80}{1170} \\ \frac{25}{255} \\ \frac{51}{74.25} \end{pmatrix} = \begin{pmatrix} 6.73 \\ 4.90 \\ 9.18 \end{pmatrix} \quad (3.21)$$

Figure 3.3:
Exemplary product design which is based on two functional requirements, three components, three processes and three resources under the assumption of free combination.



3.2.4 Discussion of the EAD Model

The further extended EAD is not just another model as it unifies several concepts, such as Mertens's (2020) conceptual EAD, which, again, is based on Suh's (1995) Axiomatic Design theory and microeconomic assumptions. Domain mapping is defined as a two-stage approach that integrates DSMs and DMMs into the model. It allows for modeling indirect dependencies of domain elements. Although the discrete choice theory is not fully integrated into the model, the matrix representation provides a foundation to integrate the customer choice process in future research. Differences across the AD, Mertens's (2020) conceptual EAD and this work's operationalized model are summarized in Table 3.1 and discussed in the following paragraphs.

Table 3.1:
Differences between the classical AD, the conceptual EAD and the model used in this work.

Criteria	Axiomatic Design (AD)	Conceptual EAD	EAD used in this work
Domains	customer, functional, physical, process	customer, functional, physical, process, resource	
Domain Mapping	via a design matrix (DMM) reflecting only direct dependencies		via an inter- (DMM) and intra-domain matrix (DSM) a reflecting, both direct and indirect dependencies
Product View	-	added	added and extended by configuration rules defining the functional product mix
Cost View	-	assume that all costs are variable representing a long run perspective	extended by fixed costs allowing to represent also a short and medium run perspective
Demand View	-	demand as an abstract function of customer needs and the product mix	first, thoughts on the integration of the discrete choice theory

While the AD solely focuses on product design, Mertens (2020) suggests a more holistic perspective. Instead of optimizing a single product, firms take advantage if they optimize the design for the entire product family (Jiao et al., 2007; D. Krause & Gebhardt, 2018). Therefore, the EAD is a product family design model rather than a single product design model. Hence, a product view is introduced where products are defined as a set of domain elements. Product management, for example, describes products via functional requirements in specification sheets (D. Krause & Gebhardt, 2018; C. Weber, 2007), product engineering via components in a bill-of-material (Hegge & Wortmann, 1991), production engineering as a set of processes such as in routing sheets (Swamidass, 2000) or controlling via their resource consumption (V. Anand et al., 2019). Compared to the AD, the EAD explains how a functional product mix, in combination with an underlying design, leads to products described by components (P_{PD}), processes (P_{PrD}), and resources (P_{RD}). Doing so induces an interaction between the design and product view, which is not mentioned by either axiomatic design or accounting models. Combining engineering design, product management, and production theory, the EAD is the next step toward an information-summarizing framework, following the call of the literature (Guenov & Barker, 2005; Thevenot & Simpson, 2009). Integrating the resource domain into the product family design allows for determining the economic consequences of a given design (Mertens, 2020). While the AD evaluates a product design solely on its structural dependencies (Gonçalves-Coelho & Mourão, 2007), the EAD also estimates costs. Coupling a design with costs is essential since Suh (2001) notes that the costs of two equally good structural designs can differ. It further allows to answer multi-attribute decision-making problems as they occur all along the product development process (Gonçalves-Coelho & Mourão, 2007; Jiao et al., 2007; Kahraman & Cebi, 2009; Kulak et al., 2010). In combination with product, demand, and cost view, typical problems such as product mix or commonality decisions (Kumar et al., 2009) can be addressed.

In distinction to the conceptual model, this work's model reflects indirect element dependencies. While direct inter-domain dependencies (entries in DMMs) are the result of explicit design decisions according to the AD theory (Suh, 1990, 1995), intra-domain dependencies (DSM entries) reflect more technical dependencies between the elements, adding another source of complexity (Lindemann et al., 2009) with potential influence on costs and product costing systems accuracy. Connecting the EAD with the theory of DSM and DMM provides a structured approach to represent the product family design and could lead to a more standardized representation in case studies (e.g., Li et al., 2021; Sawai et al., 2017 or Sinha & Suh, 2018). Another difference to existing models in accounting studies (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011) is that patterns in P_{RD} are the result of an underlying product design rather than random correlation. A last extension is made in the cost view by adding fixed costs. They allow to model both the long-run ($RC_{fix} = 0$) and the short or medium-run ($RC_{fix} > 0$).

Although the model extends the work of Mertens (2020), it has some assumptions, simplifications, and open issues that need to be solved in the future. First, the EAD assumes a linear mapping between domains defined by matrix multiplication. More complex mappings, as they occur in practice, such as $FR_1 \wedge FR_2 \Rightarrow PD_1$, cannot be modeled yet. Integrating such dependencies requires a redefinition of the domain transition. EAD's production technology is defined by a Leontief production function, which assumes linearity, additivity in production, no interaction among products (V. Anand et al., 2017; Christensen & Hemmer, 2006) and disallows substitution of inputs (Labro, 2018). This indicates a current limitation where future studies can suggest an integration of other models, such as Cobb-Douglas. Another caveat is the single-period view, which does not allow for addressing the question of capacity acquisition and a dynamic model behavior such as that done by V. Anand et al. (2017). The model further assumes that effects from previous periods have subsided and do not influence the current state. Ideas on integrating the customer choice process are reported; however, this work will not go into detail due to its complexity. Nevertheless, future research should focus on integrating such models as they allow the modeling of product demand as a function of customer needs rather than an exogenous choice. This is essential for addressing questions of future robust product mixes. Besides these limitations, the further extended EAD model allows a more detailed representation of product family designs and their production technology. The following section demonstrates how the EAD can support typical decision-making problems in firms.

3.2.5 A Quadcopter Case Study

3.2.5.1 Introduction

A palm-sized quadcopter product family illustrates EAD's ability to support a typical decision-making problem[35]. The product family consists of twelve product variants described by the functional product matrix (P_{FD}), as reported in Table 3.2. While flight time and size are mandatory properties, video is optional, indicating there are product variants with no video. The physical design is defined by the domain mapping matrix $DMM_{FD,PD}$, which maps functional requirements to components and the dependency structure matrix DSM_{PD}, which defines the dependencies between components. Product families' DSM_{PD} is reported in Figure 3.5, together with the remaining design matrices.

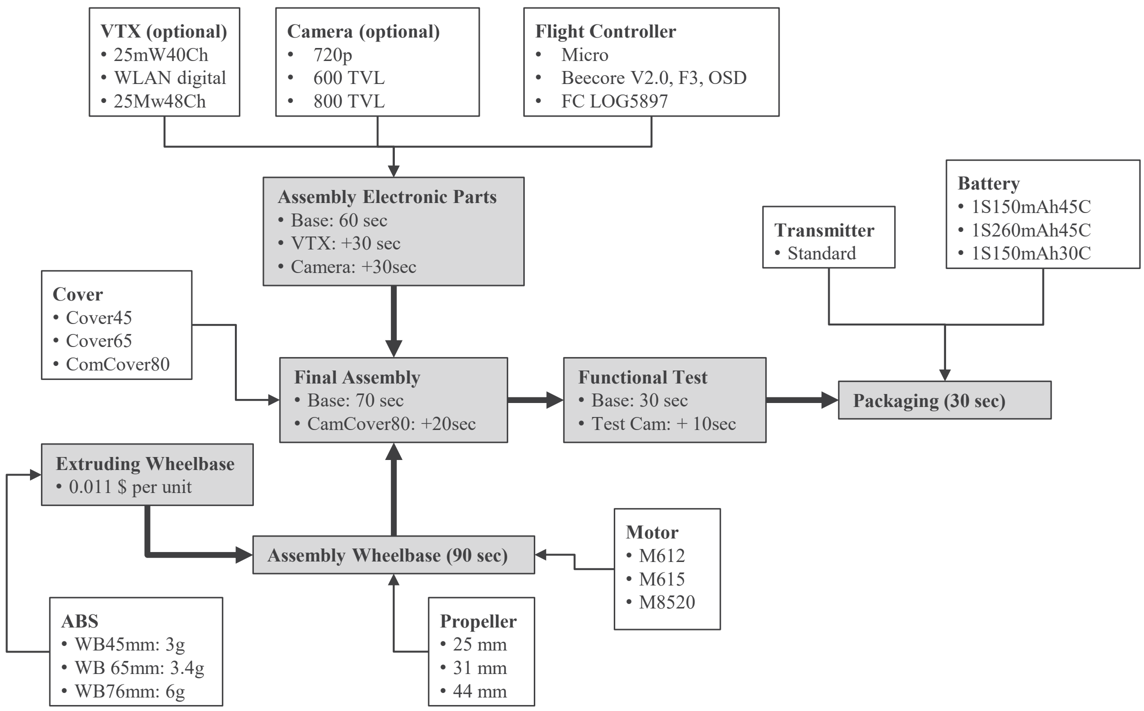
[35] All data of the quadcopter case is included in the numerical EAD model, introduced later in this work.

Table 3.2:
Product matrix with products in rows and functional requirements in columns.

		Flight Time			Size			Video		
Product	Demand	5min	7min	3min	Nano	Micro	Mini	WLAN	FPV low	FPV high
1	541	-	-	1	1	-	-	-	-	-
2	2,483	-	1	-	-	1	-	-	-	-
3	9,874	1	-	-	-	1	-	-	-	-
4	4,943	1	-	-	1	-	-	-	-	-
5	3,210	-	-	1	1	-	-	1	-	-
6	3,530	-	1	-	-	-	1	-	-	1
7	870	-	1	-	-	-	1	-	1	-
8	3,247	-	1	-	-	1	-	-	-	1
9	470	-	1	-	-	1	-	-	1	-
10	7,534	1	-	-	-	1	-	-	-	1
11	713	1	-	-	-	1	-	-	1	-
12	21,574	1	-	-	1	-	-	1	-	-

Figure 3.4 summarizes the manufacturing process, which defines the transition between the component and process domain ($DMM_{PD,PrD}$). Each component is sold directly from a supplier except for the plastic frame (wheelbase). The fictive company works mainly as a system integrator. The three variants of the wheelbase are produced in an extrusion process using one extrusion machine. In a pre-assembly step, motors and propellers are mounted onto the wheelbase. Electronic parts are pre-assembled and later unified with the wheelbase in a final assembly step. The functional test is done, and the duration depends on whether the copter has a camera. Finally, products are packed and shipped.

Figure 3.4:
Production process for quadcopter product family. Processes are grey-shaded, and components are white.

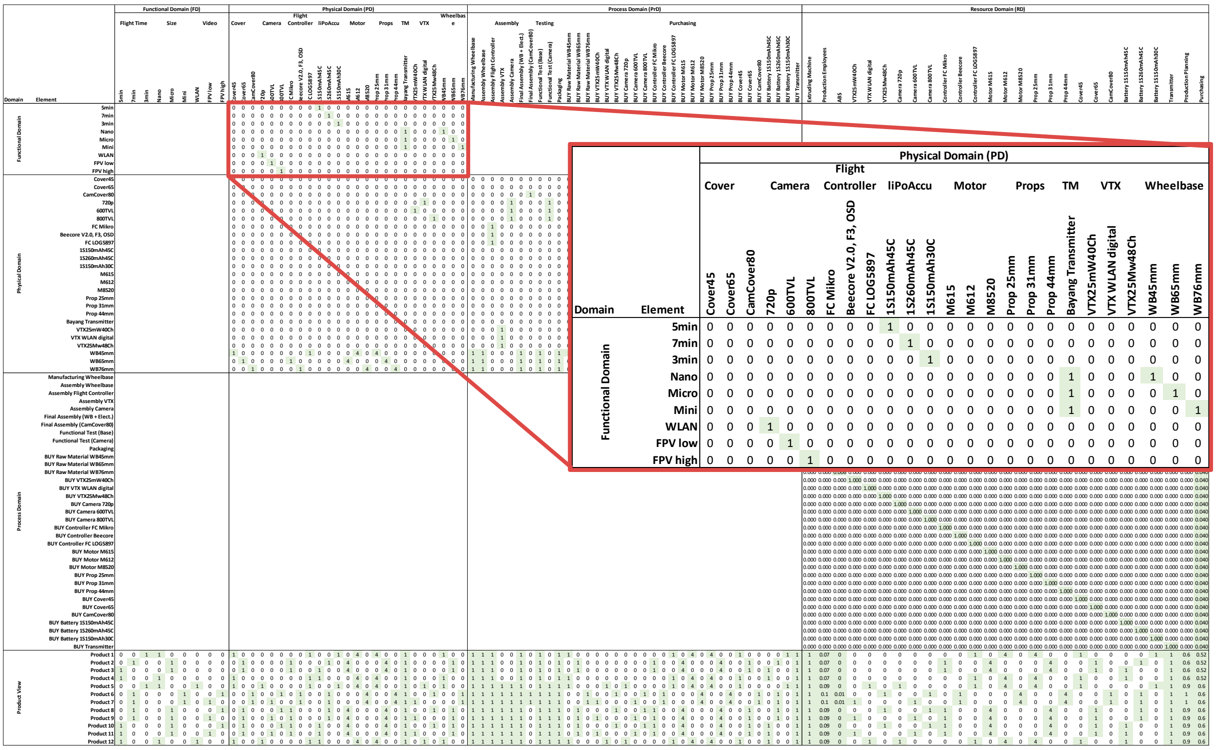


Since all components, except the wheelbase, are externally sourced, they also occur as resources. The wheelbase is made in-house and requires ABS material in different quantities depending on the size. The variable costs for one wheelbase are estimated at 1.1 cents. Each assembly process, functional testing, and packaging require employees with variable labor costs of 30 $ per hour. Table 3.3 reports the variable costs for all resources. It is further assumed that ten percent of resources' total variable costs are fixed except for the extruding machine and the 'Beecore' controller. Figure 3.5 shows the entire EAD for the quadcopter product family.

Table 3.3:
Variable resource unit costs (RCU) and total units for each resource. The fixed costs are assumed to be 10% of the total variable costs for each resource except for the extruding machine and the 'Beecore' controller.

Resource	RCU	Units	Comment	Resource	RCU	Units
Extruding Machine	0.011 $	58,989	fixed costs: 42,857 $/year	Prop 25mm	0.50 $	121,072
Production Employees	30 $/h	5,085		Prop 31mm	0.50 $	97,284
ABS	2 $/kg	200		Prop 44mm	0.50 $	17,600
VTX25mW40Ch	2.20 $	2,053		Cover45	0.50 $	30,268
VTX WLAN digital	1.65 $	24,784		Cover65	0.66 $	24,321
VTX25Mw48Ch	2.50 $	14,311		CamCover80	0.90 $	4,400
Camera 720p	1.00 $	24,784		Battery 1S150mAh45C	0.71 $	44,638
Camera 600TVL	2.20 $	2,053		Battery 1S260mAh45C	1.41 $	10,600
Camera 800TVL	2.50 $	14,311		Battery 1S150mAh30C	0.66 $	3,751
Controller FC Micro	2.65 $	24,321		Transmitter	2.16 $	58,989
Controller Beecore V2.0, F3, OSD	6.00 $	4,400	+7,000 $ license costs			
Controller FC LOG5897	2.65 $	30,268				
Motor M615	2.65 $	97,284				
Motor M612	2.50 $	121,072				
Motor M8520	3.65 $	17,600				

Figure 3.5:
Overview of the quadcopter product family in the EAD. The resource-resource mapping (DSM_{RD}) is excluded from this overview since it holds only zeros and for better readability. The demand and cost view are also not shown as they are reported separately. The full data set is included in the numerical EAD library which is introduced in section 3.5.



3.2.5.2 A Typical Decision-Making Problem

Different design alternatives are generated and evaluated based on the initial product family design regarding their costs and degree of commonality. Firms aim to minimize costs by optimizing product family design, such as increasing component commonality or modularity (Hackl et al., 2020; Labro, 2004). Searching for potential product family design alternatives and selecting the most cost-efficient one represents a typical decision-making problem throughout product development (Chen et al., 2022; Fixson, 2005). In this case study, the decision-making problem is characterized by minimizing costs while maximizing the degree of component commonality. In practice, those problems are called multi-attribute decision-making (MADM) problems as they aim to maximize or minimize confronting objectives (Gonçalves-Coelho & Mourão, 2007; Kahraman & Cebi, 2009; Kulak et al., 2010).

Component commonality describes the reuse of components across different product variants and allows firms to offer a large external variety on the market with a reasonable number of individual components (Fisher & Ittner, 1999; Swaminathan, 2001). It is a two-sided medal and follows the paradigm that the sum of all monetary and non-monetary advantages due to increased economies of scale outweigh the disadvantages, such as increased material costs (Fixson, 2005, 2006). The reviews of Hackl et al. (2020) and Labro (2004) summarize the individual effects of component commonality on costs and beyond, such as flexibility and agility (Cormier, van Horn, & Lewis, 2009; Fuchs & Golenhofen, 2019; Greve et al., 2020; Orton & Weick, 1990) and to reduce risk[36] (Baker et al., 1986; Labro, 2004). Focused on costs, increased component commonality decreases setup and part administration costs or reduces the safety stock and costs for supplier management. Additionally, fixed costs are distributed over a larger volume of products, leading to decreased costs for each unit (Fixson, 2005). A downside of commonality is the increase in material costs due to the overdesign of (Fisher & Ittner, 1999). Perera et al. (1999) note that it has a negative impact on components functionality as tradeoffs must be made during the design process. Therefore, an optimal design is not one with a maximum level of component commonality since, at a certain point, the economies of scale can no longer compensate for the increase in material costs. The objective is therefore defined as:

$$x_{opt} = min\begin{pmatrix} -PCI \\ TC \end{pmatrix} \tag{3.22}$$

The objective aims to maximize component commonality (PCI) while minimizing the total costs (TC). Commonality is measured by using Kota et al.'s (2000) product line commonality index (PCI), which is widely used[37]. Large values of PCI indicate a high degree of component sharing across products, whereas low values refer to product designs with only some components are shared. Since a maximization of PCI is preferable, a negative sign is added for mathematical reasons. The total costs (TC) are calculated according to the EAD principles introduced in the previous section.

---

[36] Some authors note that commonality increases the risk under certain conditions. A reduced number of suppliers puts companies in a position where they depend more on quality and delivery reliability, as Benton and Krajewski (1990) show.

[37] For a more detailed discussion on measures, see sections 3.4 and 3.6.

Product family design alternatives are generated by systematically oversizing four areas of the copter, as shown in Table 3.4. Entries indicate the compatibility for oversizing components. A smaller motor, for example, can be substituted by a larger, more powerful motor. Therefore, one design alternative is to use the *‘M615’* motor for the *‘WB45’* instead of the initial *‘M612’*. The same procedure is applied for the battery, controller, and camera, resulting in 64 design alternatives (S1-S64). These alternatives are modeled by changing entries in the DSM_{PD} as well as $DMM_{FD,PD}$. Each concept is modeled using the numerical EAD model, which reports the corresponding PCI values and the total costs.

Table 3.4:
Compatibility matrix for the motor, battery, flight controller, and camera of the quadcopter product family.

Part	Wheelbase (WB)			RCU	Part	Flight Time (FT)			RCU
	WB45	WB65	WB76			3 min	5 min	7 min	
	Motor (M)					Battery (B)			
M612	x			2.50 $	150mAh30C	x			0.66 $
M615	x	x		2.65 $	150mAh45C	x	x		0.71 $
M8520	x	x	x	3.65 $	260mAh45C	x	x	x	1.41 $
Part	**Wheelbase (WB)**			**RCU**	**Part**	**Video (V)**			**RCU**
	WB45	**WB65**	**WB76**			**WLN**	**FPL**	**FPH**	
	Controller (CO)					Camera (CA)			
FC LOG5897	x			2.65 $	720p	x			1.00 $
FC Mikro	x	x		2.65 $	600TVL		x		2.20 $
Beecore			x	6.00 $	800TVL		x	x	2.50 $

3.2.5.3 Solutions for the Decision-Making Problem

Since a multi-attribute optimization problem is defined, solutions lay in a two-dimensional space (Figure 3.6). While the x-axis represents the degree of commonality for each design alternative (PCI), the y-axis indicates the total costs for each design alternative (TC). Using the concept of Pareto optimality, optimal solutions are identified. A solution is called Pareto optimal if and only if no solution minimizes one objective without increasing one of the others (Arora, 2017). With two objectives, the Pareto optimal points form a curve. This line connects all Pareto optimal points (red) and separates the utopian space containing non-feasible solutions from the one containing inefficient solutions (grey).

Figure 3.6:
Solution space with the commonality index in the x-axis and the total costs on the y-axis.

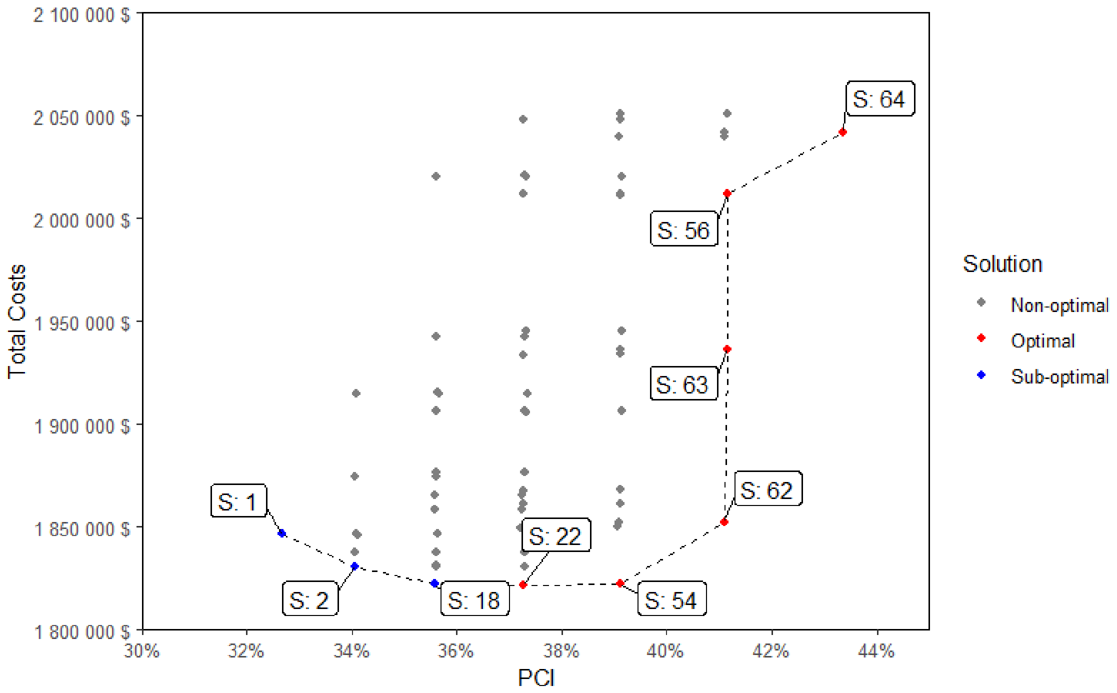


Figure 3.6 shows a U-shaped curve indicating that too little and too much commonality increases costs. It substantiates earlier findings by Fixson (2005) and Thonemann and Brandeau (2000). Six Pareto optimal design alternatives are identified and compared to the initial design (S1). These solutions represent equally good designs from a strictly mathematical perspective compared to S1. Table 3.5 shows that increasing commonality leads to a reduction of fixed costs while increasing the variable costs at the same time. Since the ratio of fixed to total costs varies between $19.6\% \leq R_{fix} \leq 22.8$, overall cost benefits are achieved only for a slight increase in variable costs. In sum, four solutions *(S2, S18, S22, S54*) show marginal lower costs compared to the initial design. For those designs, the degression of fixed costs compensates for the increased variable costs, leading to lower total costs.

Table 3.5:
Pareto optimal solutions

	Costs				Delta to S1			
S	TC [$]	TC_{fix} [$]	TC_{var} [$]	PCI	TC	TC_{fix}	TC_{var}	PCI
1	1,846,501	421,202	1,425,298	32.7%	-	-	-	-
2	1,831,030	387,571	1,443,459	34.1%	-0.8%	-8.0%	1.3%	4.3%
18	1,822,118	378,659	1,443,459	35.6%	-1.3%	-10.1%	1.3%	8.9%
22	1,822,030	378,384	1,443,647	37.3%	-1.3%	-10.2%	1.3%	14.1%
54	1,822,144	377,882	1,444,263	39.1%	-1.3%	-10.3%	1.3%	19.7%
62	1,852,495	374,360	1,478,135	41.1%	0.3%	-11.1%	3.7%	25.8%
63	1,936,605	379,347	1,557,258	41.1%	4.9%	-9.9%	9.3%	26.0%
56	2,011,856	349,237	1,662,619	41.2%	9.0%	-17.1%	16.7%	26.0%
64	2,042,207	345,716	1,696,491	43.3%	10.6%	-17.9%	19.0%	32.7%

Table 3.6 represents the design solution S54 in more detail. This solution indicates a potential trade-off between both objectives. In this design alternative, the motor 'M612' is replaced by 'M615', the battery '150mAh30C' by '150mAH45C', the flight controller 'FC LOG5897' by 'FC

Mikro' and the camera '600 TVL' by '800 TL'. These decisions reduce the total number of parts by four components. In practice, however, cost differences between scenarios *S18, S22, and S54* are neglectable, especially during the early stages where uncertainty is still high. More relevant is the abrupt cost increase as indicated by the solutions S62, S63, S56 and S64. It indicates a warning sign for engineers that higher levels of commonality are associated with significantly higher costs and, therefore, should be avoided.

Table 3.6:
Design solution 'S54' indicates an optimal trade-off between commonality and costs.

Part	Wheelbase (WB)			RCU	Part	Flight Time (FT)			RCU
	WB45	WB65	WB76			3 min	5 min	7 min	
	Motor (M)					Battery (B)			
M612				$ 2.50	150mAh30C				$ 0.66
M615	x	x		$ 2.65	150mAh45C	x	x		$ 0.71
M8520			x	$ 3.65	260mAh45C			x	$ 1.41
Part	**Wheelbase (WB)**			**RCU**	**Part**	**Video (V)**			**RCU**
	WB45	**WB65**	**WB76**			**WLN**	**FPL**	**FPH**	
	Controller (CO)					Camera (CA)			
FC LOG5897				$ 2.65	720p	x			$ 1.00
FC Mikro	x	x		$ 2.65	600TVL				$ 2.20
Beecore			x	$ 6.00	800TVL		x	x	$ 2.50

The case study highlights the two-sided face of component commonality where engineers face the trade-off between increasing economies of scale by keeping the total costs low. In doing so, 64 design alternatives are systematically created and evaluated regarding their level of component commonality and total costs. The case study shows the typical U-shaped curve, meaning that too little and too much commonality is not beneficial for firms. It is shown that costs begin to increase rapidly at a certain level of commonality. A detailed analysis of such design solutions (S62, S63, S56, S64) reveals that these designs are characterized by overdesigning cheaper, high-volume components with more expensive, low-running components. From a methodological perspective, the case study demonstrates EAD's ability to generate product family design alternatives and evaluate their economic consequences. While this case study is limited to a single product family, later parts (section 4) investigate the relationship between product designs and economic consequences in more detail. In doing so, large-scale numerical experiments are conducted in which different product family designs are systematically generated. However, two open issues must be solved a priori to conduct these experiments. First, the EAD is still a conceptual model and needs to be transferred into a computational model, according to section 2.3. Second, measures to compare generated designs are needed. Therefore, the next steps of this work are as follows: Section 3.5 introduces the computational EAD simulation model, while section 3.3 - 3.4 introduces the PFCMs.

3.3 Prior Considerations on Complexity Measures

The EAD model represents product family designs in a structured way and supports multi-attribute decision-making. However, the model does not allow for comparing product family designs in terms of their complexity as a systemic property. Therefore, this section discusses two prior considerations on the measurement complexity within the EAD before section 3.4 introduces product family complexity measures (PFCMs). While the first prior consideration discusses the integration of complexity in the EAD on a more conceptual level, the second addresses the issues on PFCM's validity raised by different authors (Blecker & Abdelkafi, 2006; Hennig et al., 2022; Sinha & Suh, 2018).

3.3.1 Complexity within the EAD Framework

Since the EAD is a single-period and, thus, a time-invariant model, only static aspects of complexity can be observed. According to the conceptual section on complexity in a systemic context, literature defines systemic complexity by multiplicity, interrelatedness, and diversity. These dimensions occur at different places within the EAD. Multiplicity refers to the number of system entities, such as the number of functional requirements (N_{FD}), components (N_{PD}), processes (N_{PrD}), and resources (N_{RD}), which represent the most granular (atomic) layer. Product family designs with more domain elements are more complex than those with only some elements. The dimensions of the inter- and intra-domain matrices (DSM, DMM) define the number of design elements. Multiplicity further occurs on higher hierarchical layers within the EAD, such as in the product view. The number of distinct products (N_{PROD}) is represented by the rows of the product matrices (P). While the minimum number of products is $N_{PROD} = 1$ since each domain requires at least one design element, the maximum number of products is restricted by the number of functional requirements. It is impossible to generate more distinct products than the power set of functional requirements minus one. This leads to a lower and upper boundary for the number of products derived from a design defined as[38]:

$$1 \leq N_{PROD} \leq 2^{N_{FD}} - 1 \tag{3.23}$$

Assuming the same number of design elements, product families with a small product mix are, therefore, less complex than those with a large product mix.

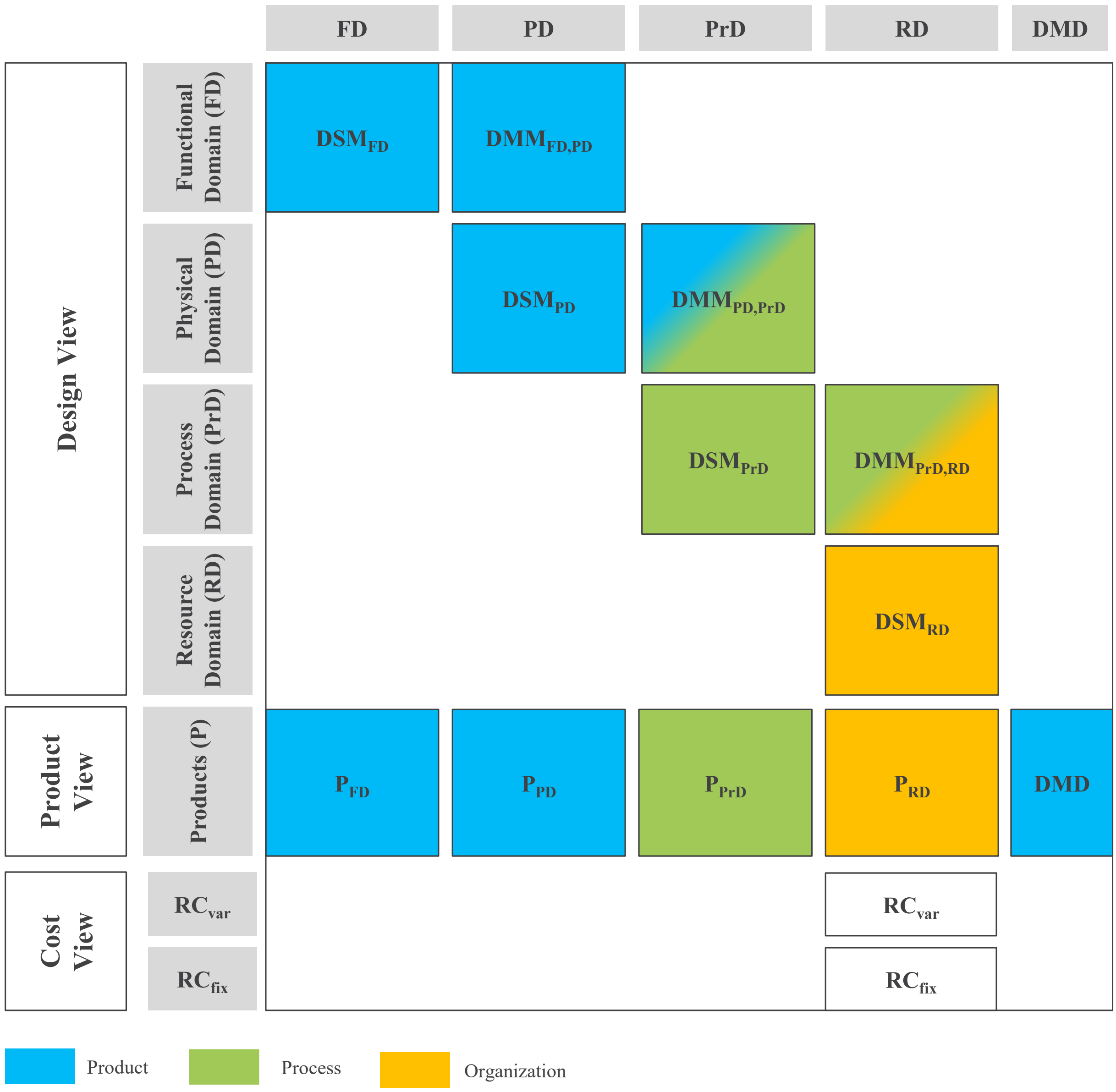
While multiplicity exists in both the design and product view, interrelatedness is limited to the design view since the EAD assumes no interactions among products. Within the EAD, there exist two types of relations. Inter-domain relations are represented by the domain mapping matrices (DMM) and intra-domain relations in dependency structure matrices (DSM). Product family designs with fewer entries in the DMMs and DSMs are, therefore, less complex than those with many entries. Diversity, the third dimension of static complexity, describes the differences between elements based on specific criteria (Buchholz, 2012). This definition implies that the diversity cannot be measured on the most granular (atomic) layer as these elements cannot be characterized in more detail. This argumentation implies that diversity is only present in EAD's product view. However, if future research adds more granular layers to the EAD, the design element's diversity measurement becomes possible. An example would be to describe components (physical domain elements) by their shape, weight, or material. Products are defined as a set of domain elements. Thus, product mixes in which products share more elements among each other are less complex as they are less diverse than those with less element sharing among products. A second way to characterize product diversity is product demand. Homogenous product demand vectors are less complex (less diverse) than those characterized by significant demand variations across products. Linking the concept of

[38] It is noted that this assumption is only valid when P_{FD} is a binary matrix.

complexity with the EAD is crucial for the later discussion of PFCMs since design view measures cannot represent diversity while products cannot reflect interrelatedness.

While the previous section provides a generic overview of how complexity manifests in the EAD, it still needs to be solved how certain parts of the model represent internal complexity, as introduced in section 2.5.4. According to Wilson and Perumal's (2010) complexity cube, internal complexity is separated into product, process, and organizational complexity. These types are represented by different domains within the EAD, where Figure 3.7 provides an overview. Product complexity contains the product portfolio complexity and the structural complexity in terms of product architecture and components interrelatedness (e.g., Jung et al., 2022; Sinha & Suh, 2018). Product portfolio complexity is represented by the product mix P_{FD} as the constrained combination of functional requirements according to DSM_{FD} and the demand vector DMD. Structural complexity is represented by the product architecture ($DMM_{FD,PD}$) and the intra-component dependency matrix (DSM_{PD}). Several studies use DSM_{PD} matrices to analyze intra-domain product complexity (G. Kim et al., 2016; Sinha & Suh, 2018) and inter-domain product complexity using $DMM_{FD,PD}$ (Jung et al., 2022) or both (Sawai et al., 2017).

Figure 3.7:
The manifestation of internal complexity in the EAD framework.



Process Complexity is manifested in $DMM_{PD,PrD}$ and the process dependency matrix DSM_{PrD}. $DMM_{PD,PrD}$ defines the interactions between the product and process domain, representing a mix of product and process complexity. As a result of the process design, P_{PrD} represents process complexity as it as it reports process differences across products. Karniel and Reich (2009), for example, use DSM_{PrD} to model the dependencies among processes. The same logic is used for the resource domain, representing organizational complexity. Bonjour and Micaëlli (2010), for example, use DSM_{PrD} and DSM_{RD} as well as $DMM_{PrD,RD}$ matrices to analyze process and organizational complexity. Their DSM_{RD} matrices, for example, represent the information flow between different roles within the organization. The mapping between processes and resources further represents the process-organizational face, as noted in Wilson and Perumal's (2010) complexity cube. Complexity caused by product-organization interactions is represented by matrices above the main DMM diagonal ($DMM_{FD,PrD}$, $DMM_{FD,RD}$ and $DMM_{PD,RD}$). However, these matrices are neglected for reasons of simplicity.

3.3.2 The Validity of Complexity Measures

A second prior consideration is related to the validation of measures. With more than 1500 citations, the framework introduced by O'Leary-Kelly and Vokurka (1998) is used as a foundation to discuss the validity of PFCMs. They note three steps that are necessary for the validation of measures. First, content validity ensures that a measure is logically related to a variable on a conceptual level, a so-called construct. Construct validity is checked in a second step and describes the extent to which a measure represents the construct. The nomological validity checks how the measure relates to other constructs (O'Leary-Kelly & Vokurka, 1998)[39]. Related to this section's underlying problem, a PFCM has content validity if it depends on at least one of the complexities' static dimensions (multiplicity, interrelatedness, diversity). This ensures that a measure can reflect the nature of complexity on, at least, a conceptual level. For example, a measure that counts the number of components depends on the multiplicity of system elements, but it cannot reflect the interrelatedness and diversity. Nomological validity is checked in the context of a specific research objective as it requires a second construct. Section 4, for example, investigates the nomological validity of PFCMs as a proxy for total costs, while section 5 checks the nomological validity regarding their ability to measure the accuracy of product costing systems.

Still undefined is the construct validity, which is proven using the widely accepted[40] studies by Weyuker (1988) and Briand et al. (1996). Their works provide criteria to prove construct validity in the context of complexity measures. More precisely, these studies formulate properties for measures that reflect systems' multiplicity or interrelatedness. Both approaches are known as property-based approaches as they *"formalize the empirical properties that […] complexity […] must satisfy in order for it to be used in the analysis of any measurement proposed for that attribute"* (Piattini, Genero, & Jiménez, 2001, p. 709)[41]. For example, Sinha and Suh (2018) use these criteria to verify the construct validity of their measures. While these works provide a solid foundation to check measures' construct validity for the dimension of multiplicity and interrelatedness, no criteria for diversity exist in the literature. Table 3.7 provides an overview of these criteria and their relevance for the dimension interrelatedness or multiplicity.

---

[39] Some studies that discuss the validity of PFCMs classify nomological validity incorrectly as face validity, for example, Sinha and Suh (2018). In the following, nomological validity is used rather than face validity.

[40] Each study has more than 1000 citations across different disciplines.

[41] Piattini et al. (2001, p. 709) differ between property-based and measurement theory-based approaches. The latter group *"check for a specific measure if the empirical relations between the elements of the real world established by the attribute being measured, are respected when measuring the attributes ".*

Table 3.7:
Property-based criteria to prove construct validity for a complexity measures 'C' as introduced by Weyuker (1988) and further extended by Briand et al. (1996) applied to different systems P, Q, R. Each system consists of elements (P_{el}, Q_{el}, R_{el}) and relations (P_r, Q_r, R_r).

#	Criteria	Description	Rel[1)]	Property
C1	nonnegativity	The complexity for a system is non-negative since nothing can be simpler than the reference simplicity. $\forall P; C(P) \geq 0$	IM	Size 1 Complexity 1
C2	null value	The size of a system is zero if the system has no elements. $C(P) = 0, if\ P_{el} = \emptyset$	M	Size 2
C3	additivity	The size of a system (P) is the sum of the size of its sub-systems Q and R $Q \subseteq P\ and\ R \subseteq P\ and\ P_{el} = Q_{el} \cup R_{el}\ and\ Q_{el} \cap R_{el} = \emptyset$ $C(P) = C(Q) + C(R),$	M	Size 3
C4	null value	The complexity of a system is zero if the system has no relations ($P_r = \emptyset$). $C(P) = 0$	I	Complexity 2
C5	monotony	The complexity of two sub-systems is always less or equal than the complexity of the combined systems with no shared relations ($R \cup Q \subseteq P\ and\ Q_r \cap R_r = \emptyset$) $C(P) + C(Q) \leq C(P; Q)$	I	Complexity 3
C6	disjoint additivity	Complexity for a system consisting of non-interacting (disjoint) sub-systems ($P_{el} \in Q_{el} \cup R_{el}\ and\ Q_{el} \cap R_{el} = \emptyset\ and\ P_r \in Q_r \cup R_r\ and\ Q_r \cap R_r = \emptyset$) is equal the sum of subsystems' complexity $C(P) = C(Q) + C(R)$	I	Complexity 4

Note. 1) Rel indicates the relevance of complexities' dimension with M=multiplicity and I=interrelatedness where multiplicity is seen as a proxy for system size and interrelatedness as a proxy for what Briand et al. note as complexity.

While Weyuker (1988) introduces nine properties, the work of Briand et al. (1996) consolidates these criteria and proposes three criteria for systems size (Size 1- 3) and four for systems complexity (Complexity 1 – 4). Both Weyuker (1988) and Briand et al. (1996) define these properties in the context of software system design, where they have been primarily used in recent years. However, as their theories define mathematical properties, an application in other contexts is possible and done by several others in the field of PFCMs (Hennig et al., 2022; Jung et al., 2022; Sinha, 2014; Sinha & Weck, 2012, 2013b).

Measures' construct validity regarding multiplicity is proven by three criteria (C1-C3). Non-negativity (C1) states that the complexity of any system is positive. A system (P) for which $C(P) = 0$ represents the reference simplicity. In this case, $C(P)$ represents a generic function to calculate the complexity measure. A measure must fulfill this condition since nothing can be simpler than the reference simplicity. This condition also applies to interrelatedness measures. The null value condition (C2) states that a multiplicity measure must be zero for a system without elements. Additivity is the last condition that needs to be fulfilled by those measures. It states that the size of two systems without shared elements equals the sum of the individual elements. In addition to C1, interrelatedness measures must fulfill four additional conditions (C4-C6). As for multiplicity, a null value condition (C4) exists also for interrelatedness. It states that interrelatedness complexity must be zero if the system has no relationships. Monotony (C5) states that the complexity of two sub-systems without any shared relations is always less or equal to the complexity of the combined system. While condition C5 allows element sharing among sub-systems, condition C6 is much stricter. The disjoint module additivity states that the complexity of two sub-systems equals the combined systems' total complexity if sub-systems neither share elements nor relations. During the selection process of PFCMs, these criteria are used to check the construct validity of the selected measures.

3.4 Measures of Product Family Complexity

This section introduces PFCMs to describe the complexity of individual EAD realizations. Separated into three parts, the first identifies measures reported in the literature and introduces a framework for classification. These measures are then introduced in the following two sections, where 3.4.2 introduces system-level and 3.4.3 on product-level measures. While system-level measures return exactly one value for an entire EAD matrix, such as the product matrix P or the design matrices (DSMs, DMMs), product-level measures return values for each product in P. Therefore, product level measures exist only in EAD's product view.

3.4.1 Search for Complexity Measures and Classification

Five seminal works that reported complexity measures over the recent years are used to identify relevant complexity measures (see Table 3.8). This follows a seminal-work-driven approach, as suggested by Hiebl (2021). These studies were selected as they were published in highly rated journals (journal's h-index) or received attention over the recent years (number of citations). In the next step, complexity measures stated in these studies are extracted and structured according to the classification framework introduced in the next section.

Table 3.8:
Overview on seminal studies used for the identification of complexity measures.

Study	Journal	Journal h-Index1)	Citations2)
Summers and Shah (2010)	Journal of Mechanical Design	126	202
Sinha and Suh (2018)	Research in Engineering Design	71	51
Trattner et al. (2019)	CIRP Journal of Manufacturing Science and Technology	53	56
Jung et al. (2022)	IEEE Transactions on Engineering Management	97	6
Hennig et al. (2022)	Journal of Mechanical Design	126	14

Note. 1) Source: SCImago, (n.d.). SJR — SCImago Journal & Country Rank [Portal]. Retrieved 09.01.2023, from http://www.scimagojr.com; 2) Number of citations according google scholar search conducted in January 2023

Measures are classified based on six criteria: application, theoretical concept, field, level, EAD matrix, and dimension. The application describes whether a measure can be integrated within the current EAD model. A measure is declared unsuitable for application when it requires input not contained in the EAD model, such as a hierarchy of functional requirements or an expert's judgment for firm-specific constants. While the first condition is a limitation of the current EAD, the expert's judgment refers to subjective complexity and, therefore, is not considered in this work. *'Concept'* describes the mathematical roots behind each measure. Count-based measures count the number of elements. Matrix-based measures use matrix properties, whereas graph-based measures are a sub-group of matrix-based measures since each graph can be converted into an adjacency matrix; however, not every matrix is a graph. Similarity-based measures are characterized by pairwise comparison of element differences using an underlying metric. Entropy-based measures refer to the information-theory perspective of entropy (Shannon, 1948). Such measures describe the amount of information necessary to characterize the state of a system and, therefore, are a proxy for the informational content of an individual message[42].

'Field' describes the community using this measure. They are identified using the journal in which these measures were introduced or recently used. Table 3.9 provides an overview of how these measures are distributed across the different fields. MISC describes measures that are stated across various fields. Engineering contains mechanical engineering as well as software engineering. The following two columns of Table 3.9 show the distribution of measures across the measurement level. During the review, 26 system- and four product-level measures were identified. While each product has a corresponding product-level measure, system-level measures return exactly one value for a matrix. Therefore, product-level measures exist only in EAD's product view, whereas system-level measures exist in all EAD matrices. Interestingly, engineering focuses exclusively on the system-level view, while cost accounting focuses on the system and product view. The following three columns of Table 3.9 show the distribution of measures across the conceptual linkage to complexities' dimensions. Engineering and operations management are the only fields that include measures across all dimensions. Accounting, however, treats complexity only by the dimension of diversity. An interesting fact is that eight measures aim to measure the interrelatedness but also depend on the multiplicity of elements. The column *'Matrix'* refers to the classification criteria and indicates which matrices are required for calculation. It is observed that accounting uses product matrices primarily, while engineering uses all kinds of matrices.

Table 3.9:
Distribution of measures across field, measurement level and complexities' dimension.

Field	Level		Dimension[1)]			Matrix[2)]				
	System	Product	M	I	D	DMM	DSM	P	DMD	RC
MISC	2	0	2	0	0	1	0	1	0	0
Engineering	16	0	11	11	2	5	8	3	0	0
Data Science	1	1	0	1	1	0	1	1	0	0
Physics	1	0	1	1	0	0	1	0	0	0
Accounting	3	3	0	0	6	0	0	5	0	1
Operation Management	3	0	1	1	3	0	0	2	2	0
Total	**26**	**4**	**15**	**14**	**12**	**6**	**10**	**12**	**2**	**1**

Note. 1) M=multiplicity, I=interrelatedness, D=diversity. A measure can be assigned to multiple dimensions and matrices at once; 2) DMM=domain mapping matrix, DSM=dependency structure matrix, P=product matrix, DMD=demand vector, RC=resource cost vector.

Table 3.10 provides a detailed overview of the 30 PFCMs. Ten measures are excluded (column *'USE'* empty) since they are not applicable or only under certain conditions. These measures are reangularity (R) and semangularity (S), introduced by Suh (1995), as they exist only for squared DMM matrices. Four measures require expert judgment such as Bashir and Thomson's (2001) product complexity (PC), Ameri et al.'s (2008) coupling complexity (CC), Macduffie, Sethuraman, and Fisher's (1996) model mix complexity (MMC) as well as Martin and Ishii's (2002) coupling index ($COPI$). Two measures require a hierarchical structure for the domains, such as the size complexity (C_{size}) and the descriptive complexity (BDC). Product

[42] Passerini and Severini (2009, p. 59) state that the Shannon entropy *„measures the amount of uncertainty of a random variable, or the amount of information obtained when its value is revealed"*.

mix heterogeneity (PMH) is dropped from further analysis as it refers to a dynamic system view where the EAD is a static model (S. W. Anderson, 1995). Jacobs's (2013) general complexity index (GCI) is dropped since its calculation depends on the element order. In sum, the number of measures is reduced from 30 to 20 (17 system-level, three product-level) after this first screening. The remaining measures spread across all fields and concepts. The last column (‘*Range*’) indicates the variable’s range, which, however, is not always the original range. Some variables are normalized within the zero-one interval during the introduction as it allows both a reporting of absolute complexity values (unnormalized) and relative values (normalized). Such measures are indicated with an *‘x’* in the column *‘NORM’*. Standardized measures allow for comparing the complexity of different-sized product families. The following two sections introduce the remaining 20 measures separated into system level (3.4.2) and product level measures (3.4.3).

Table 3.10:
Identified product family complexity measures

#	Name	Authors	USE	Concept	SYM[1]	Field	Level	Matrix	View[2]	DIM[3]	Norm[4]	Range[5]
M1	Total system size	Various authors, see review by Trattner et al. (2019)	x	Count	TSS	MISC	System	DMM	D	M	-	0-Inf
M2	Reangularity	Suh (1995);Delaš, Škec, and Štorga (2018); Tarcan & Kar (2010); Borjesson and Hölttä-Otto (2014)		Matrix Property	R	Engineering	System	DMM	D	I	A	0-1
M3	Semangularity	Suh (1995); Delaš et al. (2018); Tarcan & Kar (2010)		Matrix Property	S	Engineering	System	DMM	D	I	A	0-1
M4	Product Complexity	Bashir & Thomson (2001)		Graph Property	PC	Engineering	System	DSM	D	M	-	0-Inf
M5	System Design Complexity	Suh (1995); Guenov (2002); Modrak & Bednar (2015)	x	Entropy	SDC	Engineering	System	DMM	D	M, I	x	0-1
M6	Modularity	Newman (2006); Newman (2008); Blondel et al. (2008)	x	Graph Property	Q	Data Science	System	DSM	D	I	A	0 - 1
M7	Size Complexity	Ameri et al. (2008)		Entropy	C_{size}	Engineering	System	DSM	D	M	-	
M8	Coupling Complexity	Ameri et al. (2008); Summers & Shah (2010)		Graph Property	CC	Engineering	System	DSM	D	I	-	
M9	Structural Complexity	Kim et al. (2016); Sinha and Weck (2013b); Sinha (2014); Sinha and Weck (2016); Liu, Wang, & van Mieghem (2010); Sinha & Suh (2018); Sarkar et al. (2014); Hennig et al. (2022)	x	Matrix Property	SC	Engineering	System	DSM	D	M, I	-	0-Inf
M10	Neumann Entropy	Passerini & Severini (2009); Anand et al. (2011)	x	Entropy	NE	Physics	System	DSM	D	M, I	x	0-Inf
M11	Product Mix Heterogeneity	Anderson (1995)		Similarity	PMH	Cost Accounting	Product	P	P	D	A	0-1
M12	Proportion of Product Variants	Various authors, see review by Trattner et al. (2019)	x	Count	NPV	MISC	System	P	P	M	x	0-Inf
M13	Diversification Index	Gollop (1997); Gollop & Monahan (1991); Brahm, Tarzijan, & Singer (2017)	x	Similarity	D	Operations Management	System	P, DMD	P	D	A	0-1
M14	Product Line Commonality Index	Kota et al. (2000)	x	Matrix Property	PCI	Engineering	System	P	P	D	A	0-1
M15	Inter-product heterogeneity	Gupta (1993); Mertens (2020)	x	Similarity	INTER	Cost Accounting	Product	P	P	D	-	0-Inf
M16	Intra-product heterogeneity	Gupta (1993); Mertens (2020)	x	Similarity	INTRA	Cost Accounting	Product	P	P	D	-	0-Inf

Note. 1) SYM=symbolic notation; 2) EAD's view with P=product view, D= design view and C=cost view; 3) Complexities' dimensions defined as: M=multiplicity, I=interrelatedness, D=diversity; 4) Norm indicates whether a measure is already normalized (A); is normalized in this work (x) or cannot be normalized (-). Normalization means that a measure is independent from the system size measured via multiplicity. 5) variables range for the unnormalized Measure. Normalized measures lay within a 0-1 bound.

Table 3.10 (continued):
Identified product family complexity measures

#	Name	Authors	USE	Concept	SYM[1)]	Field	Level	Matrix	View[2)]	DIM[3)]	Norm[4)]	Range[5)]
M17	Local Outlier Factor	Breunig (2000); Xu et al. (2022)	x	Similarity	$LOF_{5\%}$	Data Science	Product	P	P	D	-	0-Inf
M18	Degree of Resource Sharing	Balakrishnan et al. (2011); Anand et al. (2019)	x	Matrix Property	DNS	Cost Accounting	System	P	P	D	A	0-1
M19	Demand Skewness	Jensen et al. (1996); Ruiz-Torres and Mahmoodi (2008)	x	Matrix Property	$DMD_{T10\%}$	Operations Management	System	DMD	P	D	A	0.1-1
M20	Cost Skewness	Balakrishnan et al. (2011); Anand et al. (2019)	x	Matrix Property	RC_{T10}	Cost Accounting	System	RC	C	D	A	0.1-1
M21	Model-Mix Complexity	MacDuffie et al. (1996)		Matrix Property	MMC	Engineering	System	P	P	M	-	0-Inf
M22	Option Variability	MacDuffie et al. (1996); Fisher & Ittner (1999)	x	Matrix Property	OV	Management Accounting	System	P	P	D	x	0-Inf
M23	Commonality Index	Martin & Ishii (1996)	x	Matrix Property	CI	Engineering	System	P	P	M, D	x	0-1
M24	Coupling Index	Martin & Ishii (2002)		MISC	COPI	Engineering	System	DMM	D	I	A	0-1
M25	Generalized Complexity Index	Jacobs (2013)		MISC	GCI	Operations Management	System	P	P	M, I, D	A	0-1
M26	Cyclomatic Complexity	McCabe (1976)	x	Graph Property	MCC	Engineering	System	DSM	D	M, I	x	0-Inf
M27	Descriptive Complexity	Broniatowski & Moses (2016)		Entropy	BDC	Engineering	System	DSM	D	M, I	-	0-Inf
M28	Halstead's volume measure	Halstead (1977); Prather (1984)	x	Entropy	HVM	Engineering	System	DSM	D	M, I	x	0-Inf
M29	Interface Complexity	Hölttä & Otto (2005)	x	Count	HIC	Engineering	System	DSM	D	M, I	x	0-Inf
M30	Jung's System Design Complexity	Jung et al. (2022)	x	Matrix Property	JSDC	Engineering	System	DMM	D	M, I	-	0-Inf

Note. 1) SYM=symbolic notation; 2) EAD's view with P=product view, D= design view and C=cost view; 3) Complexities' dimensions defined as: M=multiplicity, I=interrelatedness, D=diversity; 4) Norm indicates whether a measure is already normalized (A); is normalized in this work (x) or cannot be normalized (-). Normalization means that a measure is independent from the system size measured via multiplicity. 5) variables range for the unnormalized Measure. Normalized measures lay within a 0-1 bound.

3.4.2 System Level Measures

System-level measures describe the complexity on a system level using information from entire EAD matrices, whereas product-level measures use only parts of an EAD matrix. In the following, 17 system-level measures are introduced, separated into DMM, DSM, product matrix, and other measures such as cost and demand.

3.4.2.1 Domain Mapping Matrix

The number of system-constructing elements, such as the number of components, processes, or options, is used across several studies (Trattner et al., 2019) and refers to the dimension of multiplicity. The columns of the DMMs define system constructing elements with N_{FD} being the number of functional requirements, N_{PD} the number of components, N_{PrD} processes, and N_{RD} resources. Table 3.11 shows ten domain mapping matrices (DMM) with the corresponding values for each DMM measure introduced in this section.

Table 3.11:
Nine matrices representing different possible design matrices with their corresponding complexity measures.

#	Design	Matrix	N	SDC	SDC_N	JSDC
1	uncoupled (3x3)	$\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$	3	0	0	3
2	decoupled (3x3)	$\begin{pmatrix}1&0&0\\1&1&0\\0&0&1\end{pmatrix}$	3	1.39	0.14	4.31
3	decoupled (3x3)	$\begin{pmatrix}1&0&0\\1&1&0\\1&1&1\end{pmatrix}$	3	4.68	0.47	7.21
4	coupled (3x3)	$\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\end{pmatrix}$	3	6.07	0.61	8.93
5	uncoupled-redundant[1)] (3x4)	$\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1\end{pmatrix}$	4	0	0	4.55
6	decoupled (3x4)	$\begin{pmatrix}1&0&0&0\\1&1&0&0\\1&1&1&1\end{pmatrix}$	4	4.68	0.36	9.27
7	coupled-redundant (3x4)	$\begin{pmatrix}1&1&0&1\\0&0&1&1\\1&1&1&1\end{pmatrix}$	4	7.45	0.57	13.03
8	coupled-redundant (4x3)	$\begin{pmatrix}1&0&1\\0&1&1\\1&1&1\\0&1&0\end{pmatrix}$	3	7.98	0.48	11.43
9	coupled-redundant (4x4)	$\begin{pmatrix}1&1&0&0\\1&1&0&0\\0&0&1&1\\0&0&1&1\end{pmatrix}$	4	5.55	0.25	8
10	coupled (4x4)	$\begin{pmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\1&1&1&1\end{pmatrix}$	4	22.18	1	16

Note. 1) matrix represents a redundant design as design parameter three and four could be merged.

While N counts the number of elements, it provides no information on the distribution of matrices' entries. The number and the positions of non-zero entries define whether a design is uncoupled, decoupled, or coupled. A measure that reflects these differences is the System Design Complexity (SDC). The measure is based on the information-theoretic definition of entropy, so-called Shannon entropy (Shannon, 1948). The Shannon entropy describes the amount of information that is necessary to characterize the state of a system and is the foundation for ADs' information axiom. A DMM, therefore, transports information from one domain into another. In uncoupled designs, each target domain element depends on only one source element, and the number of messages is equal to the dimensions of a DMM. With an increasing degree of coupling, more messages are needed to transport information between the domains. Higher entropy values indicate a more coupled design, whereas small values refer to loosely coupled or uncoupled designs. As a measure of coupling complexity, SDC represents the aspect of interrelatedness. Several authors apply this concept in different contexts, such as product design (Guenov, 2002; Modrak & Bednar, 2015), product development projects (Schlick, Beutner, Duckwitz, & Licht, 2007) and beyond (Ladyman, Lambert, & Wiesner, 2013). According to Modrak and Bednar (2015), SDC measures the amount of information necessary to describe the DMM coupling and is defined as:

$$SDC(DMM_{src,tgt}) = \sum_{j=1}^{N_{tgt}} N_j * \ln(N_j) \quad (3.24)$$

where N_j is the number of interactions with source elements for each target domain element $j = 1 \dots N_{tgt}$ (column of DMM). Since the authors focus on product design complexity, this measure is only defined for $DMM_{FD,PD}$ explicitly. However, the concept can be extended to other domain transitions, leading to the physical design complexity $SDC_{FD,PD} = SDC(DMM_{FD,PD})$, process design complexity $SDC_{PD,Pr} = SDC(DMM_{PD,PrD})$ and organizational design complexity $SDC_{PrD,RD} = SDC(DMM_{PrD,RD})$. The corresponding values for the design matrices are presented in Table 3.11, where larger values indicate a more coupled design. If $SDC = 0$ then the design is uncoupled. The measure reaches its maximum if the design matrix is fully coupled (DMM_{max}). This allows normalization and eliminates the impact of system size on this measure. The normalized system design complexity (SDC_N) is written as a proportion of the maximum possible design complexity compared to the actual design complexity and is defined as:

$$SDC_N(DMM) = \frac{SDC(DMM)}{SDC(DMM_{max})} \quad (3.25)$$

An advantage of this measure is that it indicates whether a design is uncoupled or not since the DMM must not contain any zero rows $SDC = SDC_N = 0$ if and only if the design is either uncoupled or redundant-uncoupled.

Another measure that captures the DMM coupling is Jung et al.'s (2022) system design complexity ($JSDC$). While SDC is based on the concept of entropy, $JSDC$ uses matrix energy as an underlying concept. The matrix energy $E(DMM)$ is defined as the sum of singular values for a given matrix. It describes the ability to reconstruct a system with a minimum amount of information (Sinha, 2014) and *“encapsulates the ‘intricateness’ of the dependencies among the requirements and system elements and expresses the underlying topological complexity”* (Jung et al., 2022, p. 2190). The measure is defined as:

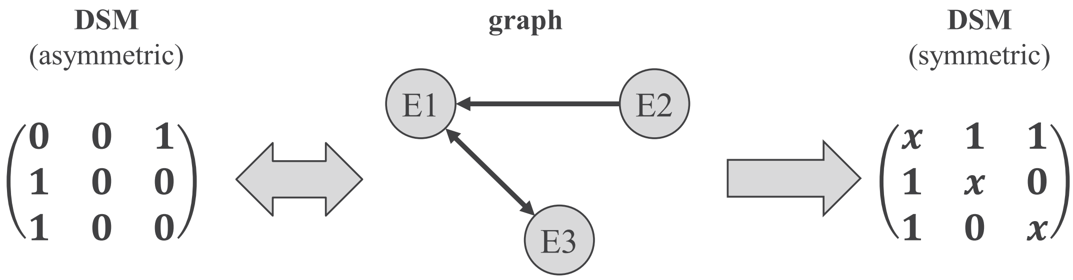
$$JSDC(DMM_{src,tgt}) = \sum_{j=1}^{N_{tgt}} DMM_{src,j} * \frac{E(DMM_{src,tgt})}{\min(N_{src}, N_{tgt})} \quad (3.26)$$

with N_{src} being the number of rows, and N_{tgt} the number of columns in $DMM_{src,tgt}$. Low values of this measure indicate a more un- or decoupled design, whereas large values refer to a more coupled design. As of today, an approach for normalization of $JSDC$ has yet to be found. Analytical upper bounds for the matrix energy exist only for adjacency matrices (e.g., Koolen & Moulton, 2001; McClelland, 1971), where the DMM represents a bipartite graph.

3.4.2.2 Dependency Structure Matrix

Dependency structure matrices contain information on the system's intra-domain couplings. DSMs are always square matrices as they contain the same elements in columns and rows, which allows for treating them as adjacency matrices of a graph[43]. Figure 3.8 shows the transition between the DSM and a graph. An asymmetric matrix with source elements in rows and targets in columns can be transformed into a directed graph and vice versa. A directed graph, however, can be transformed into a symmetric matrix but only back into an undirected graph. A distinction between symmetric (undirected) and asymmetric (directed) matrices is essential. While the asymmetry is crucial for calculating the product view in the EAD, it is less critical for measuring element coupling. Many studies use symmetric DSMs to describe interfaces between elements (AlGeddawy & ElMaraghy, 2013; Kashkoush & ElMaraghy, 2017; G. Kim et al., 2016; N. Zhang, Yang, Zheng, & Su, 2019) as interfaces are undirected in most cases[44]. In terms of configuration, asymmetric entries become vital. For example, whether A requires B or B requires A are two different statements since, in the first case, B can exists without A. For the calculation of measures, however, each DSM is transformed into a symmetric matrix a priori. This is done as it is assumed that two elements with configurational dependencies will likely have common interfaces that are often undirected.

Figure 3.8
Transformation of an adjacency matrix to a graph and vice versa.



The structural complexity (SC) is the first DSM measure. In product design, SC is used in several studies (Hennig et al., 2022; Min et al., 2016; Sinha, 2014; Sinha & Suh, 2018; Sinha & Weck, 2012, 2013b). Equation (3.27) defines SC consisting of two parts. C_1 represents the

[43] Ameri et al. (2008) differ between graph-based and entropy-based measures. However, some measures (e.g., Neumann entropy) apply the Shannon entropy on graphs. Therefore, a clear distinction seems unfruitful.

[44] An example where directed interfaces might be meaningful are electromagnetic waves, where the source sends to a receiver without getting any data from the target back. For example, G. Kim et al. (2016) use unsymmetric DSMs for a material flow within the product.

overall complexity, with α_i being a weighting factor for the complexity of each individual element. The higher α_i the more complex the i-th's element is (Sinha, 2014). Interface and topological complexity are captured by the second part where C_2 counts the number of interfaces and is scaled by β which weights the interface complexity with the component complexity (C_2) (Sinha & Suh, 2018). C_3 represents the topological complexity via the matrix energy $E(DSM)$ (G. Kim et al., 2016) and N refers to the number of elements equivalent to graphs' vertices.

$$SC = C_1 + C_2 * C_3 = \sum_{i=1}^{N} \alpha_i + \left[\sum \beta * DSM\right] * \frac{1}{N} E(DSM) \tag{3.27}$$

While C_2 represents local effects in the DSM (pairwise interactions), global structures such as the architecture are captured by C_3 (Sinha & Weck, 2013b). More specifically, $C_2 * C_3$ reflects the arrangement of matrix elements and expresses the topology of the DSM (Sinha & Suh, 2018). The term C_1, represents the sum of individual component's complexity where this work assumes that $\alpha_i = 1 \; for \; j = 1 \dots N$. Therefore, the factor equals to the number of elements ($C_1 = N$) represented by the number of rows or columns. Sinha (2014) proposes different heuristics for estimating β. For mechanical connections $\beta^{mech} = 0.5$ is suggested to avoid counting interfaces twice. Flow or energy are unidirectional as they have a specific source and target and thus $\beta^{flow} = 1$. This work assumes that all connections are undirected and defines $\beta = 0.5$ for all elements leading to:

$$SC = C_1 + C_2 * C_3 = N + \frac{1}{2} \sum DSM * \frac{1}{N} E(DSM) \tag{3.28}$$

Matrix energy $E(DSM)$ is defined as sum of singular values $\sum_{i=1}^{n} \sigma_i$ in a DSM and is estimated via singular value decomposition as:

$$DSM = \sum_{i=1}^{N} \sigma_i E_i \tag{3.29}$$

The singular values σ_i represent the degree of re-constructability. The less the topological complexity, the higher the degree of re-constructability (Liu, Wang, & van Mieghem, 2010; Sinha & Weck, 2013b). Lower values indicate more centralized designs, whereas higher values represent distributed designs (Sinha & Weck, 2013a). Table 3.12 shows the corresponding structural complexity values for six different DSMs representing integral, modular, or bus designs, according to Min et al. (2016). A size-normalization of SC is only possible for the term $C_2 * C_3$. For $\beta = 0.5$ coupling complexity (C_2) is scaled by the maximum possible coupling complexity, reached when DSM has no zero entries except the diagonal ones.

$$C_2 \leq C_{2,max} = \frac{1}{2}(N^2 - N) \tag{3.30}$$

An upper bound for the graph energy $E_{max}(DSM)$ is reported by several authors (Koolen & Moulton, 2001; McClelland, 1971). For any graph G holds[45]:

[45] There exist more specific upper bounds for the energy which, however, apply only if the graph fulfills certain pre-conditions such as regularity. For a full overview see: Koolen and Moulton (2001)

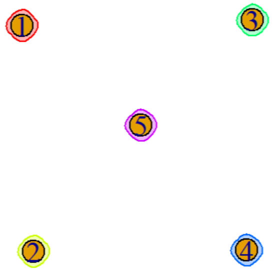
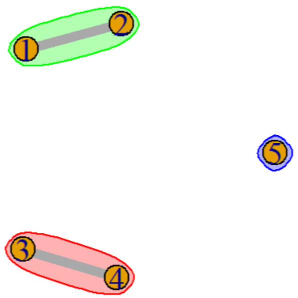
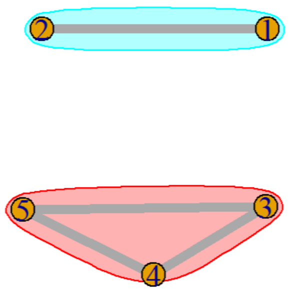
$$E(DSM) \leq E_{max}(DSM) = N * \sqrt{N-1} \tag{3.31}$$

where N is the number of elements (vertices of DSM). This results in a partially normalized structural complexity measure where the C_1-term is dropped:

$$SC_N = \frac{\sum DSM}{(N^2 - N)} * \frac{E(DSM)}{N^2 * \sqrt{N-1}} \tag{3.32}$$

Table 3.12

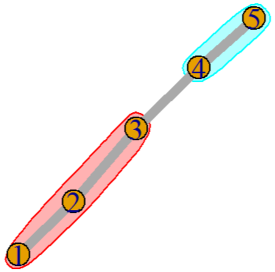
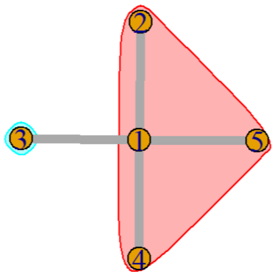
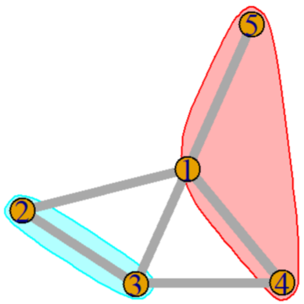
Six matrices, representing different possible dependency structure matrices with their corresponding complexity measures

#	Design	Graph	Matrix[2)]	SC	SC_N	Q[1)]	NE	NE_N	MCC	MCC_N	HVM	HVM_N	HIC	HIC_N
1	perfect modular (5 modules)		$\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}$	5	0	0	0	0	0	0	23.03	0.29	5	0.25
2	modular (3 modules)		$\begin{pmatrix}1&1&0&0&0\\1&1&0&0&0\\0&0&1&1&0\\0&0&1&1&0\\0&0&0&0&1\end{pmatrix}$	6.6	0.02	0.5	4	0.07	5	0.06	36.95	0.46	9	0.45
3	modular (2 modules)		$\begin{pmatrix}1&1&0&0&0\\1&1&0&0&0\\0&0&1&1&1\\0&0&1&1&1\\0&0&1&1&1\end{pmatrix}$	9.8	0.05	0.375	11.51	0.20	5	0.29	52.03	0.65	13	0.65

Note. 1) greedy algorithm for the a-priori community detection used

Table 3.12 (continued)

Six matrices, representing different possible dependency structure matrices with their corresponding complexity measures

#	Design	Graph	Matrix[2)]	SC	SC_N	Q[1)]	NE	NE_N	MCC	MCC_N	HVM	HVM_N	HIC	HIC_N
4	linear-modular		$\begin{pmatrix}1&1&0&0&0\\1&1&1&0&0\\0&1&1&1&0\\0&0&1&1&1\\0&0&0&1&1\end{pmatrix}$	9.4	0.04	0.219	10.46	0.18	5	0.29	52.03	0.65	13	0.65
5	bus		$\begin{pmatrix}1&1&1&1&1\\1&1&0&0&0\\1&0&1&0&0\\1&0&0&1&0\\1&0&0&0&1\end{pmatrix}$	8.2	0.03	0	11.61	0.20	5	0.29	52.03	0.65	13	0.65
6	distributed/ non-modular		$\begin{pmatrix}1&1&1&1&1\\1&1&1&0&0\\1&1&1&1&0\\1&0&1&1&0\\1&0&0&0&1\end{pmatrix}$	12.3	0.07	0	21.61	0.37	9	0.53	68	0.85	17	0.85

Note. 1) greedy algorithm for the a-priori community detection used

Based on graph theory, modularity (Q) measures the degree of coupling within a sub-group compared to the overall coupling (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008; Leicht & Newman, 2008; Newman, 2006). Values for Q vary between $0 \leq Q \leq 1$ where larger values indicate the presence of a community (Newman, 2006). In general, a community is characterized by many connections within a community and only some or no connections between communities. Therefore, modular structures reach higher values of Q compared to randomly distributed designs. A disadvantage of modularity is the definition of modules a priori, where this work uses the Greedy algorithm. Another disadvantage is measure's explainability. Table 3.12, for example, shows modularity values of $Q = 0$ for three matrices. Since this measure is already normalized within a zero-one interval, no further action is required. For the mathematical definition of Q see Blondel et al. (2008).

Like SC, the Neumann entropy (NE) represents the topology of a graph as it measures its regularity (K. Anand, Bianconi, & Severini, 2011; Passerini & Severini, 2009). Assuming a constant number of elements Passerini and Severini (2009, p. 59) note, NE is *"smaller for graphs with large cliques and short paths, i.e., graphs in which the vertices form an highly connected cluster"*. Ai (2017) notes that entropic graph measures reflect the degree of network organization. NE is calculated by applying the Shannon entropy on graphs. Let λ be the eigenvalues of the Laplacian Matrix $L(DSM)$, then the Neumann entropy is given as:

$$NE = \sum_{i=1}^{N} \lambda_i * log_2(\lambda_i) \tag{3.33}$$

Since empty DSMs are possible (no intra-domain connections) the eigenvalues can reach values of zero ($\lambda = 0$) which will result in an undefined entropy. This work, therefore, follows a widely applied assumption in a context of the Shannon entropy that $0 * log_2(0) = 0$ (Cover & Thomas, 2005; Passerini & Severini, 2009). The eigenvalues (λ) are defined as:

$$L(DSM)v = \lambda v \tag{3.34}$$

where $L(DSM)$ is the Laplacian matrix of DSM and v the eigenvectors. With the degree matrix of DSM denoted as $\triangle DSM$, $t$he Laplacian matrix of DSM is given as:

$$L(DSM) = DSM - \triangle DSM \tag{3.35}$$

Normalization is done by estimating the maximum eigenvalues of a binary matrix ($DSM_{bin} \epsilon \mathbb{N}_{0,1}^{nxn}$) which are at maximum $\lambda_{max} \leq n$. Therefore, the normalized Neumann entropy is given as:

$$NE_N(DSM_{bin}) = \frac{\sum_{i=1}^{N} \lambda_i * log_2(\lambda_i)}{n^2 * log_2(n)} \tag{3.36}$$

McCabe's (1976) cyclomatic complexity (MCC) measures the number of feedback loops within a system. It is based on the assumption that more loops (higher values of MCC) increase complexity due to higher interdependency between elements (Hennig et al., 2022). It is defined as:

$$MCC(DSM_{bin}) = \sum DSM_{bin} - n + 2 \tag{3.37}$$

and measures the number of unique paths through a graph represented by the binary DSM. Normalization is done by using the maximum value by assuming a fully connected graph with an adjacency matrix $DSM_{bin,max}$ as:

$$MCC_n(DSM_{bin}) = \frac{MCC(DSM_{bin})}{MCC\left(DSM_{bin,max}\right)} \tag{3.38}$$

Halstead's volume measure (HVM) is entropy-based and describes the amount of information to describe the intra-domain design (Halstead, 1979; Hennig et al., 2022; Prather, 1984). It is defined as:

$$HVM(DSM_{bin}) = (N + E) * log_2(N + E) \tag{3.39}$$

The number of elements (rows of DSM_{bin}) are denoted with E and the number of entries where the main diagonal is assumed to be zero and the number of elements defined as N. Normalization is done by calculating the dividing HVM through the maximum possible value HVM_{max}, which is defined as:

$$HVM_{max}(DSM_{bin}) = ([N^2 - N] + E) * log_2([N^2 - N] + E) \tag{3.40}$$

Lower values indicate a system with only a few interconnections, whereas large values reflect a design with many interconnections. Like MCC, this measure is mainly used in software engineering to indicate code complexity (e.g., Alfadel, Kobilica, & Hassine, 2017; Hariprasad, Vidhyagaran, Seenu, & Thirumalai, 2017).

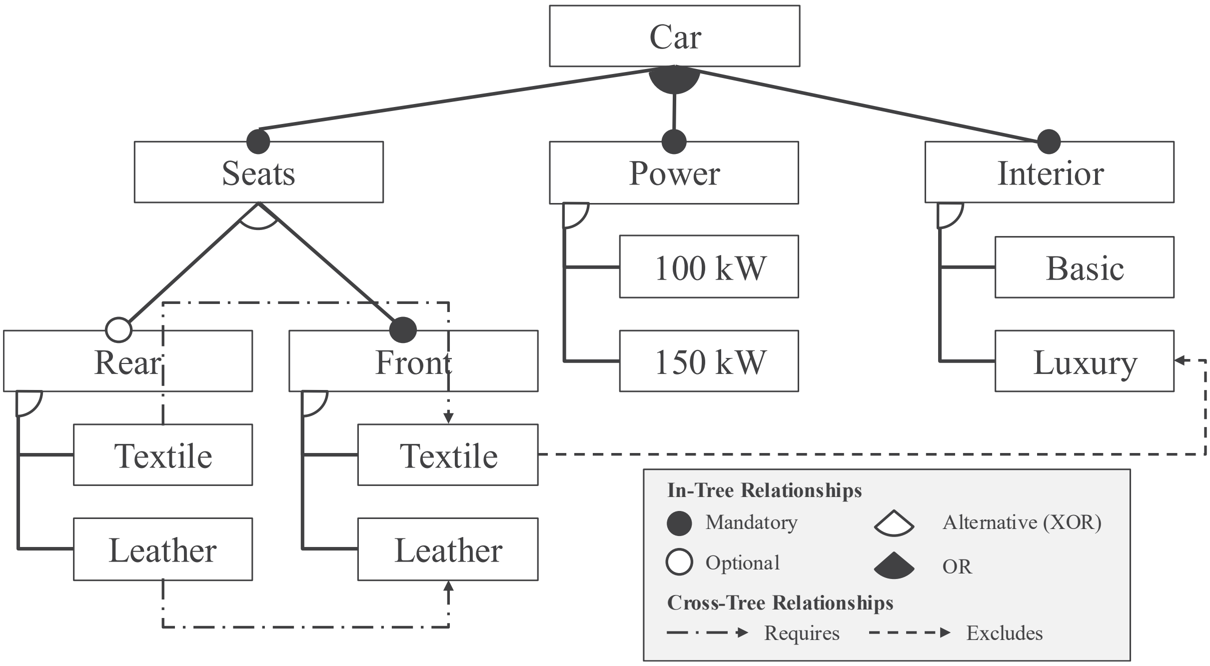
A similar and last intra-domain measure is Hölttä and Otto's (2005) interface complexity (HIC). This measure counts the entries within the DSM_{bin} and assumes that more dependencies lead to more changes and, therefore, complexity if one element is changed. While this measure is quite simple, a downside of MCC, HVM and HIC is that these measures do not reflect changes in matrices' topology. Matrix 3 - 5 from Table 3.12 all have 13 entries but at different locations; however, values for these measures stay constant. This indicates that those measures cannot represent the topological properties.

3.4.2.3 Product Matrix

Seven product matrix measures on the system level are introduced in this section. They are calculated using the product matrices (P) in EAD's product view. In theory, product measures can exist in each domain. However, not every measure is meaningful in every domain. Exclusions of measures for specific domains are explained at the end of this section.

As a count-based measure, the number of product variants (N_{PROD}) operationalizes product variety via the dimension of multiplicity. Only one measure exists for the entire EAD since the number of products does not change across domains. Several authors across various fields use this measure as a proxy for complexity, as the review by Trattner et al. (2019) suggests. In theory, counting the number of product variants is trivial (counting rows of P). In practice, it is more difficult since a combinatorial problem needs to be solved. Car manufacturing companies, for example, have $N_{PROD} = 10^{17}$ (ElMaraghy et al., 2013). A matrix representation of such enormous product variants is not efficient[46]. Facing this problem, software engineering introduced feature models (Kang, 1990) in the 1990s. The literature defines feature models as a sub-type of information models consisting of two parts – features and relationships (D. Benavides, Segura, & Ruiz-Cortés, 2010). Feature models describe each product with a set of features arranged in hierarchical order, where a feature is "*a system property that is relevant to some stakeholder and is used to capture commonalities or discriminate between systems*" (Czarnecki, Helsen, & Eisenecker, 2004, p. 267). Relationships within the model are split into two groups. The hierarchical arrangement of features allows for relationships between a parent and its children (in-tree relations). Cross-tree relationships are the second group and contain information on exclusions or inclusions (D. Benavides et al., 2010). A simple feature model for a car product family is shown in Figure 3.9.

Figure 3.9
Feature model for a car product family. Notation according to D. Benavides et al. (2010)



Based on the given feature tree, each car is described by four features (front seat, rear seat, power, and interior), with two choices for each feature. The options are textile and leather for the front and rear seats, two power choices (100 kW, 150 kW), and two interior choices (basic, luxury). In Figure 3.9, the in-tree relations are expressed using a feature attribute and information on group cardinality. The feature attribute contains the information whether the element is mandatory or optional. For example, each car has a specific power, whereas rear seats are not required for roadsters. The feature model allows for counting the number of possible product variants by multiplying the number of choices for each feature. The model describes 24 unique cars (2*3*2*2)[47], with only five features and eight choices when crosstree relationships are neglected. Cross-tree relationships reduce the number of products. In this example,

[46] To differentiate products from each other, an identifier with at least 57 bits ($2^{57} = 1.44 * 10^{17}$) is required. The total amount of storage capacity only for the identifier alone will be $7.125 * 10^{17} byte = 712.5\ PB$.

[47] A factor of three is caused by the element 'Seats: Rear', as the no-choice must be included, too.

these conditions define that customers must choose either leather or textile. They cannot choose different seat types for the front and the rear. A second condition states that combining textile seats and the luxury interior is not possible. These two constraints reduce the number of product variants from 24 to 12. For the constrained feature tree, the constraining intra-domain matrix, as defined in equation (3.3) reveals as follows:

$$DSM_{FD} = \begin{pmatrix} Rear: T \\ Rear: L \\ Front: T \\ Front: L \\ Pwr: 100 \\ Pwr: 150 \\ Int: B \\ Int: L \end{pmatrix} = \begin{pmatrix} \emptyset & \Psi & \Omega & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \Psi & \emptyset & \emptyset & \Omega & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \Psi & \emptyset & \emptyset & \emptyset & \Psi \\ \emptyset & \emptyset & \Psi & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \Psi & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \Psi & \emptyset & \emptyset & \emptyset \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \Psi \\ \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \emptyset & \Psi & \emptyset \end{pmatrix} \quad (3.41)$$

Where Ψ and Ω are proxies for expressions (non-empty entries) in the conjunctive normal form and defined as $\Psi = -1; -1$ and $\Omega = -1; 1$. $\Psi_{1,2}$, for example, states that if a textile rear seat is chosen, a rear leather seat cannot be selected. $\Omega_{1,3}$, describes that a rear textile seat requires a front textile seat. However, $\emptyset_{1,3}$is empty as a textile front seat does not require any rear seat since, otherwise, the coupé will be excluded from the product mix. As illustrated by the example, feature models allow to reduce the amount of information for describing a system. Therefore, feature models are suitable for systems whose elements share more commonalities than differences (D. Benavides et al., 2010).

The idea of element combinability is used to calculate a normalized product variety measure ($N_{PROD,N}$). While the unnormalized measure directly counts the number of products, $N_{PROD,N}$ is a measure for the combinatorial restriction. Large values of $N_{PROD,N}$ represent a less restricted product mix and, therefore, a large relative number of product variants, whereas low values represent a strongly restricted product mix and, thus, a low variety. Table 3.13 reports four product mixes (PM I - PM IV) with six products each. Therefore, $N_{PROD} = 6$ for all product mixes. The maximum number of products ($N_{PROD,max}$), however, differs across product mixes. While the first product mix (PM I) represents a strongly constrained product mix $N_{PROD,max,PM\ I} = 71$ and, therefore, $N_{PROD,N} = 0.08$ the number of possible products is much lower for the remaining product mixes ($N_{PROD,max,PM\ II} = N_{PROD,max,PM\ III} = 15$ and $N_{PROD,max,PM\ IV} = 23$) and, thus, $N_{PROD,N}$ is larger.

Table 3.13
Four product matrices with the calculated product measures on system level.

#	P_{RD}	N_{PROD}	$N_{PROD,N}$	D[1)]	PCI	DNS	CI	OV	OV_N
PM I	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 7 \\ 0 & 0 & 5 & 5 \\ 0 & 2 & 0 & 1 \end{pmatrix}$	6	0.08	0.80	0.29	0.33	0.17	1.82	0.91
PM II	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}$	6	0.40	0.79	0.29	0.33	0.50	1.82	0.91
PM III	$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$	6	0.40	0.69	0.66	0.67	0.25	1.82	0.91
PM IV	$\begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \\ 2 & 2 & 1 & 1 \\ 1 & 2 & 2 & 1 \end{pmatrix}$	6	0.26	0.61	1	1	0.12	0	0

Note. 1) the Diversification index was calculated assuming a uniform distributed demand vector which means that each product has the same share on sales.

Based on the Herfindahl Index, Gollop and Monahan (1991) introduce the Diversification index (D), which measures product differences via the Manhattan distance matrix of resource usage weighted by the demand distribution. Gollop and Monahan (1991), as well as Gollop (1997) use this measure in the context of organizational science. However, other studies in management science (e.g., Brahm, Tarzijan, & Singer, 2017) use the index too. As a composite index, it accounts for product differences and the demand distribution across products. The measure is defined as follows:

$$D = \frac{1}{2}\left[1 - \sum_{i=1}^{N_P} s_i^2 + \sum_{i=1}^{N_P} \sum_{k=1,k\neq i}^{N_P} s_i * s_k * \sigma_{ik}\right] \tag{3.42}$$

with s_i, s_k being products proportion on total sales for product P_i and P_k defined as

$$s_i = \frac{DMD_i}{\sum DMD} \tag{3.43}$$

Dissimilarity between product P_i and P_k is denoted as σ_{ik} and defined via the Manhattan distance, defined as:

$$\sigma_{ik} = \sqrt{\frac{1}{2}\sum_{j=1}^{N_{RD}} \left|P_{RD,norm,ij} - P_{RD,norm,kj}\right|} \tag{3.44}$$

where $P_{RD,norm}$ is the normalized resource consumption pattern given as:

$$P_{RD,norm} = \frac{P_{RD}}{\sum P_{RD}} \tag{3.45}$$

Gollop and Monahan (1991) note that $D = 0$ if a firm produces only one product. Perfect diversification ($\lim D = 1$) is reached when each individual product share on sales reaches zero ($s \to 0$, homogenous demand distribution) and products do not share any elements across each other ($\sigma \to 1$). Table 3.13 shows four product matrices with the corresponding Diversification index values and a homogenous demand (each product has the same share of sales). The Diversification index summarizes information from EAD's product matrices ($P_{FD}, P_{PD}, P_{PrD}, P_{RD}$) and the demand vector (DMD). However, Trattner et al. (2019) note that such composite measures are informationally dense and less understandable in practice.

Kota et al. (2000) introduced the product line commonality index (PCI) which captures the degree of component sharing across products. Assuming that each component can be potentially standardized and is equally important[48] PCI is defined as:

$$PCI_{PD}(P) = \frac{\sum_{1=1}^{N_P} \beta_i - \sum_{i=1}^{N_P} \frac{1}{\beta_i^2}}{N_{PD} * N_P - \sum_{i=1}^{N_P} \frac{1}{\beta_i^2}} \tag{3.46}$$

where β are the column sums of P calculated as:

$$\beta = \sum_{j=1}^{N_{PD}} P_{bin} \tag{3.47}$$

And $P_{PD,bin}$ the binary product matrix of P_{bin} given as:

$$P_{bin} = \begin{cases} 1, & P_{PD} > 0 \\ 0, & P_{PD} = 0 \end{cases} \tag{3.48}$$

According to the measure's definition, it is only defined for the product matrix where products are described by components (P_{PD}). However, as for the Diversification index, the concept can be extended to other domains as well. Since PCI is defined as a ratio between the current and optimal degree of commonality, its ranges lie within the bounds of zero and one. While $PCI = 0$ indicates no element sharing between products, $PCI = 1$ indicates the maximum possible degree of standardization. Kota et al. (2000, p. 407) note, *"the closer the PCI index is to 1, the better are the firm's efforts towards a value optimizing design".* Table 3.13 reports the individual PCI values for the different matrices and shows that increasing commonality results in a decrease of non-zero values in the product matrices.

Like PCI, DNS is a proxy for element sharing and used in different cost accounting studies (V. Anand et al., 2019; Balakrishnan et al., 2011; Homburg et al., 2018). It is defined as the proportion of non-zero entries within a given product matrix as:

---

[48] Equal importance means that each component is assumed to be equally complex in terms of shape, size, materials, manufacturing, and assembly. Therefore, f_1, f_2, f_3 defined by Kota et al. (2000) are identical for all components.

$$DNS = \frac{\sum P_{RD,bin}}{N_{RD} * N_P} \tag{3.49}$$

where $P_{RD,bin}$ is the binary resource consumption pattern according to equation (3.48). The possible values of DNS are within the bounds of zero and one. Balakrishnan et al. (2011) note that low values represent environments with high degree of element traceability since only some products require the elements (such as in job shops), whereas high values indicate an high level of element sharing (such as in shop floors). Again, Table 3.13 show the individual values for DNS. For product matrices with a large proportion of non-zero entries, $DNS{\sim}PCI$ since:

$$\lim_{\beta \to \infty} \sum_{i=1}^{N_P} \frac{1}{\beta_i^2} = 0 \tag{3.50}$$

which results in:

$$DNS = \frac{P_{bin}}{N_{RD} * N_P} = \frac{\sum_{1=1}^{N_P} \beta_i - 0}{N_{RD} * N_P - 0} = PCI \tag{3.51}$$

Therefore, PM III – IV show almost identical values for both measures.

The commonality index (CI) was introduced by Martin and Ishii (1996). It is defined as the proportion of unique part numbers (u) divided by the sum of parts of total parts within each product (p_i).

$$CI = \frac{u}{\sum p_i} \tag{3.52}$$

Low values refer to a highly standardized product mix, whereas high values refer to a less standardized mix. The calculated values for CI in Table 3.13 highlight two caveats of this measure. Although CI is already within a zero-one bound, it strongly depends on the number of products as well as components available. Second, values are misleading if products have components multiple times. While $CI_{PM\,IV} = 0.12$ indicates a low degree of commonality, $PCI_{PM\,IV} = 1$ and a view on the product matrix, however, clearly indicates a perfect commonality.

The option variability (OV) describes the variation of features across products by comparing the differences between an actual product mix and a random product mix (Fisher & Ittner, 1999; Macduffie et al., 1996). For a binary product matrix $P_{bin} \in \mathbb{N}_{0,1}^{N_{PROD},N_{FD}}$ with N_{FD} options, this measure is defined as:

$$OV(P_{bin}) = \sum_{j=1}^{N_{PROD}} \sqrt{\mu_j(1-\mu_j)} \tag{3.53}$$

μ_j denotes to the proportion of products which have the option j. Small values of OV indicate a low variation of options across products, whereas high values indicate a more homogeneous distribution. Normalization (OV_n) is done by dividing OV through the maximum OV_{max} as:

$$OV_{max}(P_{bin}) = N_{PPROD} * \sqrt{0.5(1-0.5)} \tag{3.54}$$

Table 3.13 shows values for the unnormalized and normalized option variability. Since PM I – PM III all have the same values, this measure does not reflect differences between the product mixes as good as other measures introduced in this section. Another disadvantage is that no engineering-related study was found using this measure.

PCI, DNS and CI all represent the commonality of domain elements across products and are already standardized within a zero-one interval. They are defined in different domains, where Table 3.14 provides an overview. The commonality index (CI) and the more robust product line commonality index (PCI) are defined in the physical domain. While Martin and Ishii's (1996) definition of CI is not limited to binary matrices, the examples indicate misleading values for non-binary matrices. Therefore, this measure is limited to the functional domain in this work’s context since P_{FD} are the only binary matrices. By definition, PCI assumes a binary matrix and, therefore, can represent the commonality of functional requirements, processes, or resources as well. It is shown that for sufficiently dense matrices $PCI = DNS$. Product line commonality and matrices' density are applicable to all domains as they first transform a product matrix into a binary matrix. Increasing values of commonality-related represent designs with more homogeneity and, therefore, less complexity. For the remaining measures, higher values refer to designs with more heterogeneity and, therefore, more complexity.

Table 3.14:
Overview on product view measure’s definition.

Measure	P_{FD}	P_{PD}	P_{PrD}	P_{RD}
N_{PROD}	defined	-	-	-
D	meaningful	meaningful	meaningful	defined
PCI	meaningful (binary)	defined (binary)	meaningful (binary)	meaningful (binary)
DNS	meaningful	meaningful	meaningful	defined
CI	meaningful (but misleading for non-binary)	defined (but misleading for non-binary)	-	-
OV	defined (binary)		-	-

The number of product variants (N_{PROD}) is independent of the mapped domain since each product exists in each domain simultaneously. The diversification index (D) is defined in the resource domain, however, the concept is meaningful for other domains as well. Option variability (OV) is defined in the functional domain and requires a binary matrix. The next section introduces two more measures that describe patterns of the demand and cost vector.

3.4.2.4 Cost and Demand Vector

Since the ABL model already contains a resource cost vector (RC), existing model measures are carried over. V. Anand et al. (2019) and Balakrishnan et al. (2011) measure resource costs’ distribution via the proportion of costs in the top 10% of resources on total costs (TC). This measure is defined as:

$$RC_{T10\%} = \frac{\sum_{j=1}^{\lceil 0.1*N_{RD} \rceil} RC_{j,>}}{\sum_{j=1}^{N_{RD}} RC_j} \tag{3.55}$$

where $\lceil 0.1 * N_{RD} \rceil$ indicates the top 10% of resources rounded upwards to the next integer and $RC_{>}$ describes the resource cost vector arranged in decreasing order. This measure lies within a range of $0.1 \leq RC_{T10\%} < 1$ and is independent from the total costs which are defines as the sum of resource costs. Low values indicate, a homogenous distribution of costs across resources and high values a skewness in cost distribution.

The same principle is used for the demand product demand vector (DMD) to describe demand diversity. Demand diversity has two dimensions, as Jensen, Malhotra, and Philipoom (1996) note. The first dimension describes the static distribution of product demand, where a uniform distribution reflects a lower variability (homogeneity) and a highly skewed distribution a high variability (heterogeneity). The second dimension is the time dependency. A constant demand over time reflects lower variability, and non-constant rates a high variability. Since the EAD is a single-period model, time dependency is neglected, reducing demand diversity to the aspect of static distribution. This allows using the same principle for RC to describe the demand distribution as follows:

$$DMD_{T10\%} = \frac{\sum_{i=1}^{\lceil 0.1 * N_P \rceil} DMD_{i,>}}{\sum_{i=1}^{N_P} DMD_i} \tag{3.56}$$

Again, low values indicate a uniformly distributed demand across products. High values represent firms where some products are responsible for a large proportion of the total demand.

3.4.3 Product Level Measures

Product-level measures produce exactly one value for each product (rows of P matrices), while system-level produce one value for the entire matrix. They exist only in EAD's product matrices P. Table 3.10 introduces three product-level measures, starting with M. Gupta's (1993) inter- and intra-product heterogeneity. Inter-product heterogeneity is defined as:

$$INTER_i = \sum_{j=1}^{N_{RD}} \left(\frac{P_{cnorm,ij} - \overline{P_{cnorm,j}}}{\overline{P_{cnorm,j}}} \right)^2 \tag{3.57}$$

where $P_{cnorm,ij}$ is the column-wise normalized product matrix. This measure describes the position of a product compared to the overall product mix (Mertens, 2020). If $INTER = 0$, the product represents the average product, placed at the center of the product mix; in case of P_{RD}, the product with an average resource consumption. According to equation (3.57) values for this measure are within the bounds of $0 \leq INTER < Inf$. Large values indicate products that have a large difference in resource consumption compared to the average product. Intra-product heterogeneity is seen as a measure of production complexity (Mertens, 2020) and measures the variation of resource consumption across one product. This measure is defined as:

$$INTRA_i = \sum_{j=1}^{N_{RD}} \left(\frac{P_{rnorm,ij} - \overline{P_{rnorm,i}}}{\overline{P_{rnorm,i}}} \right)^2 \tag{3.58}$$

with $P_{rnorm,ij}$ being the row-wise normalized product matrix. This measure lies withing the boundaries $0 \leq INTRA < Inf$. Low values indicate that the product consumes each resource by the same amount ($INTRA = 0$), whereas large values indicate a high variation in resource consumption for a product i. While $INTRA$ captures the variation of products' resource consumption and compares it to its row average ($\overline{P_{rnorm,i}}$), $INTER$ takes columns' mean values ($\overline{P_{cnorm,j}}$). Like DNS, $INTER$ and $INTRA$ are defined in the resource domain but can also be extended to all other domains. While these heterogeneity measures are intuitive in interpretation, they have some caveats under product mixes with more than one center. Figure 3.10 shows a product mix with ten products (P1-P10), and Table 3.15 the corresponding measures. Product 5 (P5) has both a low production complexity ($INTRA = 0$) and inter-product complexity ($INTER = 0.01$). Intuition, however, suggests that P5 is different compared to the remaining products. This fact is not represented by either inter- or intra-product heterogeneity. Therefore, another product-level measure is introduced to cover this aspect.

Figure 3.10:
Product mix with ten products (P1-P10) defined by two resources (RES1, RES2)

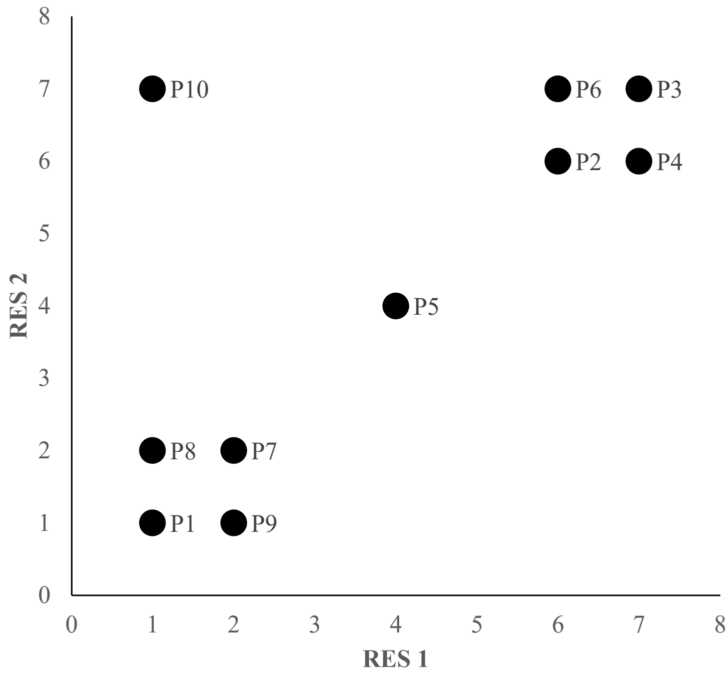


Table 3.15:
INTER, INTRA and LOF for ten products.

Product	RES1	RES2	INTER	INTRA	$LOF_{10\%}$
P1	1	1	1.12	0.00	1
P2	6	6	0.54	0.00	1
P3	7	7	1.19	0.00	1
P4	7	6	0.95	0.01	1
P5	4	4	0.01	0.00	2.83
P6	6	7	0.78	0.01	1
P7	2	2	0.50	0.00	1
P8	1	2	0.82	0.22	1
P9	2	1	0.80	0.22	1
P10	1	7	0.93	1.13	3.00

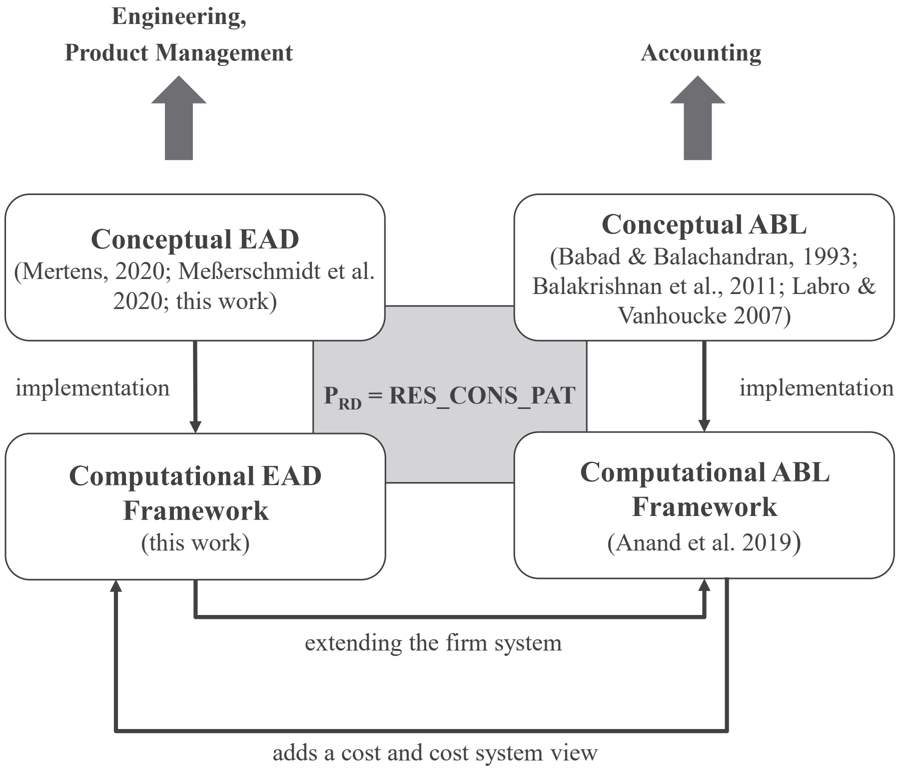
The Local Outlier Factor (LOF) is a measure used in the context of data science (e.g., Xu, Zhang, Li, & Zhu, 2022). It indicates whether a point is an outlier $LOF > 1$ or an inlier $LOF < 1$ and is defined as the proportion of spatial point density compared to its n-nearest neighbors (Breunig, Kriegel, Ng, & Sander, 2000). Therefore, LOF depends on the number of neighbors considered, where $LOF_{10\%}$ indicates the 10% nearest neighbors rounded up to the next integer. Values indicate that P5 and P10 are identified as outliers since their value is above one ($LOF_{10\%,P5} = 2.83$; $LOF_{10\%,P10} = 3.00$). This measure overcomes the limitations of $INTER$ and $INTRA$, however, it has the caveat that the number of nearest neighbors needs to be set exogenously. Breunig et al. (2000) provide some guidelines for choosing the number of neighbors. However, there are no overarching rules of thumb. A too small number of neighbors may fail to identify all outliers if the n-nearest neighbors are very similar but dissimilar to the remaining $N_P - n$ products. Too high values for n would also fail to identify outliers since the subset already represents the distribution of the basic population. For this thesis, $LOF_{10\%}$ is defined, where n is calculated as 10% of the available products. Setting the minimum number of points to 10% ensures that in product mixes with a minimum of 100 products, at least ten neighbors are used for calculation, following the guidelines of Breunig et al. (2000).

This section introduced 20 measures for product family complexity, which represent different dimensions of complexity at different locations within the EAD. All measures have in common that they can be calculated solely by using EAD matrices. No external input, such as constants or experts' judgment, is necessary, which is necessary to use these measures in the context of numerical models. However, such a variety of measures is not suitable for further analysis as it increases model complexity and increases the chance of overlap in measures. For example, it was demonstrated that PCI and DNS are equal for sufficient dense matrices. An overlap in measures further leads to problems of covariation among measures in later analysis. Similarities among these measures are therefore analyzed in the next section by conducting large-scale numerical experiments. In doing so, the EAD simulation model is introduced in the first step before section 3.6 analyzes the measures.

3.5 The Numerical EAD Framework

Numerical simulations are well suited to tackle complicated problems across several research fields for several reasons (Harrison et al., 2007). First, they are not limited by 'real-world' observations where, in practice, the range and quantity of observations are limited. For example, only a small number of case studies report product family. Numerical simulations allow for generating a large variety of observations, enabling a systematic design space exploration. For example, Hennig et al. (2022) use numerical experiments to systematically vary the design of the design structure matrix and investigate how complexity measures behave. A second advantage is that numerical investigations provide answers to questions under uncertainty. An example related to this work is the research on costing system accuracy, where numerical experiments were intensively used over recent years (Babad & Balachandran, 1993; Balakrishnan et al., 2011; Homburg et al., 2018; Labro & Vanhoucke, 2007; Mertens, 2020; Mertens & Meyer, 2018; Schmidt et al., 2023). In practice, firms have limited cost information (V. Anand et al., 2019; Hwang, Evans Ill, & Hedge, 1993). Due to the lack of information, product costing systems report biased product costs (see section 2.6.4). These errors, however, are difficult to observe in practice as they require an unbiased product costing system that requires full cost information. Numerical experiments allow the generation of such full information environments for which the error between reported and true (benchmark) costs can be calculated. Third, numerical experiments can identify complex interactions due to the large number of observations and large design space. In cost accounting, for example, numerical simulations allow for identifying the superposition of errors in costing systems (Labro & Vanhoucke, 2007; Mertens & Meyer, 2018). The latest achievement in this stream of research is the numerical framework by V. Anand et al. (2019) (hereafter: ABL), which is further the connecting point between the engineering view provided by the EAD and the accounting view on costs and costing systems. Figure 3.11 shows the symbiosis between both models.

Figure 3.11:
The resource consumption pattern (RES_CONS_PAT) as connecting element between the Anand model (ABL) and EAD.



The central element is the resource consumption pattern (RES_CONS_PAT), which is equivalent to the product matrix in the resource domain (P_{RD}). While the ABL models a product family via a single matrix by generating randomly correlated numbers, the EAD uses engineering design theory to create patterns in P_{RD}. Therefore, in terms of symbiosis, the EAD extends the firm generation process while the ABL model provides the foundation for cost and product costing system modeling. This section details the linkage between both models. In doing so, the ABL is introduced in the next section (3.5.1) as it works as a starting point for the EAD simulation model. Section 3.5.2 introduces EADs' computational model and extends the simulation model. Empirical evidence on input variable boundaries is provided in section 3.5.3.

3.5.1 The Anand Model (ABL) as a Starting Point

The ABL unifies conceptual and numerical studies published over recent years. Labro and Vanhoucke (2007) and Mertens and Meyer (2018) investigate the impact of measurement errors on product costing systems. Labro and Vanhoucke (2008) analyze the robustness of costing systems under different product mix diversity. Balakrishnan et al. (2011) investigate the impact of different costing heuristics on product costing accuracy. Homburg et al. (2018) simulate the impact of errors on product mix and pricing decisions. Schmidt et al. (2023) analyze the mechanisms behind product cross-subsidization. The model consolidates these individual models into a single framework and consists of two main structures: a firm system that transforms inputs into outputs via a production function and a cost system that represents firms' limited cost information (V. Anand et al., 2019). The remaining part of this section focuses exclusively on the firm system as it is the connecting structure between the EAD and the ABL model. Additionally, elements of the firm system that are not relevant for merging both models are excluded, such as all multi-period variables[49] or introduced in other sections of this work, such as the product costing system in section 5.

Under a linear production technology, input resources are transformed into output products modeled by a resource consumption pattern defined as:

$$P_{RD} = RES\_CONS\_PAT \in \mathbb{N}_0^{N_{PROD} \ x \ N_{RD}} \tag{3.59}$$

N_{RD} defines the number of resources (inputs, columns), and N_{PROD} the number of products (outputs, rows). Patterns within the matrix are created based on four input parameters. DNS defines the proportion of non-zero elements on a technical level and is seen as a proxy for firm's degree of resource sharing (Balakrishnan et al., 2011)[50]. Small values indicate a low level of resource sharing, such as in job shop environments, and high values indicate a high degree of resource sharing, such as in shop floor environments. COR_1 defines the correlation between the first $DISP_1$ resources ($RD_{2 \ldots DISP_1}$) and a base consumption (RD_1). COR_2 defines the correlation between the remaining resources ($RD_{DISP_{1+1} \ldots N_{RD}}$) and RD_1. The base consumption is drawn from a standard normal distribution and defined as the first resource RD_1. The base consumption is necessary to ensure that each product has a non-zero consumption. Based on COR_1 and COR_2 random vectors with the predefined correlation are generated and placed behind the base consumption in RES_CONS_PAT. A separation of $RD_{2 \ldots N_{RCP}}$ allows for defining different correlation values to a set of resources. This reflects the fact that some resources correlate on unit or batch level according to the ABC cost hierarchy (V. Anand et al., 2019; Balakrishnan et al., 2011). Resources with a high correlation are more similar to each other than those with a negative correlation[51]. RES_CONS_PAT is then multiplied by ten and rounded up to the next largest integer. Additionally, the simulation procedure ensures that each resource is used at least once. True (error-free) product costs (PC_B) are calculated as:

$$PC_B = RES\_CONS\_PAT * RCU \tag{3.60}$$

where RCU indicates the resource costs per unit. Based on a given resource cost vector (RCC) and the total resource consumption (TRU) the unit resource costs are calculated as:

$$RCU = \frac{RCC}{TRU} \tag{3.61}$$

TRU is a function, which depends on the production function and the product demand vector (DMD). It is calculated as:

$$TRU = DMD * RES\_CONS\_PAT \tag{3.62}$$

[49] All variables necessary for time-variant analyses, such as PRE_ACAP or POST_ACAP, and capacity variables (e.g., CAP) are not introduced here.

[50] For a more detailed discussion of DNS, see section 3.4.2.3 .

[51] Negative correlated resources show a significant disparity, according to V. Anand et al. (2019)

The resource cost vector RCC is created by distributing the total costs (TC) across resources. During RCC creation it is ensured that the largest $DISP_1$ resources hold $DISP_2$ percentage of the total costs and $TC = \sum RCC$.

Besides all the notable achievements gained by the model over the recent years, there are certain restrictions. Most of the studies[52] use the single-period formulation of the ABL as it dramatically reduces models' complexity and provides a solution for the capacity allocation problem, also called the Grand Program (Balakrishnan & Sivaramakrishnan, 2002). This work makes no exception; therefore, the following discussion focuses only on the single-period formulation. Literature assumes that capacity can be acquired instantaneously (V. Anand et al., 2019) and that no unused capacity exists. Since all capacity is used and can be acquired if needed, all costs are variable. While this reduces model complexity, it neglects that resources are tied to a firm in the short run (Caves, 1980; Wernerfelt, 1984). Non-unit level (fixed) costs need to be added to the model. This becomes important in settings where the product mix and demand vary. The work of Schmidt et al. (2023) is a starting point as they add batch-level costs to the ABL. Another criticism is related to the resource consumption generation procedure. It generates correlated column vectors with a density adjustment in a second step. While this procedure is highly randomized, allowing for a generalization under various settings, such as job-shop or shop floor environments with different levels of production complexity, it may not reflect the resource consumption of configurable product families well. Additionally, observing resource consumption is difficult in practice as it requires full information. Therefore, comparing generated with empirical patterns becomes challenging. A third disadvantage of ABLs' generic approach is the creation of a base consumption. A base consumption is created to ensure that each product has a non-zero consumption and that each resource is used by at least one product. This leads to anomalies during cost system design experiments under a small number of activity cost pools.

In short, the ABL provides a basis for conducting numerical experiments on product costing system design. However, it models a product family solely by the resource consumption of its products, making it difficult for the model to be used beyond accounting since the underlying product family design is not included. The EAD simulation model closes this gap by creating the resource consumption pattern resulting from design matrix multiplications.

3.5.2 The Numerical EAD Framework

Labro (2015a) identifies a niche forming over the last 25 years in accounting research and suggests that accounting could profit from a view beyond its borders[53]. Studies using the ABL are limited to the accounting field so far. However, information on product designs' costs and the accuracy of product costing systems are also relevant for other disciplines, such as

[52] See V. Anand et al. (2017) for an exception.

[53] There are also other studies calling for more interdisciplinary research. Related to the connection between the customer choice process and the effects on supply chain costs, Syam and Bhatnagar (2015, p. 12) note: *"cross-disciplinary research is necessary to bring our theoretical models closer to business reality where decision makers have to look at the overall picture".*

engineering and product management, as they are facing multi-attribute decision-making problems occurring all along the product development process (Chen et al., 2022; Kahraman & Cebi, 2009; Kulak et al., 2010). An example of such a problem is shown in section 3.2.5. The EAD simulation addresses both communities as it combines two already accepted models from engineering design (AD/EAD) and cost accounting (ABL) via the resource consumption pattern. From an accounting perspective, the numerical EAD model extends the process of generating resource consumption patterns to generate empirically related resource consumption patterns. From an engineering perspective, a cost and costing system view is added to the product family design model to estimate the economic consequences of design decisions. Extending the generation process allows for modeling the product design itself rather than only its results in terms of resource consumption. For example, engineering cannot influence the degree of resource sharing directly. However, it can control the physical product design ($DMM_{FD,PD}, DSM_{PD}$), indirectly influencing the resource consumption pattern. Therefore, the EAD simulation provides a more practical engineering and product management starting point than the ABL. A second benefit of the simulation model is that it links product family complexity measures with total costs and cost system accuracy. Accounting uses the degree of resource sharing (DNS_{RD}) as a central measure to describe the patterns in RES_CONS_PAT. However, this measure is rather abstract and difficult to obtain in practice. Instead, the previous section (3.4) introduced 20 different measures used in literature to describe the product design, which results in a certain resource consumption pattern.

The simulation model creates a product family in a three-step approach where detailed documentation and the source code are available online[54]. In the first step, the product mix and product demand are generated. Second, the domains containing the domain mapping matrices (DMM) and dependency structure matrices (DSM) are generated. Finally, costs are generated. The following paragraph briefly introduces the individual steps. A complete control flow chart is provided in thesis appendix A1.

3.5.2.1 Creating Product Mixes and Demand

Three input parameters generate the product mix. The number of functional requirements (N_{FR}), the desired density of the product matrix (DNS_{FD}), and the desired number of products method (N_{PROD}). The free product mix as a power set of functional requirements is generated in the first step, and the empty product (all zeros) is excluded. This results in a binary matrix where each product (rows) is defined by a binary vector of functional requirements as defined in equations (3.1) and (3.2). In the second step, N_{PROD} products are randomly selected. It is ensured that the resulting product mix matrix (P_{FD}) has a certain, user-defined density (DNS_{FD}). Matrix density is defined as the proportion of non-zero entries. This method allows the generation of empirically observed product mixes, characterized by a low density in P_{FD} (see section 3.5.3). The procedure further ensures that each functional requirement is used at least once (no zero columns) and that each product is unique. Following the ABL model, product demand is an exogenous variable. A product demand vector (DMD), therefore, varies in

[54] The code is available as an R package under: https://github.com/olemesser/EAD

three dimensions: first, the number of products which are specified by the current product mix and generated in the previous step; the total demand (TD) defined as the number of total units a firm sells; and distribution of sales across products (Q_{var}). The procedure generates a vector of random integers that fulfills two criteria. First, the sum of entries equals the total demand ($\sum DMD = TD$), and second, the vector has a log-normal distribution with the logarithmic standard deviation equal to $sd(\log(DMD)) = Q_{var}$. Table 3.16 summarizes the input parameter in this first step. More detailed documentation for all procedures related to the computational EAD model is provided in the online manual.

Table 3.16:
Variables for the product mix and demand creation

Variable	Description	low values	high values
DNS_{FD}	defines the desired density in P_{FD}.	product matrix is sparse, meaning that products have only some features selected on average	dense product matrix in which products have many features selected
N_{PROD}	number of products (rows) in P_{FD}.	small product mix with only some products	large product mix with many products
N_{FR}	number of Elements in the functional domain	only a few functional requirements necessary to realize the product mix.	many functional requirements necessary to realize the product mix.
Q_{var}	desired logarithmic standard deviation for the demand vector	sales are equally distributed across products.	some products with a high share on sales and many products with low quantity.
TD	total number of sales for the created product mix	product mix with low number of total sales.	product mix with high number of total sales.

3.5.2.2 Creating Domain Mapping Matrices

In the second step, inter- and intra-domain matrices are generated. The procedure is exemplarily shown for one domain transition and starts with creating the domain mapping matrices (DMM). Source domain elements are noted with the indices src and the target domain elements with the indices tgt. A domain transition consists of an inter-domain matrix $DMM_{src,tgt}$ as well as a dependency structure matrix (DSM_{tgt}) and is defined as:

$$P_{tgt} = P_{src} * [(DMM_{src,tgt} * DSM_{tgt}) + DMM_{src,tgt}] \quad (3.63)$$

P_{src} represents products in the source domain, and P_{tgt} in the target domain. The mapping between the physical domain (PD) and process domain (PrD), as well as between PrD and resource domain (RD), are defined as shown in equation (3.63). The mapping between the functional domain (FD) and PD is an exception since this mapping follows equation (3.12). The procedure for creating $DMM_{src,tgt}$ requires four user-specified input parameters, as shown in Table 3.17. The first two parameters define the dimensions of $DMM_{src,tgt}$, where N_{src} refers to the number of rows and N_{tgt} to the number of columns. All DMMs are directed matrices where the source elements (rows) require target elements (columns). Sampling the patterns of non-zero entries in $DMM_{src,tgt}$, DMM_{PAR} defines the desired normalized system design complexity ($0 \leq SDC_N \leq 1$). The model starts with an empty $DMM_{src,tgt}$ and optimizes the position of non-zero entries until the user-defined value of DMM_{PAR} is reached. Small values of SDC_Nrefer to a more uncoupled design, whereas larger values result in a more coupled design. The framework allows the generation of non-binary DMMs. In doing so, the default value of $uB = 1$ is increased. An increase of uB results in more heterogeneity within

$DMM_{src,tgt}$. Interrelatedness, therefore, increases with increasing SDC_N and uB values. At the end of the generation of the DMMs and DSMs and multiplication according to equation (3.63), it is ensured that each product (rows in P_{tgt}) is unique. Additionally, it is ensured that each target element is used by at least one product.

Table 3.17:
Variables for the domain mapping matrix generation

Variable	Description	low values	high values
N_{src}	number of source elements, rows of the later DMM.	small number of source elements.	large number of source elements.
N_{tgt}	number of target elements, columns of the later DMM.	small number of target elements.	large number of target elements.
DMM_{PAR}	DMM design parameter (SDC_N) which specifies the pattern created.	an uncoupled design matrix (diagonal matrix)	a (strongly) coupled design matrix
uB	upper bound of an integer uniform distribution for integer sampling.	small heterogeneity in design.	large heterogeneity in design.

Figure 3.12 shows different DMMs created via the procedure introduced above. The number of source and target elements is kept constant for better comparison. On top of each subplot, the desired SDC_N value is indicated. If and only if $SDC_N = 0$ an uncoupled design is generated, in all other cases, the design is coupled where the degree of coupling depends on the value of SDC_N.

Figure 3.12:
*Patterns created using EAD's domain mapping matrix procedure. The numbers on top of the subplots represent the normalized system design complexity (**SDC_N**).*

3.5.2.3 Creating Dependency Structure Matrices

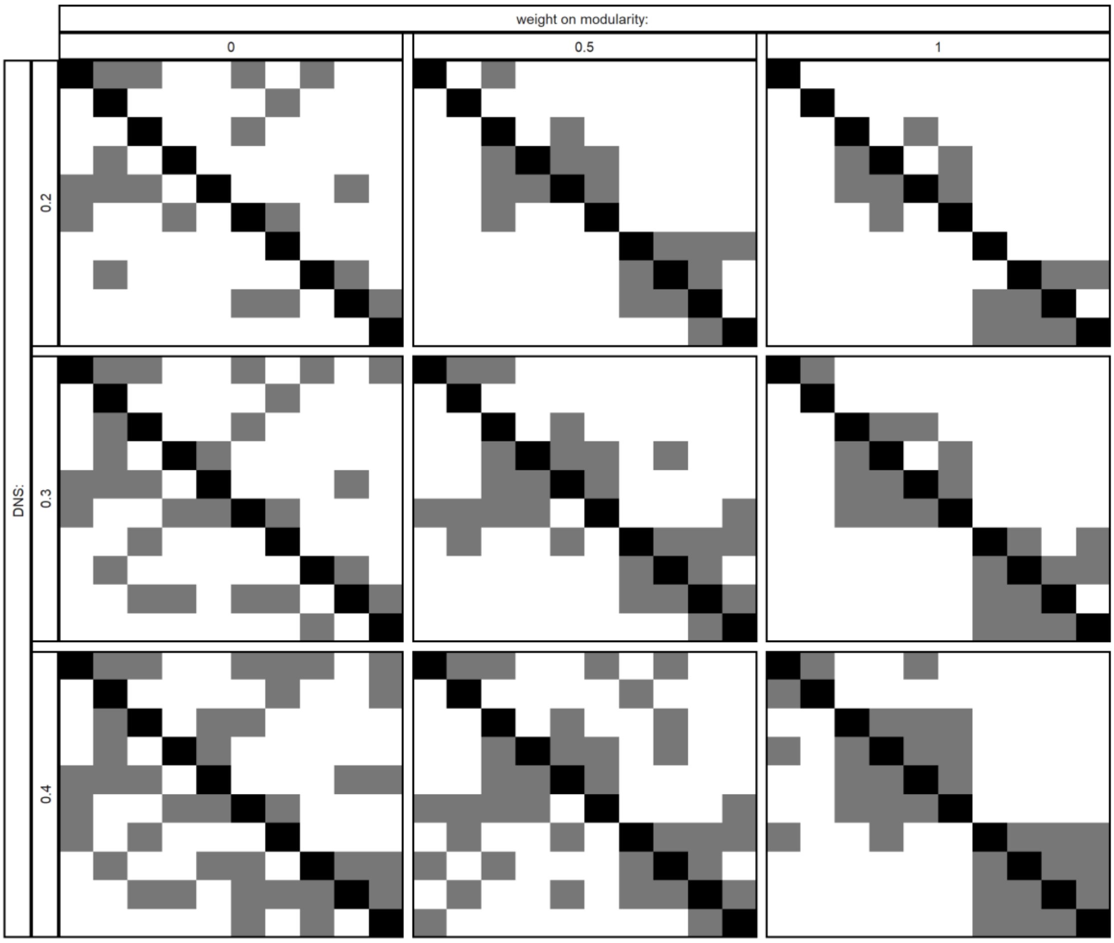
The target dependency structure matrix ($DSM_{tgt} \epsilon \mathbb{N}^{Ntgt \times Ntgt}$) is generated in the second step of the domain generation process. Table **3.18** shows the relevant input parameter for the matrix generation. N_{tgt} defines the size of the DSM. Since DSMs are intra-domain matrices, they are always square matrices. However, they do not need to be symmetrical as they contain configurational knowledge rather than element interfaces. The DSM is generated in a two-stage approach. In the first stage, the distribution of non-zero entries is defined. In doing so, the function uses $0 \leq DNS \leq 1$, which defines the percentage of non-zero elements. Compared to DMMs, DSMs can contain no entries at all. In these cases, the design is solely defined by the DMM, as under the classical AD. The variable $0 \leq modular \leq 1$ defines the focus on the modular structure. If $modular = 0$ values are randomly distributed and if $modular = 1$ the non-zero entries are assigned into diagonal block matrices. Therefore, the variable $modular$ is an input measure for the weight of the modular structure.

Table 3.18:
Variables for the dependency structure matrix generation

Variable	Description	low values	high values
N_{tgt}	number of columns and rows	small number of target elements.	large number of target elements.
DNS	percentage of none zero values	less intra-domain coupling	strong intra-domain coupling
modular	weight on modular structure	random distribution of entries	entries are arranged in block matrices
uB	upper bound of an integer uniform distribution for integer sampling	small heterogeneity in design	large heterogeneity in design

Figure 3.13 shows nine different DSMs with three levels of density (rows) and three levels of modularity (columns). While the first column represents random patterns with different densities, the last column represents a modular structure (e.g., Hennig et al., 2022). The latter is characterized by diagonal block matrices, representing the modules. Modules are characterized by a strong coupling within modules and a weak coupling across modules. In the last step, uB is used to replace all non-zero entries with a random integer number drawn from a uniform distribution $U_{int}[1; uB]$. According to the domain transition, the product matrices are calculated using the product matrix in the source domain (P_{src}), the domain mapping matrix ($DMM_{src,tgt}$), and the target domain dependency structure matrix (DSM_{tgt}). The procedure is repeated for each domain transition. After all domains are generated, the validity of the design is checked. In doing so, two conditions must be fulfilled for each product matrix (P_{PD}, P_{PrD}, P_{RD}). First ($cond1$), each product must be unique to ensure that no product uses the same set of domain elements. Second ($cond2$), every domain element must be used at least once (no zero columns allowed).

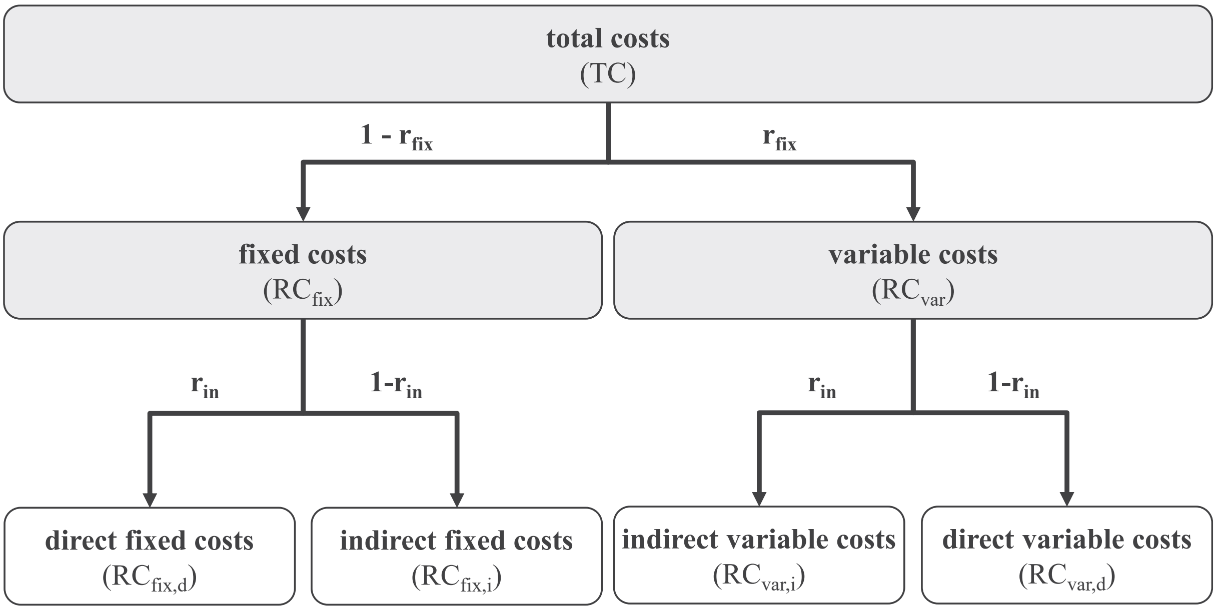
Figure 3.13:
Patterns created using EAD's DSM generation procedure.



3.5.2.4 Creating Resource Costs

In the last step of the EAD generation procedure, costs for each resource are generated. In contrast to the ABC cost hierarchy, this work differs solely between unit- and non-unit-level costs, referring to the traditional costing view. Later parts of this thesis show that most firms use the traditional costing view while only 15 – 40% use ABC (see Table 5.1). According to the traditional view total costs are separated into fixed (RC_{fix}) and variable (RC_{var}) costs. Fixed costs represent firms short- and medium-run. Remaining untouched is the assumption that, in the long run, all costs are variables since firms have control over their acquired capacity (Horngren et al., 2012). Additionally, fixed costs allow the investigation of volume effects, as done in the quadcopter case study (section 3.2.5). The study highlights that a reduction in fixed costs outweighs the increase in variable costs for specific designs. Fixed and variable costs are further separated into direct ($RC_{var,d}, RC_{fix,d}$) and indirect costs ($RC_{var,i}, RC_{fix,i}$). This indicates a difference compared to existing numerical frameworks that model indirect costs solely (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011).

Figure 3.14:
Overview of the cost structure within the numerical EAD model.



The EAD generates these cost vectors in two steps, where Table 3.19 report the input variables. In the first step, variable (RC_{var}) and fixed cost vectors (RC_{fix}) are randomly sampled. It is ensured that these vectors fulfill the following three conditions:

$$RC_{var,j} > 0 \wedge RC_{fix,j} > 0 \,, for\ j = 1 \ldots N_{RD} \tag{3.64}$$

$$\sum RC_{var} = TC * (1 - r_{fix}) \wedge \sum RC_{fix} = TC * r_{fix} \tag{3.65}$$

$$\sigma_{log}^2(RC_{var}) = RC_{log} \wedge \sigma_{log}^2(RC_{fix}) = RC_{log} \tag{3.66}$$

The first condition ensures that the costs for each resource are non-negative. The second condition defines the total fixed and variable costs. The sum of the fixed cost vector, for example, is defined by $TC * r_{fix}$ where TC are the total costs and r_{fix} as the proportion of fixed costs on total costs. The last condition states that the generated cost vectors have a user-defined log-normal standard deviation. In the following step, each cost vector is separated into a direct and indirect cost vector, where r_{in} represents the proportion of indirect costs on total costs.

Table 3.19:
Variables for the cost generation procedure

Variable	Description	low values	high values
N_{RD}	number of resources (length of the cost vector)	production environment with only some resources	production environment with many resources
TC	total costs for the given product family design under a given demand vector DMD.	small total costs	large total costs
r_{in}	proportion of indirect costs compared to total costs.	small proportion of costs are indirect costs	large proportion of costs are indirect costs
r_{fix}	proportion of fixed (indirect) costs compared to total indirect costs.	small amount of fixed costs; indirect costs are mainly variable.	large amount of fixed costs; indirect costs are mainly fixed.
RC_{sdlog}	logarithmic standard deviation for the resource cost vector	direct costs uniformly distributed across resources	some resources are responsible for a large proportion of direct costs

Low values of r_{in} represent environments with mainly direct costs, such as in single-product firms where all costs are direct costs or those with a sophisticated product costing system. Remember, indirect variable costs are also called imaginary indirect costs since they result from PCSs' limitation to trace these costs onto cost objects. The proportion of fixed costs to total costs is defined by r_{fix}. Large values represent firms where a large proportion of costs are fixed, such as in R&D-intensive companies (e.g., aviation or pharma industry). Resource costs' heterogeneity is defined by RC_{sdlog}, representing the log-normal standard deviation of the resource cost vectors. Low values $RC_{sdlog} \rightarrow 0$ indicate homogenous costs across resources, while higher values represent firms where some resources are responsible for the majority of costs.

3.5.3 Empirical Boundaries for Input Variables

The EAD framework is the foundation for conducting large-scale numerical experiments. A design of experiments (DoE) defines input variables, their levels, and ranges. It further provides a foundation for effective and efficient experimental planning, design, and the analysis of results (Antony, 2014). While EADs' variables are defined in the previous section, this section focuses on variables' levels and ranges. Balakrishnan and Penno (2014) highlight the importance of this step. They note that the number of model runs increases exponentially with the number of input variables and levels for each variable. Since the number of variables is defined by the model formulation, this step aims to define ranges for some of the input variables using empirical data. In doing so, DMMs and DSMs, as reported in the literature, are used to estimate variable input ranges for DNS_{DSM} and SDC_N. Table 3.20 summarizes the results, which are discussed in more detail in the following paragraphs. Except for product designs reported by Li et al. (2021) and Morkos et al. (2012), the individual data sets are included in the computational EAD model[55].

[55] It was not possible to extract the data set based on the provided information in those studies. For example, if matrices are included as pictures with low resolution. An overview of available data sets is available by running the following command in 'R' after loading the library *data(package = 'EAD')*.

Table 3.20:
Case studies reporting product designs with the corresponding density (DNS) and normalized system design complexity values ($\boldsymbol{SDC_N}$)

Authors	Study	Matrix	Dim	Variable	Value	DS name[1)]
Sinha & Suh (2018); Kim et al. (2016)	architecture design of a train bogie	DSM_{PD}	149 x 149	DNS_{DSM}	2.1%	trainbogie
Pandremenos and Chryssolouris (2011); AlGeddawyand El-Maraghy (2013); Kashkoush & ElMaraghy (2017)	architecture design of car metal frame	DSM_{PD}	38 x 38	DNS_{DSM}	14.7%	carframe
Li et al. (2021)	architecture design of an electric sanitation vehicle	DSM_{PD}	58 x 58	DNS_{DSM}	-	-
Sawai et al. (2017)	architecture design of an industrial three-axis robot (complete robot)	DSM_{PD}	151 x 151	DNS_{DSM}	1.8%	robot_DSM_full
	architecture design of an industrial three-axis robot (traverse part)	DSM_{PD}	26 x 26	DNS_{DSM}	8.6%	robot_DSM_traverse
		$DMM_{FD,PD}$	32 x 26	SDC_N	0.029	robot_DMM_traverse
	architecture design of an industrial three-axis robot (control part)	DSM_{PD}	17 x 17	DNS_{DSM}	11.8%	robot_DSM_control
		$DMM_{FD,PD}$	30 x 17	SDC_N	0.015	robot_DMM_control
Jung et al. (2022)	architecture design of a power steering system in automobiles (RDU1)	$DMM_{FD,PD}$	31 x 20	SDC_N	0.027	RDU1
	architecture design of a power steering system in automobiles (RDU2)	$DMM_{FD,PD}$	31 x 20	SDC_N	0.036	RDU2
Morkos et al. (2012)	product family of a firm producing automation solutions	DSM_{PD}	214 x 214	DNS_{DSM}	-	-
Bonjour & Micaëlli (2010)	gear box	$DMM_{PrD,RD}$	21 x 14	SDC_N	0.074	gearbox_DMM_PrD_RD
	DSM_{RD} contain duplicated resources which are unified	DSM_{PrD}	21 x 21	DNS_{DSM}	9.1%	gearbox_DSM_PrD
		$DMM_{PD,PrD}$	12 x 21	SDC_N	0.006	gearbox_DMM_PD_PrD

Note. 1) DS name refer to the name of the data set within the EAD package. A list of all available datasets can be requested by using the R commend “data(package = ‘EAD’)” after loading the library.

Several case studies report component-to-component dependencies in DSM_{PD}, such as for a train boogie (G. Kim et al., 2016; Sinha & Suh, 2018), a car metal frame (AlGeddawy & ElMaraghy, 2013; Kashkoush & ElMaraghy, 2017; Pandremenos & Chryssolouris, 2011), a three-axis robot (Sawai et al., 2017), or a gearbox (Bonjour & Micaëlli, 2010). There are also other studies reporting component intra-domain dependencies (e.g., Li et al., 2021; Morkos et al., 2012); however, it was not possible to extract the datasets from those studies. Other studies report domain mapping matrices ($DMM_{FD,PD}$) for an industrial three-axis robot (Sawai et al., 2017) or a power steering system (Jung et al., 2022). A mapping between the physical domain and processes ($DMM_{PD,PrD}$) as well as processes to resources ($DMM_{PrD,RD}$) is reported by Bonjour and Micaëlli (2010). These data sets allow the calculation of six density values (DNS_{DSM}) and six normalized system design complexity values, (SDC_n) where Table 3.21 shows the descriptive statistics. While values close to $DNS_{DSM} = 0$ indicate empty matrices, large values indicate dense intra-domain matrices and, therefore, many in-domain dependencies. The analysis shows that values reach from 1.8% up to 14.7%. Low values of SDC_n indicate sparse DMMs, where $SDC_n = 0$ refers to an uncoupled design. Large values of $SDC_n = 1$ refer to a fully coupled design. The range for this variable is identified as $0.006 \leq SDC_n \leq 0.08$.

Table 3.21:
Descriptive statistics for the density of DSMs (DNS_{DSM}) and the normalized system design complexity for DMMs ($\boldsymbol{SDC_N}$)

Variable	Min	25% Quantile	Median	Mean	75% Quantile	Max	SD	N
DNS_{DSM}	1.8%	3.8%	8.9%	8.0%	11.1%	14.7%	5.2%	6
SDC_N	0.006	0.018	0.028	0.031	0.034	0.074	0.024	6

Empirical data gathered during industry projects is used to estimate input ranges for the product mix variables to generate P_{FD} and DMD. For better comparability, all product mixes are transformed into a binary matrix $P_{FD} \in \mathbb{N}_{0,1}^{N_{PROD} x N_{FD}}$. For example, a product matrix with two functional requirements, weight and speed, where each requirement has two choices, is transformed into a four-by-four binary matrix, as shown below.

$$P_{FD} = \begin{pmatrix} low & fast \\ medium & fast \\ medium & slow \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \tag{3.67}$$

Table **3.22** provides an overview of the analyzed product families. The product matrix is described by its density (DNS_{FD}) as well as the number of functional requirements (N_{FD}) and the number of products (N_{PROD}). Low values of DNS_{FD} indicate heterogeneous product mixes, whereas higher values indicate homogeneous ones. The demand vector is analyzed using $DMD_{T10\%}$ as defined in section 3.4.2.4 and log-normal standard distribution (Q_{var}). The latter is used only for the EAD generation as there exist programming libraries that can create a vector with a given Q_{var} efficiently. This measure is further used by Mertens (2020). For the measurement, however, $DMD_{T10\%}$ is used as it has a defined upper bound ($0.1 \leq DMD_{T10\%} < 1$). An additional measure RU describes the reuse of products and is defined as the total demand divided by N_{PROD}. Low values ($RU \rightarrow 1$) indicate that each product is fully customized, while higher values indicate that customers purchase the same product variant. Table **3.22** reports the values for the eight product mixes, and Figure 3.15 visualizes the demand distribution.

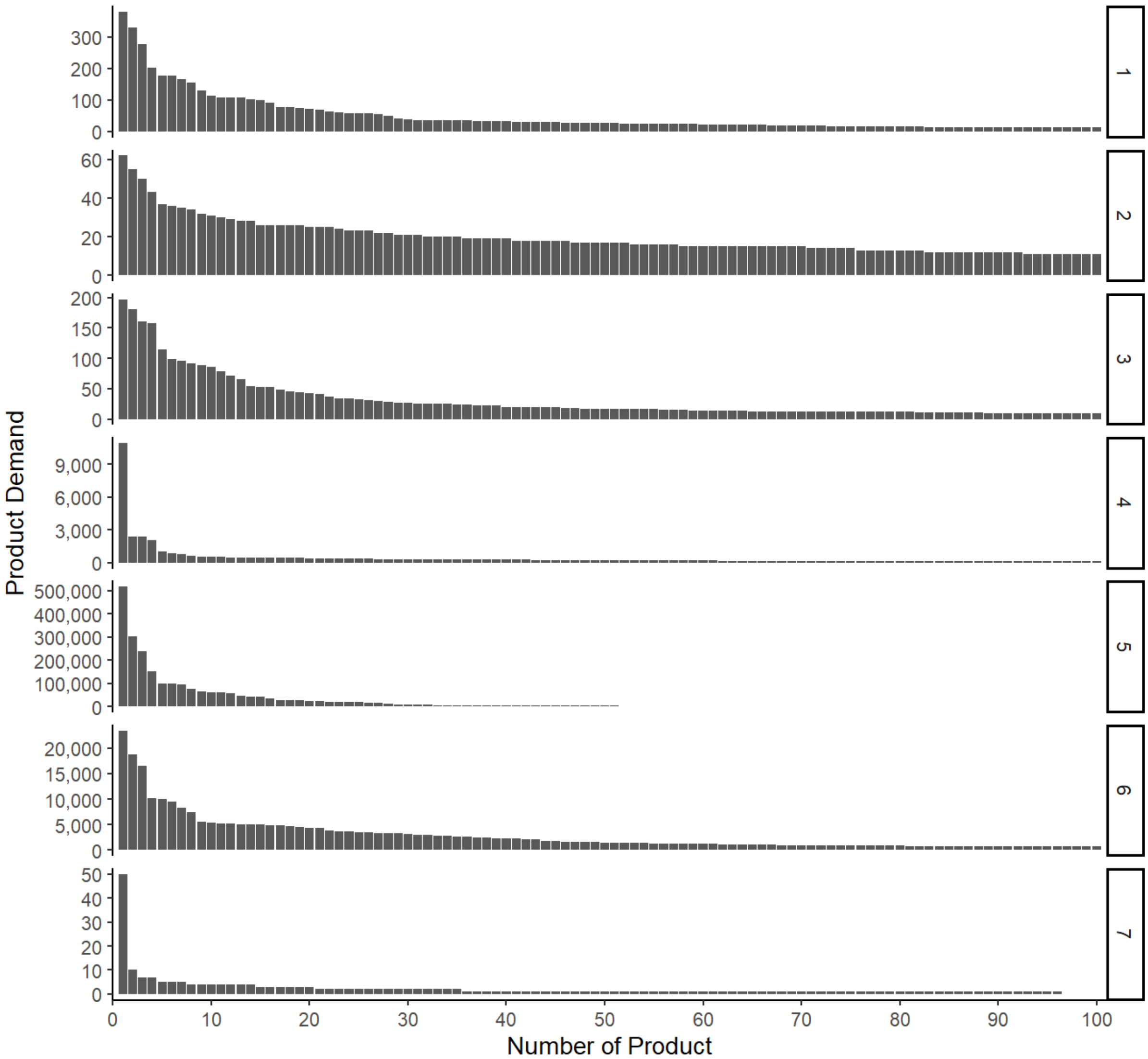
Table 3.22:
Description of eight different product families used for the meta-analysis.

#	Product Family	N_{FD}	N_{PROD}	DNS_{FD}	RU	$DMD_{T10\%}$ 1)	Q_{var} 1)
1	system integrator	77	1277	19%	6.8	41.1%	.826
2	system integrator	94	2099	43%	3.2	21.1%	.384
3	system integrator	120	786	33%	5.8	40.0%	.773
4	component manufacturer	127	434	19%	123	50.3%	.744
5	component manufacturer	130	1154	14%	2.4	70.7%	1.630
6	component manufacturer	195	7665	15%	2.0	39.2%	.850
7	system integrator	280	96	30%	2.4	44.7%	.694

Note. Calculated using the top 100 products with the largest sales share.

The analysis reveals that the product matrices are sparse since P_{FD} is a binary matrix and $DNS_{FD} < 50\%$ for all product mixes. Based on the empirical data, the product mix density is set within the bounds of $8\% \leq DNS_{FD} \leq 50\%$. Product demand vectors show a wide range of patterns, starting with almost uniformly distributed demand ($Q_{var} = .384$; $DMD_{T10\%} = 21.1\%$) up to a highly skewed distribution in product mix five, where ten percent of the products are responsible for 70.7% of the total sales ($Q_{var} = 1.63$). Therefore, a log-normal distribution within the bounds of $0 \leq Q_{var} \leq 1.7$ is suggested.

Figure 3.15:
Distribution of product demand for seven empirical product mixes. The first 100 products are sorted in decreasing order.



3.6 Similarity among Measures

These empirical observations are used to model variables' input ranges for the numerical experiments conducted in this section. In these experiments, a variety of different EADs are generated. For each EAD realization, measures introduced in section 3.4 are calculated. Correlation analyses are conducted to identify highly correlated measures that can be used as proxies for each other. If two measures show a high correlation, one can be dropped since the second measure does not provide much additional information. In the last step, the results gained from the experiment are discussed and mapped to existing findings from previous research.

3.6.1 Design of Experiments

Figure 3.16 shows the two-stage approach to analyze the interactions among measures. A preliminary analysis investigates the independent interaction of measures within DMMs and

DSMs using the sub-routines described in section 3.5.2. Conducting the preliminary has two advantages. By replicating the results of Hennig et al. (2022), systematic errors during measures implementation can be detected. In addition, another check of measures implementation is done by replicating studies that report both the input data and measures' value (see Table 3.23) [56]. This is done for the most complex measures such as structural complexity (SC), system design complexity ($JSDC, SDC$), and modularity (Q) since the implementation of count-based measures is straightforward by simply counting the elements. No available study reports data and the corresponding Neumann entropy value (NE). The study of Passerini and Severini (2009) was used to implement this measure. A second motivation for conducting the preliminary analysis is the independent creation of DSM and DMM patterns, representing individual design decisions. Product matrices, however, depend on the design defined by DSMs and DMMs, which makes it necessary to analyze such measures in combination with the DSM and DMM patterns.

[56] To be precise, replicating reported results does not guarantee correctness. It is a necessary but not sufficient condition.

Figure 3.16:
Two-stage approach for the analysis of measure interactions.

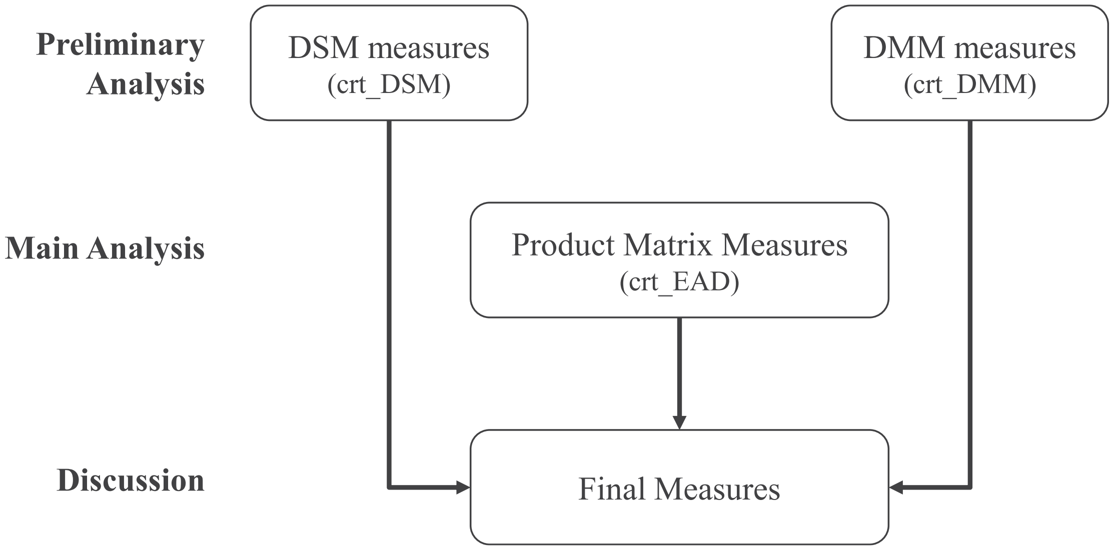


Table 3.23:
Replication results for topology measures

Matrix	Measure	Value	Authors
DMM	JSDC	$JSDC_{RDU1} = 389.5$ $JSDC_{RDU1} = 466.1$	Jung et al. (2022)
	SDC	two example matrices in Figure 2 $SDC_a = 1.39$ $SDC_b = 33.3$	Modrak and Bednar (2015)
DSM	Q	train boogie after decomposition $Q = 0.74$	Sinha and Suh (2018)
	SC	two example matrices in Figure 2 $SC_{F2,left} = 4.9$ $SC_{F2,left} = 6.83$	Sinha and Suh (2018)
	NE	-	-

The preliminary design of experiments is reported in Table 3.24, which is inspired by the work of Hennig et al. (2022), who vary key aspects of DSMs such as the size of the matrix (N_{el}), the number of entries (DNS_{DSM}) and the topology ($random - modular$). While their study creates 18 different architectures, this experiment creates 5,000 individual DSMs. The same principle is used for the creation of the DMM data set. The size is varied via the number of source (N_{src}) and target domain elements (N_{tgt}). The number of entries and the topology are varied by defining a pre-defined SDC_N value. Results of the preliminary analyses allow for dropping highly correlated measures.

Table 3.24:
Design of experiments for the supplemental analyses.

Analysis	Variable	Values		Range
		Low	High	
DSM	N_{el}	small DSM	large DSM	U[20;100]
	DNS	sparse DSM	dense DSM	U[0:0.14]
	modular	small weight on modularity	higher weight on modularity	U[0;1]
	Runs			5000
DMM	N_{src}	Low number of source elements	Large number of source elements	U[10;40]
	N_{tgt}	Low number of target elements	Large number of target elements	U[20;80]
	SDC_N	Uncoupled design	Coupled design	U[0;0.08]
	runs			5000

Note. U[lb;ub] indicates a uniform distribution within the lower bound (lb) and upper bound (ub)

Since product matrices depend on the underlying design matrices, the entire EAD is created for the main analysis. This allows for analyzing potential correlations between design and product view measures and correlations across domains. Table 3.25 reports the DOE for the main analysis.

Table 3.25:
Design of experiments for the main experiment to analyze the interactions among complexity measures across domains

Group	Variable	Description	low values	high values	Range[1)]
			Independent		
Product Mix & Demand	DNS_{FD}	density of the product feature matrix	products share only some features (product mix matrix is sparse)	products share many features among each other (product mix is dense)	U[0%; 50%]
	N_{PROD}	number of products	small product mix	large Product mix	U[50;100]
	TD	total number of sales for the created product mix	product mix with low number of total sales	product mix with high number of total sales	U[100; 10,000]
	Q_{var}	coefficient of variation for the demand vector	Sales are equally distributed across products	some products with a high share on sales and many products with low quantity	U[0;1.7]
Product Family Design	SDC_N	system design complexity, representing the normalized entropy in inter domain mapping matrices	uncoupled design	coupled design	U[0; 0.08]
	DNS_{DSM}	density of DSM matrices	no interactions within the individual DSM matrices	interactions within the individual DSM matrices	U[0%; 35%]
Product Family Design	N_{FR}	number of Elements in the functional domain	only a few functional requirements necessary to realize the product mix	many functional requirements necessary to realize the product mix	U_{int}[30;50]
	N_{DD}	number of Elements in the physical domain	only a few components necessary to realize the product mix	many components necessary to realize the product mix	U_{int}[30;50]
	N_{PrD}	number of Elements in the process domain	only a few processes necessary to realize the product mix	many processes necessary to realize the product mix	U_{int}[30;50]
	N_{RD}	number of Elements in the resource domain	only a few resources necessary to realize the product mix	many resources necessary requirements to realize the product mix	U_{int}[30;50]
			Dependent		
Simulation	RUNS	number of runs	only a few runs per parameter combination	many runs per parameter combination	5000
Simulation	seed	random seed to reproduce the results			1234

Note. 1) U[X;Y] representing a uniform distribution within the lower bound X and the upper bound Y. U_{int}[X;Y] the integer uniform distribution.

Since the product mix and demand are exogenous, they are created directly by the input parameter DNS_{FD} and Q_{var}. As reported in section 3.5.3, empirical boundaries are used for the relevant variables. For each EAD realization (rows in DOE), all applicable measures, as introduced in section 3.4, are calculated, resulting in two data sets. The data set 'DS1_system' contains the system level measures while 'DS1_product' contains twelve product level measures. Table 3.26 reports the summary statistics for the generated system-level data set on system-level ('DS1_system'). Since SDC_N, DNS_{DSM} and DNS_{FD} are directly created by the simulation routine, they match closely with the input parameter. The product matrix density in the remaining domains (DNS_{PD}, DNS_{PrD}, DNS_{RD}) are a result of the underlying product design and, therefore, is not directly specified by the user. Balakrishnan et al. (2011) note that low values of DNS_{RD} refer to a low degree of resource sharing, such as in job shops, whereas high values refer to shop floors, characterized by a high degree of resource sharing. Descriptive statistics indicate that the EAD generates only high-density environments, representing shop floor environments ($median_{DNS_{RD}} = 1$).

Table 3.26:
Summary statistics for the generated EADs in 'DS1_system'

Measure	Domain	Min.	25% quantile	Median	Mean	75% quantile	Max.	SD
DNS	FD	.152	.248	.310	.323	.398	.543	.092
	PD	.194	.593	.714	.694	.814	.994	.155
	PrD	.383	.957	.988	.962	.998	1.000	.065
	RD	.628	1.000	1.000	.999	1.000	1.000	.009
SDC_N	FD,PD	.002	.027	.047	.046	.065	.085	.023
	PD,PrD	.000	.021	.041	.041	.061	.081	.023
	PrD,RD	.000	.020	.040	.040	.060	.080	.023
DNS_{DSM}	PD	.000	.032	.072	.069	.104	.151	.040
	PrD	.000	.036	.072	.071	.106	.146	.041
	RD	.000	.036	.072	.073	.109	.149	.042

Cost accounting models (e.g., V. Anand et al., 2019) generate P_{RD}, defining a desired density exogenously. This allows generating a much wider range of possible resource consumption patterns compared to the EAD. Under the EAD model, matrix density results from multiple matrix multiplications, which increase the density continuously across domains ($DNS_{median,FD} = 31.0\% < DNS_{median,PD} = 71.4\% < DNS_{median,PrD} = 98.8\% < DNS_{median,RD} = 100\%$). The limitation of the EAD to high-density environments may be one caveat of the current model as it represents shop floor environments exclusively. A detailed comparison of both the ABL and EAD models is done in section 5.

3.6.2 Correlation Analyses

In the first part of this subsection, the results of the preliminary analyses are reported in Table 3.27 for the DSM and Table 3.29 for the DMM measures. Starting with the DSM experiment, the analysis reveals that structural complexity (SC) is mainly driven by the absolute number of entries (HIC, $c = .987, p < .05$) and, therefore, also by system size (N, $c = .762, p < .05$). This indicates that the amount of topological information transported by SC is low due to the high correlation with count-based measures. The same applies to the Neumann entropy (NE)

since SC and NE show an almost perfect correlation ($c = .993, p < .05$). An investigation of the normalized structural complexity (SC_N) indicates that this measure has a medium and negative correlation to the system size ($c = -.623, p < .05$), which indicates that the size normalization was not successful for this measure. The normalized Neumann entropy, however, is almost independent of system size ($c = .078, p < .05$) and strongly correlates with the count-based measures ($HIC_N, c = .991, p < .05$). As a last topological measure, the correlation of modularity (Q) is analyzed. Like the standardized count-based measures (MCC_N, HVM_N, HIC_N), modularity is already standardized and, thus, shows no significant correlation with system size (N). Modularity shows negative correlation coefficients to all other measures, which is caused by the mathematical definition and is insensitive to changes in system size ($c = .007, p > .05$). Higher values of SC and NE indicate that more information is needed to reconstruct a matrix, while increasing values of Q refer to a decomposed system. Therefore, increasing values refer to less complex designs. Sinha and Weck (2013a) argue that structural complexity and modularity are not necessarily negatively associated. Analysis results and theoretical discussions (e.g., Baldwin & Clark, 2000), however, indicate a negative association on average. Results of unnormalized count-based measures (MCC, HVM, HIC) show a high correlation among each other ($c \geq .987, p < .05$). The same applies to the normalized count measures (MCC_N, HVM_N, HIC_N), which show at least a correlation of $c \geq .960$ ($p < .05$).

Table 3.27:
Pearson correlation of DSM measures

Measure	N	MCC	HVM	HIC	MCC_N	HVM_N	HIC_N	SC	Q	NE	SC_N	NE_N
N	1**	.815**	.820**	.831**	.015	-.004	-.004	.762**	.007	.782**	-.623**	.078**
MCC		1**	.987**	.988**	.475**	.415**	.415**	.963**	-.314**	.985**	-.277**	.505**
HVM			1**	.999**	.434**	.409**	.409**	.990**	-.362**	.998**	-.274**	.499**
HIC				1**	.440**	.413**	.413**	.987**	-.366**	.996**	-.280**	.505**
MCC_N					1**	.960**	.961**	.453**	-.689**	.474**	.588**	.959**
HVM_N						1**	1.000**	.451**	-.825**	.453**	.671**	.992**
HIC_N							1**	.451**	-.828**	.452**	.670**	.991**
SC								1**	-.435**	.993**	-.217**	.534**
Q									1**	-.395**	-.567**	-.818**
NE										1**	-.232**	.539**
SC_N											1**	.579**
NE_N												1**

Note. ** indicates significance at p<0.05

The preliminary analysis shows similarities among groups of measures. Twelve measures are dropped based on the assumption that highly correlated measures hold only a small amount of additional information. The normalized structural complexity (SC_N) is dropped since size standardization was not successful. The remaining measures are assigned to two groups for which the correlation of measures within one group is high ($c \geq .7, p < .05$). For each group, the simplest measure is used since a simple measure is preferred over a more complicated one following the practice of good modeling (Robinson, 2008). The group assignment is presented in Table 3.28, where the sum of entries (HIC) is selected for the first and HIC_N for the second group. These two groups represent size-normalized (group 2) and non-size-normalized (group 1) measures. While Hennig et al. (2022) suggest HVM as a measure, it does not fulfill all construct validity criteria. The null value condition for interrelatedness (C4) states that a system without any relations ($E = 0$) has no complexity. However, a proof of condition C4 indicates that HVM does not fulfill this condition:

$$HVM(DSM_{bin}) = (N + 0) * log_2(N + 0) > 0, DSM_{bin} \in \mathbb{N}_{0,1}{}^{N\,x\,N} \wedge N \geq 1 \quad (3.68)$$

Another advantage of using HIC and HIC_N as DSM measures is their simplicity. HIC avoids a logarithmic scaling, and HIC_N is equivalent to DSMs' density. A disadvantage of HIC_N is that it does not reflect the topological properties of the DSM. However, additional topological information provided by other measures, such as modularity, is small since a high correlation is observed ($c_{Q,HICN} = -.828, p < .05$). Even though modularity (Q) provides the most additional topological information, it requires a priori group assignment increasing its complexity and reducing its reproducibility.

Table 3.28:
Groups of highly correlating measures. 'S' indicates the selected measures and 'D' measures which are dropped from further analyses.

Group	SC	Q	NE	MCC$_N$	HVM$_N$	HIC$_N$	MCC	HVM	HIC	N	SC$_N$	NE$_N$
1	x		x				x	x	S	x		
2		x		x	x	S					D	x

The same approach is used to identify similarities among DMM measures. Table 3.29 shows a high correlation between SDC and $JSDC$ $(c = .987, p < .05)$, indicating a high similarity. Both measures describe matrices' topology and depend on system size as they show a strong correlation between the number of source and target elements matrix $(\min(c_{Nsrc}, c_{Ntgt}) > .672, p < .05)$. Results further indicate that the standardization of the system design complexity was successful since there is no correlation between SDC_N and the system size $(\max(|c_{Nsrc}|, |c_{tgt}|) < .002, p > .05)$. This work recommends SDC_N as a DMM measure since the number of source (N_{src}) and target elements (N_{tgt}) are already measured by HIC. Standardized measures allow for describing the interrelated complexity independently from the system size.

Table 3.29:
Pearson correlation for DMM measures.

Measure	SDC	SDC$_N$	JSDC	N$_{src}$	N$_{tgt}$
SDC	1**	.546**	.987**	.672**	.690**
SDC$_N$		1**	.500**	-.002	-.002
JSDC			1**	.654**	.738**
N$_{src}$				1**	.861**
N$_{tgt}$					1**

Note. ** indicates significance at p<0.05

For the main analysis, the number of connections (HIC, HIC_N) are selected for intra-domain measures and SDC as well as SDC_N for the inter-domain measures. The remaining product measures are calculated for each of the 5,000 individual EADs, leading to two data sets. The data set *'DS1_system'* contains the system level measures while *'DS1_product'* contains the twelve product level measures (three measures in four domains). For each data set, two correlation analyses are performed. The first correlation analysis calculates Pearson correlation coefficients, which indicate the strength of the linear relationship between two variables. In contrast, the second analysis calculates the Spearman rank correlation, measuring the non-linear relationship between measures. Only strong correlations are of interest in this study. Therefore, coefficients below $c \leq .5$ are not reported. Nevertheless, measures are only substituted if they show at least a high correlation ($c \geq .7$) to another proxy. Correlation results for system-level measures are reported in Table 3.30. Table 3.31 reports the correlation among product-level measures. All coefficients reported in this section are significant at $p < 0.05$. Non-significant values are not reported for better readability.

Table 3.30:
Correlation coefficients for each pairwise measure interaction. The upper triangle matrix represents Spearman rank correlation while the lower the Pearson correlation.

	Product & Demand View																		Design View						
	-	FD							PD				PrD			RD			FD	PD		PrD		RD	
Measure	**DMD_{T10}**	**D**	**N**	**NPV**	**CI**	**PCI**	**DNS**	**OV**	**D**	**CI**	**PCI**	**DNS**	**D**	**PCI**	**DNS**	**D**	**PCI**	**DNS**	**N**	**HIC_N**	**SDC_N**	**HIC_N**	**SDC_N**	**HIC_N**	**SDC_N**
$DMD_{T10\%}$	1	-.558																							
D_{FD}	-.622	1			.504	-.712	-.712	-.706	.677				.526			.522									
N_{PROD}			1		-.589																				
NPV_{FD}				1															-.980						
CI_{FD}			-.586		1	-.824	-.824	-.836		.613															
PCI_{FD}		-.535			-.774	1	1.000	.997	-.536		.570	.570													
DNS_{FD}		-.535			-.774	1.000	1	.997	-.536		.570	.570													
OV_{FD}					-.829	.954	.954	1	-.530		.567	.567													
D_{PD}		.743				-.501	-.501		1	.750	-.862	-.862	.746	-.508	-.508	.603				-.553					
CI_{PD}					.593				.668	1	-.903	-.903	.542	-.543	-.543					-.640					
PCI_{PD}						.564	.564	.550	-.805	-.875	1	1	-.624	.589	.589					.686					
DNS_{PD}						.564	.564	.550	-.805	-.875	1	1	-.624	.589	.589					.686					
D_{PrD}		.605							.786	.525	-.610	-.610	1	-.839	-.839	.776									
PCI_{PrD}										-.513	.515	.515	-.735	1	1	-.574						.640			
DNS_{PrD}										-.513	.515	.515	-.735	1	1	-.574						.640			
D_{RD}		.645							.673				.810			1	-.514	-.514							
PCI_{RD}																	1	1							
DNS_{RD}																	1	1							
N_{FD}				-.892															1						
$HIC_{N,PD}$									-.528	-.606	.699	.699								1					
$SDC_{N,FD,PD}$																					1				
$HIC_{N,PrD}$														.508	.508							1			
$SDC_{N,PD,PrD}$																							1		
$HIC_{N,RD}$																								1	
$SDC_{N,PrD,RD}$																									1

Note. 1) EAD domain with FD=functional domain, PD=physical domain, PrD=process domain, RD=resource domain; All shown coefficients are significant at $p<0.05$ and above a threshold of 0.5.

Table 3.31:
Correlation coefficients for each pairwise measure interaction on product level. The upper triangle matrix represents Spearman rank correlation while the lower the Pearson correlation. All shown coefficients are significant at p<0.05 and above a threshold of 0.5.

		FD			PD			PrD			RD		
Domain	**Measure**	**$INTER_{FD}$**	**$INTRA_{FD}$**	**$LOF_{10\%,FD}$**	**$INTER_{PD}$**	**$INTRA_{PD}$**	**$LOF_{10\%,PD}$**	**$INTER_{PrD}$**	**$INTRA_{,PrD}$**	**$LOF_{10\%,PrD}$**	**$INTER_{RD}$**	**$INTRA_{RD}$**	**$LOF_{10\%,RD}$**
	$INTER_{FD}$	1.000			.628								
FD	$INTRA_{FD}$		1.000			.702							
	$LOF_{10\%,FD}$			1.000									
	$INTER_{PD}$	.628			1.000			.567					
PD	$INTRA_{PD}$		.628			1.000			.530				
	$LOF_{10\%,PD}$						1.000			.614			
	$INTER_{PrD}$				.566			1.000			.823		
PrD	$INTRA_{,PrD}$					.530			1.000				
	$LOF_{10\%,PrD}$						.614			1.000	.608		.840
	$INTER_{RD}$							.822		.608	1.000		.642
RD	$INTRA_{RD}$											1.000	
	$LOF_{10\%,RD}$									.840	.642		1.000

Table 3.30 reports the pairwise correlation coefficients for the system-level measures in EAD's product view. The Diversification indices show a strong correlation across domains. For example, D_{FD} shows a strong, non-linear, and significant correlation to diversification in the physical domain ($c_{spear,DFD,DPD} = .743, p < .05$), the process domain ($c_{spear,DFD,DPrD} = .605, p < .05$), and the resource domain ($c_{spear,DFD,DRD} = .645, p < .05$). It indicates that diversification in the functional product mix also affects the diversification in other domains. Thus, a diverse product mix has a higher chance of leading to diverse resource consumption patterns. As further expected, the Diversification index (D_{FD}) and demand heterogeneity ($DMD_{T10\%}$) are negatively associated ($c_{pear,DMDT10\%,DFD} = -.558, p < .05$). This aligns with findings by Gollop (1997), who states that demand homogeneity (low values of $DMD_{T10\%}$) increases the Diversification index. Perfect diversification $D \rightarrow 1$ is reached when products do not share any elements among each other, and there are infinitely many products for which $DMD_j = 1, for\ j = 1 \dots N_{PROD}$. No diversification ($D = 0$) is reached under single-product firms. Another association is observed between the homogeneity among products and the Diversification index. Results show that commonality shows a string negative associated with D ($c_{pear,PCIFD,DFD} = -.712, p < .05$). Since higher values of commonality indicate more homogeneity, products share more elements among each other and Diversification increases. These findings underline the composite character of the Diversification index. A problem with composite measures is their difficulty in interpretation, as Trattner et al. (2019) note.

As argued, $PCI = DNS$ is valid for sufficiently dense matrices leading to a perfect correlation between both measures in each domain ($c = 1, p < .05$). This allows to use PCI and DNS as proxies. Since commonality is more prominent in engineering, PCI is used for the functional and physical domains, and DNS for the process and resource domains. Results further show that product matrix measures show a high correlation with each other. Option variability (OV_{FD}), for example, correlates strongly with the commonality index ($c_{pear,OVFD,CIFD} = -.836, p < .05$), commonality ($c_{pear,OVFD,PCIFD} = .997, p < .05$), and the Diversification index ($c_{pear,OVFD,DFD} = -.706, p < .05$). The same pattern repeats in the physical domain, where CI_{PD}, PCI_{PD} and DNS_{PD} show strong correlations with each other. Finally, there are also medium correlations across domains. For example, PCI_{FD} and PCI_{PD}show a medium correlation ($c_{pear,PCIFD,PCIPD} = .570$) as well as PCI_{PD} and PCI_{PrD} ($c_{spear,PCIPD,PCIPrD} = 0.589$). This indicates that the commonality is inherited across domains to some degree. For example, a high degree of component commonality leads to a certain degree of process commonality, as the correlation coefficients suggest. The lower right part of Table 3.30 reports the correlation for the pre-selected inter- and intra-domain measures. As expected, there are no correlations among design view measures as these matrices are created independently. However, a medium correlation between $HIC_{N,PrD}$ and product matrix measures ($PCI_{PD}, CI_{PD}, D_{PD}, DNS_{PD}$) is observed ($|c| \geq .553, p < .05$). This indicates that component commonality (homogeneity) is primarily driven by intra-domain matrix density. The same applies to the process domain.

Correlation coefficients among product level measures are reported in Table 3.31. Only two measures show high correlations with each other. Inter-product heterogeneity in the process and resource domain are significant and positively associated ($c_{pear,INTER_{PrD},INTER_{RD}} =$

$.823, p < .05$) as well as the local outlier factor ($c_{pear,LOF_{PrD},LOF_{RD}} = .840, p < .05$). As for product matrix measures on the system level, product-level measures show a medium correlation across domains. Inter-product heterogeneity, for example, shows a medium correlation between the functional and physical domains ($c_{pear,INTER_{FD},INTER_{PD}} = .628, p < .05$). The low correlation between M. Gupta's (1993) product measures ($INTER$, $INTRA$) and the local outlier factor ($LOF_{10\%}$) highlights the need to keep $LOF_{10\%}$ as a measure in future studies.

3.6.3 Selection of Measures

Based on correlation results, highly correlated measures are dropped as they provide little additional information. In doing so, highly correlated measures are clustered. For each cluster, a single measure is selected based on the following criteria:

- **Construct validity:** The selected measure must fulfill the criteria for construct validity (only multiplicity and interrelatedness measures).
- **Simplicity:** The selected measure should be simple.
- **Standardization:** Standardized and non-standardized measures should be available.
- **Completeness:** The set of overall selected measures must account for the three dimensions of static complexity in a systemic context (content validity).

While the first and last conditions are must-have properties as they ensure the content and construct validity, criteria two and three are desirable (should-have) properties. Construct validity criteria are introduced in section 3.3.2 and checked for each selected measure individually. As noted, this check is only possible for measures related to multiplicity or interrelatedness. While the first three criteria are checked for each measure individually, the completeness criterion is checked for the entire set of selected measures. To ensure content validity, the suggested and non-standardized set of measures should account for each dimension of complexity (multiplicity, interrelatedness, and diversity). The simplicity criteria refer to good modeling practice (see Robinson, 2008), where a simple explanation is preferred over a difficult one. For example, Trattner et al. (2019) note that composite measures such as the Diversification index aggregate much information in one value, which makes it difficult to explain. Another desirable property is the standardization of the system size within a zero to one range. If a measure can be standardized, it is preferred over another. Size-standardized measures allow for reporting systems relative complexity. While different-sized systems can have different absolute complexity (measured via non-standardized measures), they can have the same relative complexity (measured via standardized measures) related to their size. It allows comparing different-sized systems in terms of interrelatedness and diversity with each other. Some measures, such as the PCI, density (DNS) or modularity (Q) are already standardized. For other measures, a standardized version is introduced in section 3.4.2, such as HIC_N, HVM_N, and SDC_N.

Table 3.32 reports the set of selected system-level measures consisting of 17 individual measures: seven in EAD's product and demand view and ten in its design view. Each group contains measures showing at least a high correlation ($|c| > .7$) with its proxy. All measures except for modularity (Q) are included as it is already standardized and shows a high correlation to HIC_N.

Table 3.32:
Suggested set of complexity measures on system level.

View	Measure	Proxy for[1)]	Standardized	Construct Validity	Dimension[2)] M	I	D
Product & Demand	$DMD_{T10\%}$	-	already	NA			x
	N_{PROD}	-	NPV	x	x		
	PCI_{FD}	DNS_{FD}***, OV_{FD}***, CI_{FD}***, D_{FD}*	already	NA			x
	PCI_{PD}	DNS_{PD}***, CI_{PD}*, D_{PD}**	already	NA			x
	DNS_{PrD}	PCI_{PrD}***; D_{PrD}**	already	NA			x
	DNS_{RD}	PCI_{RD}***	already	NA			x
	D_{RD}	-	already	NA			x
Design	N_{FD}	-	-	x	x		
	N_{PD}	-	-	x	x		
	N_{PrD}	-	-	x	x		
	N_{RD}	-	-	x	x		
	HIC_{PD}	SC_{PD}***, NE_{PD}***, MCC_{PD}***, HVM_{PD}***	$HIC_{N,PD}$	x		x	
	$SDC_{FD,PD}$	$JSDC_{FD,PD}$***	$SDC_{N,FD,PD}$	x		x	
	HIC_{PrD}	SC_{PrD}***, NE_{PrD}***, MCC_{PrD}***, HVM_{PrD}***	$HIC_{N,PrD}$	x		x	
	$SDC_{PD,PrD}$	$JSDC_{PD,PrD}$***	$SDC_{N,PD,PrD}$	x		x	
	HIC_{RD}	SC_{RD}***, NE_{RD}***, $MCCR_{D}$***, HVM_{RD}***	$HIC_{N,RD}$	x		x	
	$SDC_{PrD,RD}$	$JSDC_{PrD,RD}$***	$SDC_{N,PrD,RD}$	x		x	

Note. [1)] Measure and proxy have an absolute Pearson or Spearman correlation of >|0.7| (*), >|0.8| (**) or >|0.9| (***); [2)] M=multiplicity, I=interrelatedness, D=diversity

Demand heterogeneity is the first measure included in the final set. It describes demand vectors' diversity, whereas low values indicate a homogenous demand distribution in which each product has the same share of sales. High values indicate a skewed demand distribution where some products are responsible for a large proportion of total sales. $DMD_{T10\%}$ is not perfectly standardized within a range of zero to one since the minimum value is $\min(DMD_{T10\%}) = 0.1$ and refers to the dimension of diversity. Therefore, the construct validity criterion is not applicable. Product line commonality in the functional and physical domain (PCI_{FD}, PCI_{PD}) as well as the degree of process/resource sharing (DNS_{PrD}, DNS_{RD}) describe diversity in EAD's product view. Low values indicate a small degree of element sharing, while high values indicate a large degree. Thus, large values of PCI or DNS refer to more homogeneous product matrices (P), while low values refer to more heterogeneous patterns. These measures are already standardized, making them a good choice for the final set. A last diversity-related measure is the Diversification index in the resource domain (D_{RD}). While other measures are good proxies for diversification in other domains, only a medium correlation between D_{RD} and DNS_{RD} observed. The reason for such differences is that DNS_{RD} only differs between zero and non-zero elements, while D_{RD} also accounts for the differences in resource usage. This effect becomes only relevant in the resource domain as the resource consumption pattern (P_{RD}) shows higher differences in element usage compared to the remaining domains[57]. Therefore, it is necessary to keep D_{RD} in the final set, although Trattner et al. (2019) highlight the difficulties of composite measures. The Diversification index increases with increasing demand homogeneity and decreases with increasing product homogeneity as it is negatively associated with both $DMD_{T10\%}$ and PCI.

---

[57] Diversity of element usage increases with each domain due to the matrix multiplication.

Multiplicity is operationalized via the number of products (N_{PROD}) and the domain size ($N_{FD}, N_{PD}, N_{PrD}, N_{RD}$). In contrast to the number of domain elements, N_{PROD} does not count system elements directly. Since products are a combination of domain elements, this measure and its standardized counterpart (NPV) refer more to the combinatorial multiplicity rather than the system element multiplicity. Nevertheless, construct validity criteria C1-C3 are checked for this measure. Non-negativity (C1) is given since products are defined as a set of domain elements. Therefore, $N_{PROD} \geq 0$ applies to all valid EAD systems. The null value condition (C2) is satisfied since the power set of an empty set ($N_{FD} = 0$) results in zero possible combinations $N_{PROD} = 0$. While systems elements ($N_{FD} > 0$) are a necessary condition for $N_{PROD} > 0$, they are not sufficient since in- and cross-tree relationships can forbid the combination of any product. Additivity (C3) is fulfilled since the total number of products equals the sum of the products for each individual product mix ($PM\ I, PM\ II$), assuming that both product mixes do not share any elements among each other. ($N_{FR,PM\ I} \cap N_{FR,PM\ II} = \emptyset$). While the number of base elements (N_{FD}) restricts the combinatorial multiplicity (N_{PROD}, NPV), the system size can reach any value of $N \geq 0$ for all domains and, therefore, cannot be standardized. Although the number of domain elements strongly correlated with intra-domain measures such as HIC, HVM, MCC and SC, the domain size measures are stated as individual variables. It allows for measuring the interrelatedness independent from the system size if the standardized inter- and intra-domain measures are used.

Interrelatedness within domains is operationalized by the interface complexity (HIC), which counts the number of entries within a DSM. Small values indicate sparse DSMs, whereas high values indicate dense intra-domain matrices. Although Hennig et al. (2022) recommend HVM, it does not fulfill the criteria for construct validity (C4 is violated). While topological measures (Q, SC, NE) also account for the distribution of entries, they show a high correlation with the much simpler HIC or HIC_N measures. This indicates that the amount of topological information transported by these measures is small. Another reason against Q, SC and NE is their high complexity. Interface complexity refers to the interrelatedness as it counts the number of elements relations. Therefore, the fulfillment of construct validities' criteria C1 and C4-C6 is proven in Appendix A2.

The system design complexity (SDC) is selected to operationalize interrelatedness between domains. Based on the Shannon entropy, this measure uses a domain mapping matrix and measures the degree of coupling. Low values indicate an uncoupled design. High values indicate a more coupled design. Compared to the design complexity suggested by Jung et al. (2022), SDC is simpler in both interpretation and computation. Jung's design complexity relies on the graph energy, for which Hennig et al. (2022) identified an inconsistent behavior. Another reason for choosing SDC is the existence of a size-standardized measure (SDC_N). The system design complexity fulfills all criteria for construct validity, whereas Appendix A3 shows the proof.

The set of 17 individual system-level measures addresses all three dimensions of complexity. The number of domain elements and products covers multiplicity. Diversity exists only on the product level since domain elements are the most granular layer. Diversity is measured via the product line commonality index in the functional (PCI_{FD}) and physical domains (PCI_{PD}).

In the process and resource domains, diversity is measured by the degree of element sharing (DNS_{PrD}, DNS_{RD}). Interrelatedness between domains is operationalized via SDC (SDC_N) and within a domain via HIC (HIC_N). As Table 3.32 shows, the suggested set represents all dimensions of static complexity in a systemic context and, therefore, shows content validity.

On the product level, only two measures show high pairwise correlations. Therefore, only $LOF_{10\%,PrD}$ and $INTER_{PrD}$ can be dropped from a strictly analytical perspective. From a practical perspective, however, it is only necessary to keep some measures since the discussion on product-level measures primarily focuses on the functional and resource domain in the literature. Product management, for example, defines product level diversity in terms of exoticness based on differences across their functional requirements (e.g., Rebentisch, Schuh, Riesener, Gerlach, & Zeller, 2016; Schuh, 2005; Schuh, Riesener, & Rudolf, 2014). Cost accounting defines product-level heterogeneity via products' resource consumption (e.g., M. Gupta, 1993; Horngren et al., 2012; Mertens, 2020). Although product heterogeneity is also discussed in the physical and process domain, their discussion is limited to the system-level view (e.g., Kota et al., 2000). Therefore, in the further proceeding of this work, only the functional domain and resource domain product-level measures are used[58], resulting in six product-level measures, as reported in Table 3.33.

Table 3.33:
Product level measures used in further steps of this work.

Domain	Measure	meaning of values: low	meaning of values: high
FD	$INTER_{FD}$	product's features reflect the average product in the functional product mix	product is placed at the boundaries of the functional product mix
	$INTRA_{FD}$	low variation across features	high variation across features
	$LOF_{10\%,PD}$	-	product with an exotic combination of functional requirements
RD	$INTER_{RD}$	product's resource consumption reflects the average product based on it resource consumption	product's resource consumption is different than those from an average product
	$INTRA_{RD}$	low production complexity	high production complexity
	$LOF_{10\%,RD}$	-	product's resource consumption is exotic

Low values of $INTER$ indicate that a product is placed in the center of the product mix, whereas high values indicate a distance from its center (Mertens, 2020). As shown in section 3.4.3, a product placed at the border of the product is neither a sufficient nor a necessary condition for the product to be exotic ($LOF_{10\%,} \gg 1$). Thus, low values of $INTER$ do not imply that products are non-exotic. Low values of $LOF_{10\%,} < 1$ indicate that the product lies in a group of similar products, which has no meaningful interpretation in the context of this work. Intra-product heterogeneity is associated with the production complexity (Mertens, 2020). Low values indicate that the product has no variations in resource consumption or its feature vector, whereas large values indicate differences.

[58] Another argument for dropping product-level measures in the physical and process domain is their medium correlation to their functional/resource domain counterparts.

3.6.4 Discussion

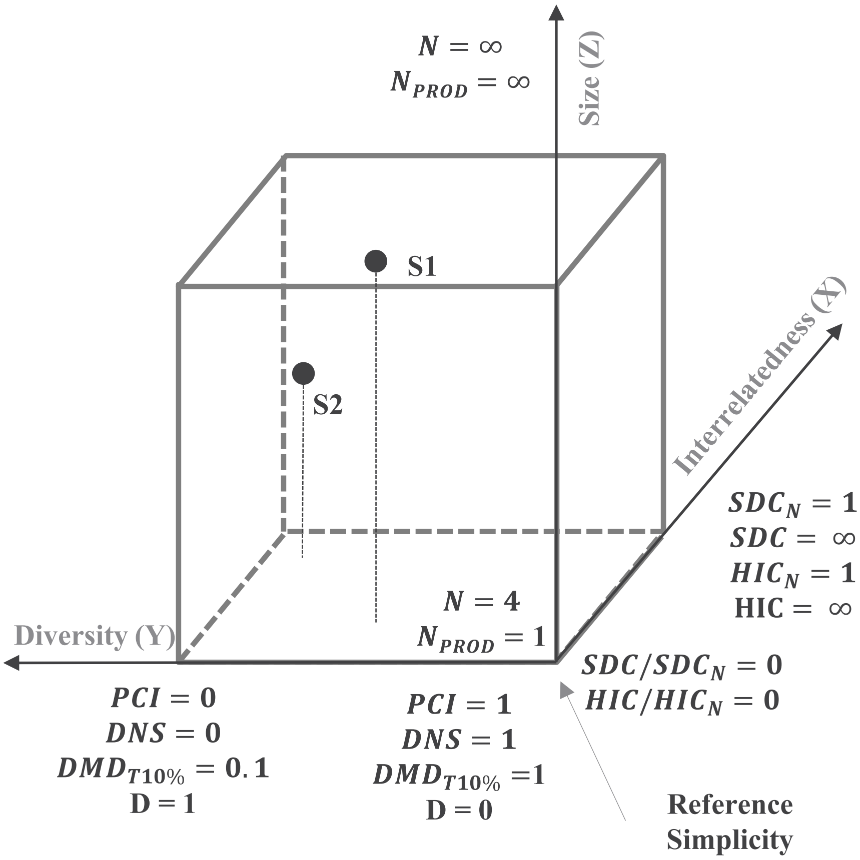
The literature review identified 30 unique PFCMs in the context of product family design. A two-stage approach is performed to suggest a reduced set of PFCMs. In the first step, 5000 unique DMMs and DSMs are created, representing empirically related patterns such as modular and random designs in DSMs and coupled-uncoupled designs in DMMs. It is observed that empirically observed inter- (DMMs) and intra-domain matrices (DSMs) are sparse. Corresponding measures are calculated for those matrices. Performing multiple correlation analyses identifies similarities among measures. The analyses reveal that several measures are highly correlated (c≥ .7) and can be substituted by proxies since the additional information provided by those measures is low. Similar results were found during the main analysis for the product view measures. For example, the commonality index (CI), the product line commonality index (PCI), the matrix density (DNS), and the option variability (OV) all show high correlations with each other. The only exception is the Diversification index, a composite measure reflecting diversity in the product matrix and the demand vector. Except for the resource domain, the Diversification index highly correlates with the remaining product matrix measures. Different results are found for the product-level measures ($INTER$, $INTRA$, LOF), which cannot be substituted by each other. However, they show a medium correlation across domains (e.g., $INTER_{PrD}$ and $INTER_{RD}$). Based on correlation results, measures are reduced based on multiple criteria (construct validity, simplicity, content validity, standardization). Finally, this work suggests a set of six product-level and 17 validated system-level measures to describe internal complexity within the EAD.

Related to the issues of PFCM's variety, this work follows the call for measure consolidation (Hennig et al., 2022; Park & Okudan Kremer, 2015). Results indicate a high correlation among groups of measures. This finding is especially relevant for topological DSM measures such as structural complexity (SC), modularity (Q), or Neumann entropy (NE). In theory, these measures can differ between centralized and random designs. Correlation analysis, however, shows that topological measures are primarily driven by the number of entries rather than the distribution of entries under empirical DSM patterns. In sum, findings in this section are symptomatic of the *"fragmented and divergent state"* (Hennig et al., 2022, p. 13) in PFCM literature, raising two questions. First, whether such complex measures are necessary since additional information on matrices topology content is small. A second question is whether future studies should introduce new PFCMs or, instead, focus on consolidating existing measures.

Literature raises concerns regarding PFCM's validity (Blecker & Abdelkafi, 2006; Jung et al., 2022; Sinha & Suh, 2018) and the large variety of existing validation approaches (Hennig et al., 2022). This work suggests a procedure for checking PFCMs' content and construct validity. Content validity ensures measures' sensitivity to changes in at least one of complexity's dimensions (multiplicity, interrelatedness, diversity). Figure 3.17 provides a graphical representation of measures' content validity as it assigns each measure to the corresponding axis in a three-dimensional complexity space. While the z-axis represents the dimension of multiplicity, the x-axis represents the aspect of interrelatedness, and the y-axis the diversity. All measures assigned to the x- and y-axis have standardized counterparts (indicated with an index N). Using these standardized values allows for describing the relative complexity of two systems in the xy-plane. Two systems (S1 and S2) are exemplarily placed inside the cube. While system S2 is medium-sized with almost no commonality and many connections, S1 represents a large system with a medium degree of commonality and interrelatedness. From a perspective of absolute complexity, it is difficult to answer whether S1 or S2 is more complex. However, from a perspective of relative complexity, it becomes clear that S2 is more complex as it has more interrelatedness and diversity compared to its size. As shown in Figure 3.17, the complexity cube further integrates the reference simplicity placed at the origin. Remember, the (objective) reference simplicity represents the simplest system possible, defined as consisting of four elements ($N = 4$)[59], no interrelations ($HIC = 0, SDC = 0$), and no diversity ($PCI = DNS = 1, D = 0$). Since such a system has only one element per domain and each product must be distinct, there is inevitably only one product possible ($N_{PROD} = 1$). Regarding the construct validity, this work follows other related studies (Hennig et al., 2022; Jung et al., 2022; Sinha & Suh, 2018) which use the criteria by Weyuker (1988) and Briand et al. (1996). Although construct validity was only fully checked for the final set of measures, it is shown that some existing PFCMs violate certain conditions for construct validity.

[59] One element in each domain ($N_{FD} + N_{PD} + N_{PrD} + N_{RD}$).

Figure 3.17:
Complexity Cube with the selected set of measures.



3.7 Conclusion

This section operationalizes internal complexity using the EAD simulation model. In doing so, the EAD is extended by intra-domain matrices (DSMs) and fixed costs. EAD's ability to support multi-attribute decision-making problems is demonstrated via a case study. In line with analytical studies (e.g., Thonemann & Brandeau, 2000), a U-shaped curve of costs under varying levels of component commonality is identified. This indicates that there is an optimal degree of component commonality. While increasing commonality utilizes the potential of fixed cost regression, too much commonality results in higher variable costs due to components' over-design. In the next step, the conceptual EAD is transferred into a numerical model, which allows for generating a variety of product family designs. The individual realizations of the EAD represent individual product families with a certain level of internal complexity. The model enables researchers to address a bunch of questions discussed in literature through numerical experiments such as the effects of product family complexity on costs (e.g., Hackl et al., 2020; Labro, 2004; Trattner et al., 2019) or product costing system accuracy (e.g., Labro & Vanhoucke, 2007; Mertens, 2020). Product family complexity measures (PFCMs) are needed to answer these questions on an operational level as they represent the independent variables. Literature, however, notes challenges in the context of product family complexity due to the variety of existing measures and raises concerns regarding their validity (Hennig et al., 2022; Jung et al., 2022; Trattner et al., 2019). 30 PFCMs reported in the literature are identified, and 5000 unique EADs are generated using the numerical model. By performing correlation analyses, highly correlated measures are substituted by proxies. For the reduced set of PFCMs, content and construct validity are checked to address the concerns regarding the validity of measures.

This work has several implications for researchers and practitioners (see Figure 3.18). From the researcher's perspective, this study underlines the *"fragmented and divergent state"* (Hennig et al., 2022, p. 13) in PFCM literature as it identifies a high similarity among many measures. Specifically, this study shows that complex topological measures are mainly driven by the number of interrelations rather than their distribution in DSMs representing empirical patterns. It indicates a lack of measures that are less sensitive to the number relations and focus more on the design patterns of a DSM. For example, topological measures are mainly driven by matrices' density rather than its structure. Since many highly correlated measures exist, another implication is made regarding introducing new measures. Specifically, this work calls for a more substantial reflection of existing PFCM literature before introducing new measures. It will prevent a further increase of measures that do not provide new information about the static system state. A third implication is related to the EAD model itself. Integrating DMMs and DSMs into the EAD allows the representation of many product family designs, as reported in recent case studies (e.g., G. Kim et al., 2016; Sawai et al., 2017). Future research can use the EAD notation for a more standardized reporting of product family designs. Product family designs reported in the literature are included in the numerical EAD framework. A more standardized representation of product family designs would allow future research to replicate results and enable cumulative research.

Figure 3.18:
Section's implications for research and practice.

Research

- Search for topological DSM measures that are less sensitive to matrices' density.
- Before introducing new measures, make sure that these measures significantly differ from existing ones.
- Use the EAD for a standardized reporting of product family designs. It will ease replication and cumulative research.

Practice

- The EAD allows for creating design alternatives and evaluating their economic consequences systematically.
- Prefer simple complexity measures over more complex ones since the additional information provided by complex measures is low.

The findings of the sections also have implications for practice. In the case study, 64 different unique product family designs were systematically created. These design alternatives represent the design space, mapped onto a solution space consisting of the dimension's costs and commonality. This example highlights EAD's ability to use it as an optimization tool. It allows for a systematic creation of product family designs and the evaluation of their economic consequences. The study by Sinha and Suh (2018) shows the importance of systematic system design optimization. However, their study is limited to a single DSM and does not evaluate design alternatives' economic consequences. A second implication is made regarding PFCMs. Several studies discuss the effects of PFCMs on costs and operational performance (e.g., Trattner et al., 2019). Investigating these relations is highly relevant for firms since product design decisions are made long before effects on costs and operational performance become visible. Firms must evaluate design alternatives with little knowledge of later economic effects (Fixson, 2006; Skirde et al., 2016). PFCMs as proxies for economic consequences address this dilemma and allow a more objective comparison of alternative designs. Related to the correlation analyses, this study claims that firms should use simple measures instead of complex topological measures for two reasons. First, additional system information reported by complex measures is small compared to simple measures. However, complex measures require more estimation efforts. A second reason is the explainability and understandability of complex measures in practice. If system designers do not fully understand the influences driving a certain PFCM, they will likely refuse them to use. As noted in the literature (Trattner et al., 2019) and demonstrated in this section, the Diversification index is an example of a complex measure driven by several factors.

This section has some limitations, which are briefly discussed. Interrelatedness in DSMs is measured via the density (HIC), which cannot reflect topological properties. Therefore, this work cannot analyze differences in total costs or product costing system accuracy between random or modular designs. Measures' similarities under dense matrices are not investigated as they do not or only rarely appear in practice. However, the empirical data set is a tiny snapshot of existing product family designs. A systematic research of product family designs can tackle this limitation. Another limitation is that the resulting set comprises 17 system-level and six product-level measures. While this highlights the complexity in operationalizing complexity, it may not be ideal reporting results.

Based on these findings several open topics for future research are identified. First, topological measures are sensitive to matrices' density. Future research should search for topological measures that are less sensitive to such changes. This is important as it allows for comparing the economic effects of different DSM patterns. A second direction for future research addresses the issue of measure's variety. Future studies introducing new PFCMs should raise the self-critical question, to what extent their measure differs from existing ones. A last direction for future research is related to the issue of measures' validity. First, there is still no standardized approach to check the construct validity of PFCMs. Second, validation criteria are limited to the dimension of multiplicity and interrelatedness. Currently, no criteria exist to prove the construct validity regarding the diversity dimension.

4 Effects on Total Costs

4.1 Introduction

This section investigates the impact of internal complexity on costs. According to the impact model (see section 2.6.2), several cause-effect relationships have been mentioned in literature over the recent decades. While these studies are either rather general and conceptual by stating many qualitative relationships (e.g., Fixson, 2005; Perera et al., 1999) or focusing on one specific aspect in isolation (e.g., Eynan & Rosenblatt, 1996; Gavirneni, 2002; Thonemann & Brandeau, 2000), there exist only some studies in between (Hackl et al., 2020; Lyons et al., 2020; Santos et al., 2020; Stäblein et al., 2011; Trattner et al., 2019). Since those studies were published mainly over recent years, research and practice are increasingly interested in analyzing the economic consequences of internal product family complexity. Nevertheless, these in-between studies still note an unclear picture of costs induced by internal complexity. Table 4.1 shows statements on future research made by three studies. Those studies raise the need for a more in-depth investigation of internal complexity's economic consequences and call for operationalizing the conceptual relationships.

Table 4.1:
Statements on future research of related studies.

Author(s)	Journal	Study	Future Research
Lyons et al. (2020)	International Journal of Production Economics	empirical study among U.K. manufacturing firms (N=162), analyzing the economic consequences of product variety	*"Understanding the effects of variety mitigating capabilities on business activity and business process performance is a necessary, future line of enquiry."* (p.11)
Hackl et al. (2020)	Journal of Mechanical Design	introducing the impact model based on a literature review and case studies	*"Future work should include quantifying the links in the network model. These equations would need to provide estimates of how the output node would change as a function of the input nodes. Many equations are therefore required to quantify this model."* (p.10)
Trattner et al. (2019)	CIRP Journal of Manufacturing Science and Technology	literature review investigating the effect of internal complexity on operational performance such as costs, time, quality	*"Future research areas include investigating the relationships between product complexity and operational cost […] to understand when the relationships are linear and when they are logarithmic or quadratic."* (p. 80)

This section systematically investigates the economic consequences of internal complexity. In doing so, complexity-induced cost effects, reported by Hackl et al.'s (2020) impact model, are operationalized. Specifically, eight models estimating complexity-induced costs[60] are adapted from the literature. In the second step, these models are integrated into the numerical EAD framework, and a large-scale numerical experiment is conducted. The experiment creates 5000 individual product family designs. The degree of component overdesign is increased for each product family design, and the effects on complexity-induced cost are analyzed. Analyses are separated into two parts. First, in-depth analyses are made for each cost effect using correlation analyses and structural equation models. In the second step, the impact on total complexity-induced costs is investigated. Finally, these results are related to the existing stream of literature, and implications for both practice and research are made.

This study is positioned between the three fields of existing literature, characterized by conceptual studies (e.g., Perera et al., 1999; Hackl et al., 2020), empirical studies (e.g., Lyons et al., 2020; Ripperda & Krause, 2017), and analytical studies (e.g., Thonemann & Brandeau, 2000; L. L. Zhang et al., 2020; Hillier, 2002; Takai & Sengupta, 2017). It follows the call of recent research (Lyons et al., 2020; Trattner et al., 2019), where Hackl et al. (2020) suggest an operationalization of the - yet conceptual - impact model. An operationalized model is the foundation for a better understanding product family design's economic consequences. This is important since increased overdesign leads to cost-savings mainly in the indirect cost centers, whereas additional material costs are direct costs. Second, an operationalized model allows for generating various design alternatives and evaluating their economic consequences, supporting early-stage decision-making during product development (Fixson, 2006; Rezaie et al., 2008; Skirde et al., 2016). Finally, the operationalized model allows for investigating the complex relationships among variables driving complexity-induced costs by conducting numerical experiments. For example, several authors note the two-sided cost effects of component commonality leading to a U-shaped cost curve (Labro, 2004; Thonemann & Brandeau, 2000; Trattner et al., 2019). This curve is the result of cost-increasing and cost-decreasing effects. Takai and Sengupta (2017) and Takai (2019) demonstrate the interwoven nature of component commonality's economic consequences. Numerical experiments allow for a systematic analysis of these complicated cause-effect relationships by identifying direct and indirect effects among variables as well as conditions under which increased economies of scale cannot compensate for the increase in material costs caused by overdesign anymore.

This study demonstrates the cost-reducing effects of increased component overdesign and commonality for various departments such as development, intra-logistics, or purchasing. While these findings align with prior studies, a mixed effect on setup costs is observed. Component overdesign decreases setup costs only if the volume-pooling effects of replaced components outweigh the additional number of tasks to perform. It increases setup costs if replaced components share only some or no processes. The reason is that the overdesigned component requires both (prior independent) processes

---

[60] Literature also notes complexity-induced costs as variety or complexity costs (Wilson and Perumal, 2010; Hvam et al., 2020; Meßerschmidt et al., 2020).

but at a higher demand rate. Due to the cost in- and decreasing effects, a U-shaped cost curve of component commonality is observed. The apex of this curve indicates the optimal degree of component commonality at which total complexity-induced costs reach their minimum. Although the general cost curve is U-shaped, local minima exist due to the interaction of individual cost effects, such as noted for the setup costs. This study further demonstrates that component commonality is a lever for firms to increase product variety and, therefore, selling opportunities by keeping the total complexity-induced costs constant. In contrast to some studies (e.g., Ripperda & Krause, 2017; Schuh, 2005; Syam & Bhatnagar, 2015), a degressive rather than a progressive increase of complexity-induced costs under increasing product variety is observed. A progressive behavior occurs if higher-order effects such as learning and forgetting curves are added or cost rates are assumed to depend on product variety.

The results of this section are a further step towards operationalizing the effects of overdesign on complexity-induced costs. Since cause-effect relationships are more complicated, as Hackl et al.'s (2020) impact model suggests, further in-depth investigations are needed. Nevertheless, the work generalizes the two-sided face of component commonality as noted in the literature (e.g., Labro, 2004). It is noted that an increase in component commonality is not always the best choice for firms as it depends on the location of the current product family design (e.g., before or beyond the optimal point). Finally, this section highlights the extensive dimensions of the product family design space and calls for component overdesign heuristics.

The remainder of this section is organized as follows. Prior considerations are made in the next section. Analytical models for each of the eight complexity cost types are introduced in section 4.3. The experimental setup and the design of experiments are reported in section 4.4. The resulting data set is analyzed in section 4.5, which is separated into the individual complexity-induced cost effects as well as the total cost effects. Finally, this part ends with a conclusion, summarizing the results and discussing the implications for practitioners and researchers (section 4.6).

4.2 Prior Considerations

The starting point for this study is the impact model introduced in 2.6.2, which notes two sources of complexity-induced costs: overdesign and product variety[61]. Overdesign means that two or more components are replaced by an overdesigned one, which must fulfill the highest requirements of the components it replaces, leading to higher material costs (e.g., Eynan & Rosenblatt, 1996; Hackl et al., 2020). On the other side, it allows the reduction of unique components, leading to increased economies of scale due to

[61] To be precise and according to the induction of external complexity into firms via a product mix, overdesign is already a firm's reaction to product variety. From a practical perspective, product variety is the only cause of internal complexity. Nevertheless, this section aims to investigate the effects of overdesign under different product mixes (product variety and demand vectors). In doing so, overdesign and product variety are defined as independent variables to model the search for design alternatives under different product mixes.

increased reuse of components. Commonality describes the reuse of components across several products (Fisher, Ramdas, & Ulrich, 1999) and is a direct effect of increased component overdesign (Hackl et al., 2020). Related to the complexity theory, overdesign refers to all three dimensions of complexity in a systemic context (see Table 4.2). The number of system elements (multiplicity) is affected since an oversized one substitutes two or more components. Diversity is reduced (commonality increased) since more and more parts are standardized across products, leading to more homogenous product mix matrices in the physical, process, and resource domains.

Table 4.2:
The linkage between component overdesign, product variety effect and complexity.

Cause	Multiplicity	Interrelatedness	Diversity
Overdesign	-	-/+	-
Product Variety	+		-/+

While increasing overdesign has a complexity-decreasing effect on diversity and multiplicity, it has a two-sided effect on interrelatedness. Since more functional requirements need to be fulfilled by an oversized component, the relative number of interrelationships increases while the absolute number remains constant or decreases. Three domain mapping matrices representing the transition between functional requirements and physical domain ($DMM_{FD,PD}$) are given below. DMM_0 represents an initial design and DMM_1 as well as DMM_2 a stepwise overdesigned design. In the first step, components PD_3 and PD_2 are substituted by PD_{2*}. In the second step, PD_{2*} and PD_1 are further unified.

$$DMM_0 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix};\ DMM_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix};\ DMM_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad (4.1)$$

The absolute number of relations within DMM_0 and DMM_1 remain constant ($r_{abs} = 4$). However, component three is substituted and, therefore, DMM_1 turns into a 3-by-2 matrix by dropping the last column. In such cases, overdesign increases relative interrelatedness as the proportion of relations to system elements increases ($r_{rel} = 4/9 = 0.44\ vs.\ r_{rel} = 4/6 = 0.66$) while the absolute interrelatedness remains constant[62]. In the second step, the number of absolute relations is decreased ($r_{abs} = 3$) since both components already share requirements (FR_2). Since empty columns are dropped, DMM_2 turns into a 3-by-1 matrix with full density ($r_{rel} = 1$).

Product variety is the second independent variable. Often operationalized via the number of products, literature notes the importance of accounting for product differences (Buchholz, 2012; Trattner et al., 2019). For example, Stäblein et al. (2011) argue that product variety differs whether firms offer ten very similar or ten significantly different product variants. Since product variety is more than just the number of products, it is related to the dimensions of multiplicity and diversity. While the positive association

[62] Total complexity decreased since the number of system elements was reduced.

between variety and multiplicity is trivial, a mixed effect on diversity exists. Based on a given product mix, diversity is increased if the newly introduced products are dissimilar to the existing ones on average. Diversity reduces if more similar products are introduced. Although variety is more than just the sum of products offered, this thesis follows the common view by reducing product variety to the number of product variants (Trattner et al., 2019).

Complexity-induced costs are separated into costs occurring in one period or across multiple periods. Table 4.3 shows examples of complexity-induced costs according to the work of Hackl et al. (2020). While some costs, such as development or tooling, occur only in the initial period, some occur across multiple periods (e.g., stock or part administration costs). Additional material costs caused by component overdesign occur each time a product containing overdesigned components is produced. Since the EAD is a single-period model, there is no difference between these two groups. However, the distinction highlights the importance of measuring complexity-induced effects as costs sum up across periods.

Table 4.3:
As noted in the impact model, complexity-induced cost effects are categorized into one-time costs and costs occurring over multiple periods.

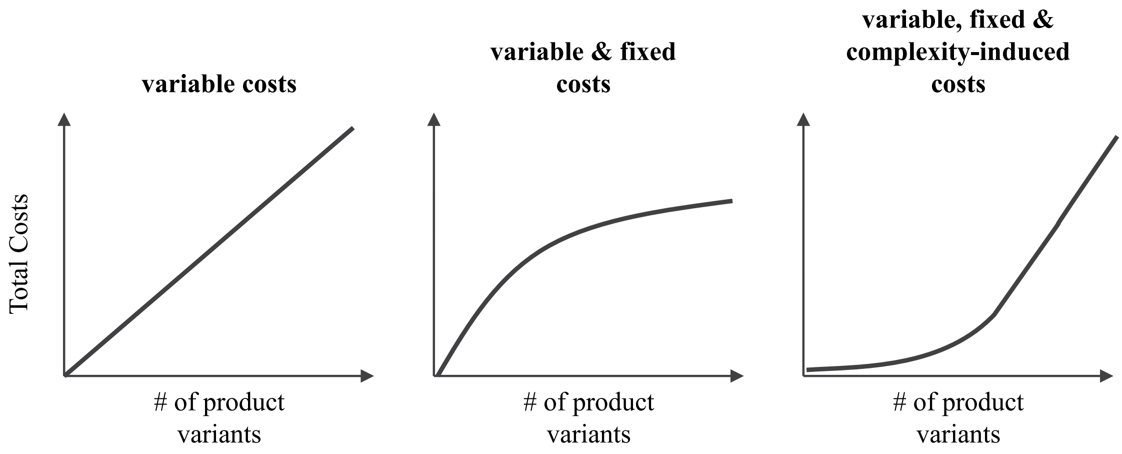
			Period	
Department	**Impact**	**Notation**	**Single**	**Multiple**
Development	Development costs (total & per unit)	TC_{devel}	x	
	Additional material costs due to overdesign	TC_{od}		x
	Costs for part administration	TC_{pa}		x
Procurement & Supply Chain	Stock costs	TC_{stock}		x
	Costs for supplier management	TC_{supply}		x
	Procurement order costs	TC_{order}		x
Manufacturing	Tooling costs	$TC_{tooling}$	x	
	Setup costs	TC_{setup}		x

The impact model raises no claim to completeness, and those eight cost drivers are only a subset of consequences existing in practice. Nevertheless, these effects are noted by many reviews and empirical studies across different fields, such as engineering, accounting, or operations management. For example, Perera et al. (1999) show the effects of component commonality along the entire lifecycle. They note that increased commonality has advantages all along the lifecycle. Therefore, commonality decisions should consider all lifecycle costs rather than phase-individual effects. While the cost advantages gained in a single lifecycle phase cannot outweigh the disadvantages, such as increased material costs, the sum of all advantages across all life phases can. A review by Labro (2004) further summarizes the two-sided face of component commonality. She highlights the trade-off between increasing commonality and decreasing costs firms face. This results in an optimal degree of component commonality, noted by case studies (e.g., Takai & Sengupta, 2017) and analytical studies (e.g., Heese & Swaminathan, 2006; Thonemann & Brandeau, 2000). A review by Trattner et al. (2019) summarizes the general impact of internal complexity on costs and operational performance. The latest achievement is the impact model by Hackl et al. (2020), which provides a foundation for this thesis. While these studies focused mainly on internal complexity in terms of commonality and coupling, another stream of literature investigates cost effects associated with product variety. S. W. Anderson (1995) shows the impact of product mix heterogeneity on overhead costs in manufacturing. Conducting a case study on different sides of a car manufacturer reveals that product, component, and process variety increases total labor and overhead hours, assembly downtime, and inventory (Fisher & Ittner, 1999). Other case studies confirm this impact (Ittner & Macduffie, 1995; Vachon & Klassen, 2002). However, they note that *"the impact of […] product variety […] is generally much less than the conventional manufacturing wisdom would predict"* (Macduffie et al., 1996, p. 350). Latest empirical studies, however, underline the cost-increasing effect of variety (e.g., Wan et al., 2012) where Lyons et al. (2020) surveyed 162 U.K. manufacturing firms and found a significant cost impact of product variety on the manufacturing process, material, process investment costs (e.g., tooling), purchasing and transportation.

In line with the microeconomic production model introduced in section 2.4.2, studies highlight the positive association between product variety and total costs (Lyons et al., 2020; Syam & Bhatnagar, 2015). However, the literature shows a wide range of cost curves with increasing product variety (see Figure 4.1). Using a Leontief production function, the ABL shows a linear cost increase with increasing product variety, as their model assumes that all costs are unit-level (variable) costs. By adding non-unit level (fixed) costs[63], a degressive behavior is expected since new products reuse already existing elements (tools, components, …). Therefore, Trattner et al. (2019) summarize that adding products similar to existing ones (related variety) is advantageous for firms while adding unrelated variety is not. According to the complexity-induced cost effects, development, part administration, supplier management, and tooling costs are fixed.

[63] See section 2.6.1 for a distinction between the traditional fixed/variable perspective on costs and the ABC cost hierarchy.

Figure 4.1:
Total costs as a function of product variety.



Literature investigating complexity-induced costs notes a progressive increase in total costs under increasing product variety (e.g., Ripperda & Krause, 2017; Syam & Bhatnagar, 2015). A progressive curve occurs if resource unit costs (cost rates) or resource consumption depend on product variety. Labro (2004) notes that some studies report an association between internal complexity and cost rates. For example, an increased product variety requires better-trained workers to handle more process variants and more distinct tooling equipment. Skilled staff, however, requires higher wages, increasing resource unit costs. The same applies to other costs, such as inventory holding, order processing, or supplier management costs. Labro (2004) adds that these effects still need to be clarified in literature, and empirical evidence is rare. Resource consumption as a function of product variety is a second influence, leading to progressive costs. For example, the effects of forgetting (Adler & Clark, 1991; Globerson & Levin, 1987; Yelle, 1979) lead to increased resource consumption under increasing variety. Workers require more time, resulting in a higher resource consumption to perform tasks that lie more in the past. Ripperda and Krause (2017) summarize that complexity-induced costs indicate inefficiencies in resource usage. The development time for a component dedicated to a single product variant is less time-consuming than designing a component for multiple variants due to additional integration efforts. Since this model assumes constant cost rates and does not model inefficiencies in resource usage, a degressive cost curve is expected. The following section introduces the individual models for simulating complexity-induced costs.

4.3 Operationalizing the Impact Model

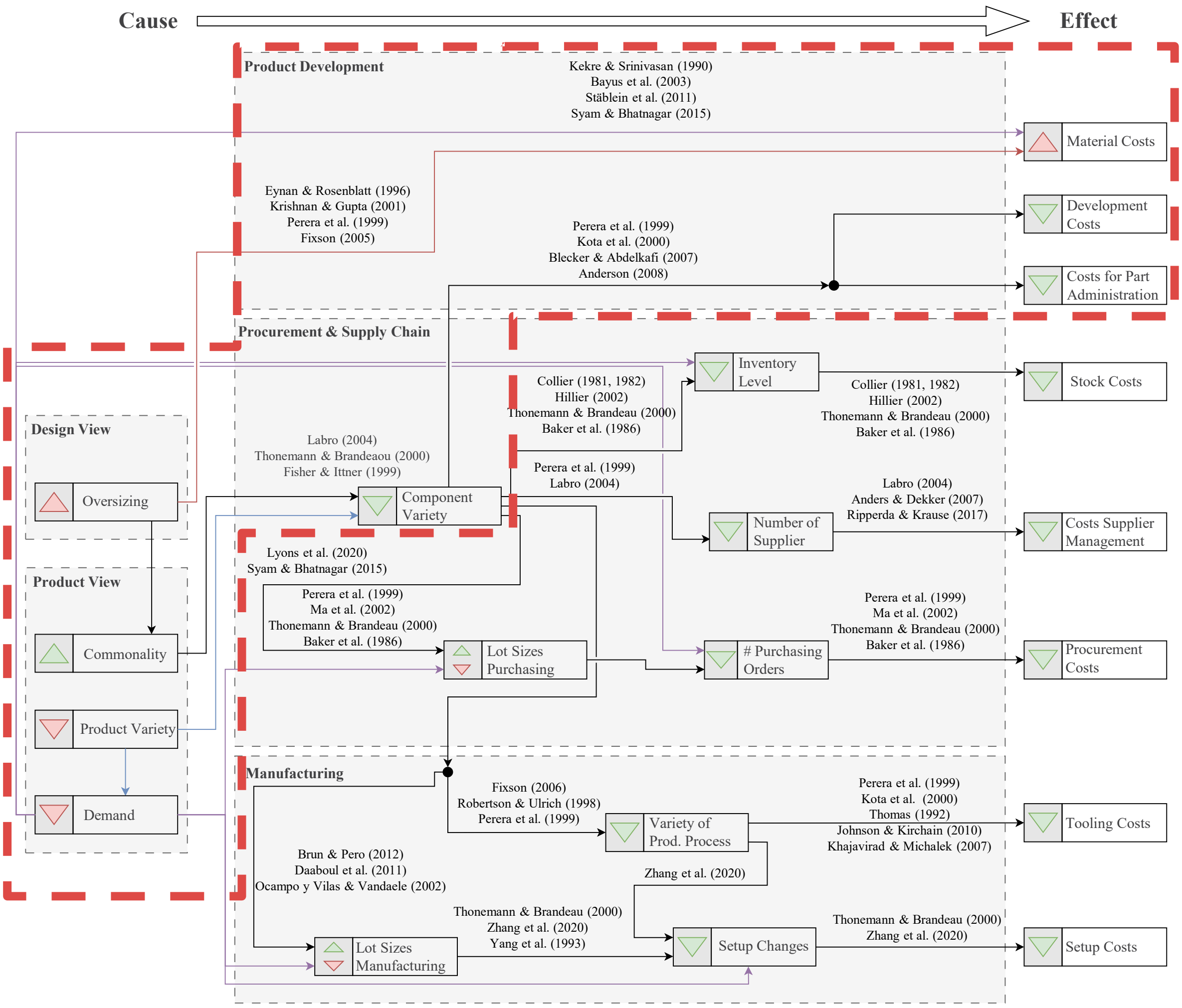
Analytical models for each complexity-induced cost effect, as noted in the impact model (see section 2.6.2), are introduced to investigate the economic consequences of internal complexity. Using existing models from literature and integrating them into the EAD allows the simulation of complexity-induced cost effects in the next step (sections 4.4 - 4.5). For better understanding, each driver is visualized by an example, where an interactive version is provided in this work's online appendix[64]. The last section (4.3.4) summarizes the individual models, relates them to the conceptual EAD model, and discusses the expected behavior of each driver.

4.3.1 Cost Effects in Development

Figure 4.2 provides an overview of cost drivers related to product development, as noted in the impact model. Additional costs of overdesign are mentioned by several authors (e.g., Krishnan & Gupta, 2001; Perera et al., 1999). According to Mertens (2020), overdesign is related to what Marc Meyer and Lehnerd (1997) call vertical leveraging. By substituting two or more components with an oversized one, components are standardized across product variants, increasing the degree of component commonality (Hackl et al., 2020). Material and (variable) manufacturing costs for overdesigned components are higher compared to none overdesigned ones as they need to fulfill more requirements (Eynan & Rosenblatt, 1996; Fixson, 2005; Labro, 2004) and result in unused functional requirements for specific product variants (Hackl et al., 2020).

[64] The framework is available under https://github.com/olemesser/EAD. Interactive examples are provided in the package's vignettes. An overview of available vignettes can be requested by running *utils::vignette(package = 'EAD'*)

Figure 4.2:
Cost effects of internal complexity in the development department.



While component overdesign increases the variable costs, the remaining drivers have cost-reducing effects under increasing commonality. Since the EAD takes the traditional cost view (unit- and non-unit-level costs), development and part administration costs are assumed to be fixed. Development costs cover all expenses arising during a component's initial development. They are fixed as they do not depend on the output quantity. Therefore, firms profit from reusing components across products since these costs are shared across larger volumes (Fixson, 2005; Harland & Uddin, 2014). Product variety is seen as a second lever influencing development costs since introducing new variants requires new components, enabling the additional functional requirements (Lyons et al., 2020). If firms offer only a small product mix, they must develop fewer components, reducing total development costs (Perera et al., 1999). The same applies to part administration and management costs, which depend on the number of material numbers. These costs result from redesigning components for internal or external reasons (Terwiesch & Loch, 1999) and their documentation in firms' IT systems. These costs arise within each period as reasons for changes occur all along the product lifecycle, such as changes in (external) legal norms or (internal) changes within the shop floor, as well as component optimization in terms of quality or costs (Chang, Shih, & Choo, 2011).

Different authors state that these costs can take up 20 to 50 percent of engineering capacity (Chang et al., 2011; Ehrlenspiel, Kiewert, Lindemann, & Hundal, 2007; Terwiesch & Loch, 1999). According to Blecker and Abdelkafi (2007), part management costs for a single component lay between \$5.000 to \$100.000 over the component lifecycle. The following sub-sections introduce the models for each complexity-induced cost effect associated with the development.

4.3.1.1 Component Overdesign

Overdesign is modeled by increasing the coupling with the domain mapping matrices $DMM_{FD,PD,S0}$ (e.g., Mertens, 2020), where the index $S0$ indicates the initial design state without any overdesign (reference simplicity). Each further manipulated design is denoted with a strictly increasing index $S1, S2, \ldots S_{Smax}$. Figure 4.3 to Figure 4.5 show an example product family consisting of four products. Starting with the initial state ($S0$), the components PD_1 and PD_2 are replaced by an overdesigned one (PD_{2*}) in the first step. PD_{2*} enables the functional requirements FR_1 and FR_2. Since the replaced component PD_1 required PD_4, the new component PD_{2*} is also coupled to PD_4 (step 2). In the third step, overdesign leads to further changes in $DMM_{PD,PrD,S1}$ as the new component requires processes used by PD_1. A view on the element unit costs indicates that the costs for PD_{2*} are equal to the sum of components it replaces. Under further increased overdesign ($S2$) PD_4 and PD_{2*} are replaced by PD_{2**} in step four. Since PD_4 is now fully integrated in PD_{2**}, the intra-domain relationship can be removed (step five). However, the new component PD_{2**} needs to fulfill more functional requirements and thus requires the process PrD_4, used by the replaced component PD_4 (step six). Due to a further increased overdesign the unit costs for PD_{2**} increase compared to PD_{2*}.

Figure 4.3:
Initial product family design (S0)

S0																			
	FR_1	**FR_2**	**FR_3**	**PD_1**	**PD_2**	**PD_3**	**PD_4**	**PrD_1**	**PrD_2**	**PrD_3**	**PrD_4**	**RD_1**	**RD_2**	**RD_3**	**RD_4**	**DMD**
FR_1				1	0	0	0									
FR_2				0	1	1	0									
FR_3				0	0	1	1									
PD_1				0	0	0	1	1	0	0	0					
PD_2				0	0	0	0	0	1	3	0					
PD_3				0	0	0	0	0	2	0	0					
PD_4				0	0	0	0	0	0	0	1					
PrD_1								0	0	0	0	1	0	0	0	
PrD_2								0	0	0	0	0	3	5	0	
PrD_3								0	0	0	0	0	0	1	0	
PrD_4								0	0	0	0	0	0	0	2	
P1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	2	10
P2	0	1	0	0	1	1	0	0	3	3	0	0	9	18	0	5
P3	0	0	1	0	0	1	1	0	2	0	1	0	6	10	2	2
P4	0	1	1	0	1	2	1	0	5	3	1	0	15	28	2	4
variable Resource Costs (RC_{var})												100	800	600	500	
fixed Resource Costs (RC_{fix})												700	700	700	700	
total Consumption (TCC,TPC,TRC)				10	9	15	16	10	39	27	16	10	117	222	32	
variable Unit Costs (PDUC, PrDUC)				10	42,13	68,05	31,25	10	34,03	2,7	31,3	10,0	6,8	2,7	15,6	
fixed Unit Costs ($PDUC_{fix}$, $PrDUC_{fix}$, $RDUC_{fix}$)				70,0	43,2	67,4	43,8	70,0	33,7	3,2	43,8	70,0	6,0	3,2	21,9	
fixed Costs (PDC_{fix}, $PrDC_{fix}$)				700	389	1011	700	700	1315	85	700					
total variable Costs (TC_{var})					2000				2000				2000			
total fixed Costs (TC_{fix})					2800				2800				2800			
total Costs (TC)					4800				4800				4800			

Figure 4.4:
Product family design after component PD_1 and PD_2 have been replaced by an overdesigned one PD_{2*}

S1																
	FR_1	**FR_2**	**FR_3**	**PD_1**	**PD_{2*}**	**PD_3**	**PD_4**	**PrD_1**	**PrD_2**	**PrD_3**	**PrD_4**	**RD_1**	**RD_2**	**RD_3**	**RD_4**	**DMD**
FR_1				0	1	0	0									
FR_2				0	1	1	0									
FR_3				0	0	1	1									
PD_1				0	0	0	0	0	0	0	0					
PD_{2*}				0	0	0	1	1	1	3	0					
PD_3				0	0	0	0	0	2	0	0					
PD_4				0	0	0	0	0	0	0	1					
PrD_1								0	0	0	0	1	0	0	0	
PrD_2								0	0	0	0	0	3	5	0	
PrD_3								0	0	0	0	0	0	1	0	
PrD_4								0	0	0	0	0	0	0	2	
P1	1	0	0	0	1	0	1	1	1	3	1	1	3	8	2	10
P2	0	1	0	0	1	1	1	1	3	3	1	1	9	18	2	5
P3	0	0	1	0	0	1	1	0	2	0	1	0	6	10	2	2
P4	0	1	1	0	1	2	1	1	5	3	1	1	15	28	2	4
variable Resource Costs (RC_{var})																
fixed Resource Costs (RC_{fix})												700	700	700	700	
total Consumption (TCC,TPC,TRC)				0	19	15	21	19	49	57	21	19	147	302	42	
variable Unit Costs (PDUC, PrDUC)				0	52,13	68,05	31,25	10	34,03	2,7	31,3	10,0	6,8	2,7	15,6	
fixed Unit Costs ($PDUC_{fix}$, $PrDUC_{fix}$, $RDUC_{fix}$)				0,0	69,67	51,75	33,33	36,8	25,88	2,32	33,3	36,8	4,8	2,3	16,7	
fixed Costs (PDC_{fix}, $PrDC_{fix}$)				0	1324	776	700	700	1268	132	700					
total variable Costs (TC_{var})					2668				2668				2668			
total fixed Costs (TC_{fix})					2800				2800				2800			
total Costs (TC)					5468				5468				5468			

Figure 4.5:
Product family design after component PD_{2*} and PD_4 have been replaced by an overdesigned one PD_{2**}

S2	FR_1	FR_2	FR_3	PD_1	PD_{2**}	PD_3	PD_4	PrD_1	PrD_2	PrD_3	PrD_4	RD_1	RD_2	RD_3	RD_4	DMD
FR_1				0	1	0	0									
FR_2				0	1	1	0									
FR_3				0	1	1	0									
PD_1				0	0	0	0	0	0	0	0					
PD_2				0	0	0	0	1	1	3	1					
PD_3				0	0	0	0	0	2	0	0					
PD_4				0	0	0	0	0	0	0	0					
PrD_1								0	0	0	0	1	0	0	0	
PrD_2								0	0	0	0	0	3	5	0	
PrD_3								0	0	0	0	0	0	1	0	
PrD_4								0	0	0	0	0	0	0	2	
RD_1												0	0	0	0	
RD_2												0	0	0	0	
RD_3												0	0	0	0	
RD_4												0	0	0	0	
P1	1	0	0	0	1	0	0	1	1	3	1	1	3	8	2	10
P2	0	1	0	0	1	1	0	1	3	3	1	1	9	18	2	5
P3	0	0	1	0	1	1	0	1	3	3	1	1	9	18	2	2
P4	0	1	1	0	2	2	0	2	6	6	2	2	18	36	4	4
variable Resource Costs (RC_{var})																
fixed Resource Costs (RC_{fix})												700	700	700	700	
total Consumption (TCC,TPC,TRC)				0	25	15	0	25	55	75	25	25	165	350	50	
variable Unit Costs (PDUC, PrDUC)				0	83,38	68,05	0	10	34,03	2,7	31,3	10,0	6,8	2,7	15,6	
fixed Unit Costs ($PDUC_{fix}$, $PrDUC_{fix}$, $RDUC_{fix}$)				0,0	84,73	45,45	0	28,0	22,73	2	28	28,0	4,2	2,0	14,0	
fixed Costs (PDC_{fix}, $PrDC_{fix}$)				0	2118	682	0	700	1250	150	700					
total variable Costs (TC_{var})					3105				3105				3105			
total fixed Costs (TC_{fix})					2800				2800				2800			
total Costs (TC)					5905				5905				5905			

EADs' ability to model the increasing variable costs caused by overdesign is demonstrated using the example where Table 4.4 summarizes costs and the corresponding PFCMs. As argued, overdesign (SDC_N) is positively associated with the degree of component commonality (PCI_{PD}). Since the new component is at least as expensive as the most expensive component it replaces (Eynan & Rosenblatt, 1996), the variable costs increase with increasing overdesign. Fix costs remain constant as their effect is modeled separately in the development and part administration costs.

Table 4.4:
Summary of design scenarios.

Scenario	Description	N_{PD}	N_{PrD}	N_{RD}	SDC_N	PCI_{PD}	DNS_{PrD}	DNS_{RD}	Costs var	Costs fix	Costs total
S0	Starting Scenario	4	4	4	0.11	52%	56%	63%	2000	2800	4800
S1	Component PD_1 and PD_2 replaced by PD_{2*}	3	4	4	0.21	83%	88%	94%	2668	2800	5468
S2	Component PD_4 and PD_{2*} replaced by PD_{2**}	2	4	4	0.36	87%	100%	100%	3105	2800	5905

In order to separate complexity-induced costs from non-complexity-induced ones, the additional material costs caused by overdesign (TC_{od}) between the initial starting design (S_0, TC_{var,S_0}) and the current design (S_N, TC_{var,S_N}) are defined as:

$$TC_{od} = TC_{var,S_N} - TC_{var,S_0} \tag{4.2}$$

The total material costs are defined as the sum product of components' variable costs and their demand as:

$$TC_{var} = \sum_{i=1}^{N_{PD}} C_{var,i} * DMD_{PD,i} \tag{4.3}$$

Eynan and Rosenblatt (1996) note that the variable cost of an oversized component PD_* ($C_{var,PD*}$) fulfills the following equation:

$$C_{var,PD*} \geq \max\,(C_{var,PD1}; C_{var,PD2}) \tag{4.4}$$

$C_{var,PD*} = \max\,(C_{var,PD1}; C_{var,PD2})$ if one of the replaced components already fulfills the functional requirements of the others. This leads to the notion that $TC_{od} \geq 0$ if the degree of overdesign increases from S_0 to S_N and $TC_{od} \leq 0$ if overdesign decreases. Following this constraint and aiming to further specify the increase in material costs for PD_*, a uniform probability distribution ($U[lb; ub]$) with $lb \geq 0$ being the lower and $lb \leq ub \leq 1$ being the upper bound is added to equation (4.4) leading to:

$$C_{var,PD*} = \max\big(C_{var,PD1}; C_{var,PD2}\big) + \min\big(C_{var,PD1}; C_{var,PD2}\big) * \; U[lb; ub] \tag{4.5}$$

The model ensures that the minimum costs of the overdesigned component are always equal to or greater than the maximum costs of those it replaces, satisfying the constraint noted by Eynan and Rosenblatt (1996). Low levels of lb and ub reflect cases where component´s PD_1 and PD_2 already share similarities and, therefore, the costs of the oversized component are lower than the sum of individual material costs. High values for lb and ub, on the other side, reflect the substitution of dissimilar components. To be precise, $ub \leq 1$ is not the theoretical possible upper bound. If two completely different and highly complex components are standardized, the variable costs of PD_* can be larger than the sum of the individual components. However, this model assumes that the maximum costs are less than the sum of individual component costs. As another assumption, the model always takes two components in each overdesign step. In practice, however, cases might exist where several components are replaced by one.

4.3.1.2 Component Development

The previous driver argued that oversized components must fulfill more requirements at once, and unit material costs are at least the maximum costs of all components it replaces (Eynan & Rosenblatt, 1996). The same argument applies to the development costs. Unifying two individual components removes a prior existing interface between both components, which has a reducing effect on development costs. On the other hand, integrating more functional requirements into one component requires more time and, thus, increases development costs (Ripperda & Krause, 2017). Takai and Sengupta (2017) argue that there are cases when designing an oversized component is cheaper than designing several individual components due to bundling of efforts. However, they also note that the condition of Eynan and Rosenblatt (1996) still applies. Therefore, the development costs of a joined component PD_* ($C_{dvl,PD*}$) are defined as follows:

$$C_{dvl,PD*} = \max\left(C_{dvl,PD1}; C_{dvl,PD2}\right) + \min\left(C_{dvl,PD1}; C_{dvl,PD2}\right) \; * \; U[lb; ub] \quad (4.6)$$

Setting the upper bound to $ub = 1$ ensures that the joined development costs are less than the individual development costs, reflecting empirical patterns. Empirical studies note that development costs decrease under increasing commonality, although individual components hold more functional requirements (e.g., Ericsson & Erixon, 1999; Hackl et al., 2020). Component development costs are modeled as a ratio (R_{dvl}) of components fixed costs (PDC_{fix})[65] according to:

$$C_{dvl} = R_{dvl} * PDC_{fix} \quad (4.7)$$

Components' fixed costs (PDC_{fix}) are implicitly given by EAD by tracing fixed resource costs (RC_{fix}) back into the physical domain. This is done by estimating the transformation matrix from the physical domain via processes into the resource domain according to the rules of the EAD. Since this transformation is already shown at several points in this work, the transformation matrix is just denoted as ($A_{PD,RD}$). Multiplication of $A_{PD,RD}$ with the component demand vector (DMD_{PD}) and normalization of columns lead to the proportional resource consumption for each component (PRC_{PD}).

$$PRC_{PD,ij} = \frac{A_{PD,RD,ij} * DMD_{PD,i}}{\sum A_{PD,RD,ij} * DMD_{PD,i}} \quad (4.8)$$

Matrix vector multiplication with the fixed resource costs leads to component's fixed costs defined as:

$$PDC_{fix} = PRC_{PD,ij} * RC_{fix} \quad (4.9)$$

Summarizing the development costs for all components results in the total development costs given as:

$$TC_{devel} = \sum_{i=1}^{N_{PD}} R_{dvl,i} * PDC_{fix,i} = \sum_{i=1}^{N_{PD}} C_{dvl,i} \quad (4.10)$$

Product variety is a second effect, driving development costs. When firms decide to introduce new products, these products may require new components to be developed, increasing development costs. However, dropping products does not reduce the development costs due to cost stickiness (remanence) in practice. However, if the EAD is used before the main development activities start, this effect is neglectable, and a decrease in product variety can lead to a reduction in development costs. Figure 4.6 shows three product mixes representing the physical domain (P_{PD}). While the first product mix ($S0$) contains four products, the number of products is reduced to three in $S1$ and to two for $S2$. The example further assumes that the initial development costs are $R_{dvl} = 30\%$ of component's fixed costs. It shows that as long as a component is used by at least one product with non-zero demand (total component consumption, $TCC > 0$), development costs (TC_{dvl}) occur. Since PD_1 is used exclusively by product P_1 which is dropped

---

[65] For an example, see Figure 4.3.

in step $S1$, the development costs decrease from $TC_{dvl,S0} = 840$ to $TC_{dvl,S1} = 630$. In another scenario ($S2$), product P_4 is excluded. Fixed costs, however, do not decrease further since P_4 uses components that are still required by other products within the product mix.

Figure 4.6:
Three design scenarios for the reduction of fixed costs through decreasing product variety.

S0	PD_1	PD_2	PD_3	PD_4	DMD
P1	1	0	0	1	10
P2	0	1	1	0	50
P3	0	0	1	1	100
P4	0	1	2	1	30
TCC	10	80	210	140	
PDC_{fix}	700	389	1011	700	
R_{dvl}	30%	30%	30%	30%	
R_{PA}	20%	20%	20%	20%	
TC_{dvl}	210	117	303	210	**840**
TC_{PA}	140	78	202	140	**560**
TC_{dvl}+TC_{PA}	350	194	506	350	**1400**

S1	PD_1	PD_2	PD_3	PD_4	DMD
P1	1	0	0	1	0
P2	0	1	1	0	50
P3	0	0	1	1	100
P4	0	1	2	1	30
TCC	0	80	210	130	
PDC_{fix}	700	389	1011	700	
R_{dvl}	30%	30%	30%	30%	
R_{PA}	20%	20%	20%	20%	
TC_{dvl}	0	117	303	210	**630**
TC_{PA}	0	78	202	140	**420**
TC_{dvl}+TC_{PA}	0	194	506	350	**1050**

S2	PD_1	PD_2	PD_3	PD_4	DMD
P1	1	0	0	1	0
P2	0	1	1	0	50
P3	0	0	1	1	100
P4	0	1	2	1	0
TCC	0	50	150	100	
PDC_{fix}	700	389	1011	700	
R_{dvl}	30%	30%	30%	30%	
R_{PA}	20%	20%	20%	20%	
TC_{dvl}	0	117	303	210	**630**
TC_{PA}	0	78	202	140	**420**
TC_{dvl}+TC_{PA}	0	194	506	350	**1050**

4.3.1.3 Part Management and Administration

Part management and administration costs are caused by the redesigning of components and their documentation of changes across firms' IT systems. These costs decrease with increasing overdesign or reducing component variety (Fisher & Ittner, 1999; Labro, 2004) or product variety (Lyons et al., 2020). This work assumes that each component represents an individual material number. There are no part administration costs ($C_{pa,j}$) if a component is not used in any product. Within the model, total part administration costs (TC_{pa}) are defined as:

$$TC_{pa} = \sum C_{pa,j} * \zeta_j; \quad \zeta_j = \begin{cases} 1, & TCC > 0 \\ 0, & TCC = 0 \end{cases} \tag{4.11}$$

Part management costs for a component j are denoted as $C_{pa,j}$. ζ_j indicates whether a component is used ($TCC > 0$) or not ($TCC = 0$) in the current product mix. TCC represents the total component consumption and is calculated as:

$$TCC = P_{PD}^{T} * DMD \tag{4.12}$$

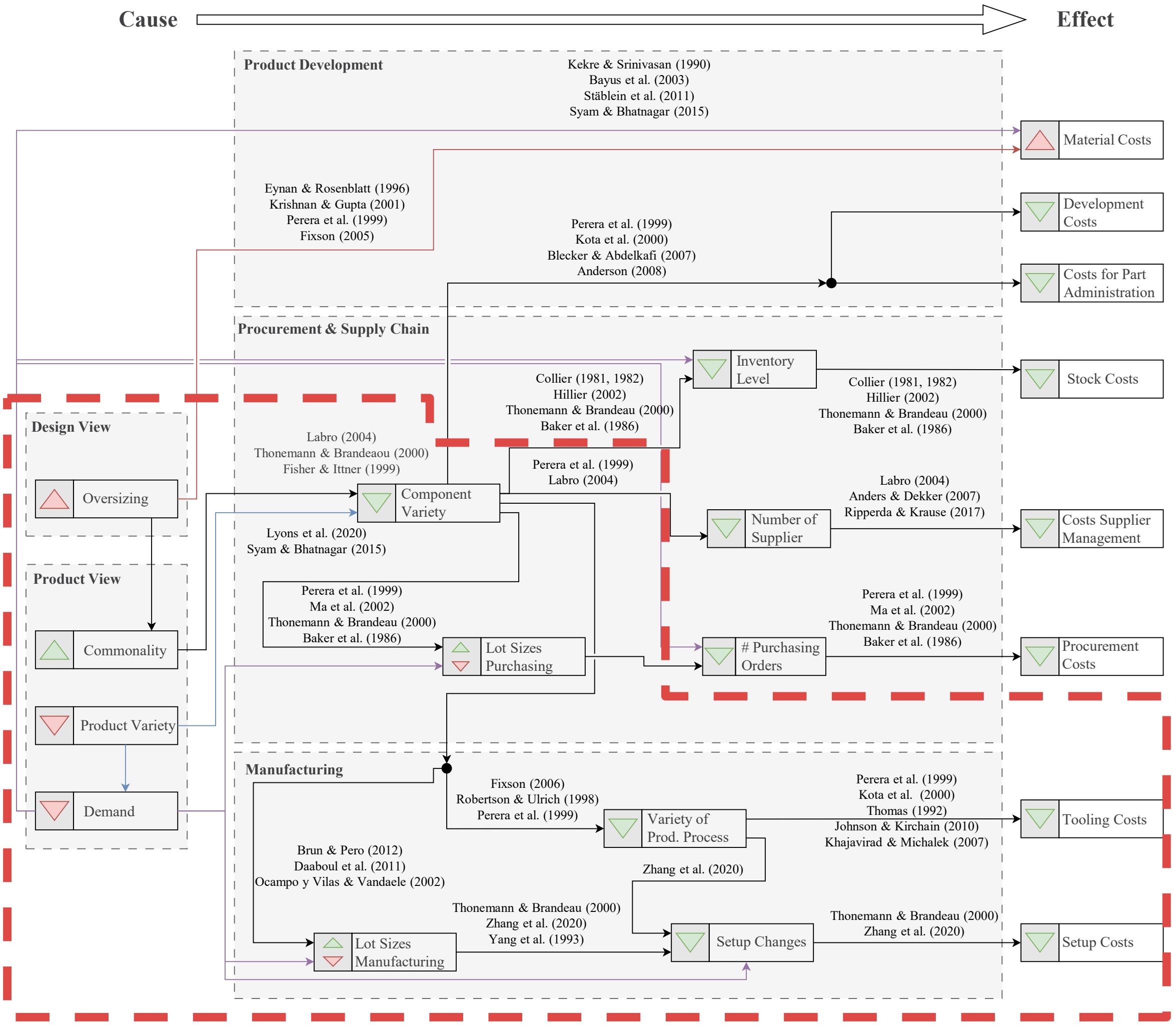
Figure 4.6 and Figure 4.5 show an example where costs decrease from $S0$ to $S1$ since PD_1 is used exclusively by products that are dropped in $S1$. However, they remain constant between $S1$ and $S2$ since the reduction of variety does not allow for dropping further components. As for the development costs, this driver assumes an ideal environment where later product demand is known beforehand. In practice, components are developed without knowledge of the later demand, thus leading to development and part administration costs of unused components.

4.3.2 Cost Effects in Production

In this section, the effect of internal complexity on setup and tooling costs is modeled for the production department (see Figure 4.7). Setup changes occur when a successor component manufacturing task requires a different setup on the same machine, such as different holding devices, materials, tools (L. L. Zhang et al., 2020), or trail runs and adjustments (Horngren et al., 2012) compared to the current machine setup[66]. Thus, a higher number of individual components leads to more setup changes and, therefore, additional costs as it increases the chance of setup changes. Typically, an increased number of individual components is caused by an increase in product variety where several studies note the positive association between product variety and the total setup costs (Brun & Pero, 2012; Daaboul et al., 2011; Escobar-Saldívar, Smith, & González-Velarde, 2008; Trattner et al., 2019). Since commonality and modularity reduce the number of components, they allow firms to dampen the increase in setup costs (Chiu & Okudan, 2012; Fixson, 2007; Thonemann & Brandeau, 2000). Processes for which increasing product variety causes a slight cost increase are called flexible, while processes for which costs increase strongly are called inflexible (Ocampo y Vilas & Vandaele, 2002). Maimon, Dar-El, and Carmon (1993) add that setup costs are not solely driven by component variety alone. In a case study, they found that some scheduling methods perform better regarding throughput (all standardized components at once), while others perform better regarding the work-in-progress (similar components follow each other) than traditional sequential manufacturing. Collier (1981) investigates the amount of inventory carrying and setup costs under different levels of commonality and lot-size models. He finds that lot-for-lot models profit most from increased commonality but still have higher costs than economic order quantity models.

[66] L. L. Zhang et al. (2020) explicitly differ between costs due to tool and setup changes. However, both costs follow the same logic and are summarized as setup costs for this work.

Figure 4.7:
Complexity-induced costs in the production department.



Tooling costs are noted as a second cost driver in manufacturing, as noted in the literature (M. D. Johnson & Kirchain, 2010; Khajavirad & Michalek, 2007; Kota et al., 2000; Perera et al., 1999; Thomas, 1992). Tooling equipment is necessary to place components into manufacturing machines, transport components within the shop floor, or hold components during assembly in place. These costs are caused by component and process variety and show a wide range of costs. While simple metal frames are sufficient for assembling and transporting specific components, pressing or injection molding tools for certain processes can easily cost up to $100.000. Therefore, tooling costs depend on component variety and process variety (N_{proVar}), which is the distinct combination of components and processes.

4.3.2.1 Setup Costs

Setup costs result from setup activities on the shop floor. They are variable costs as the number of setups depends on the produced demand (Horngren et al., 2012). In order to model this effect, this study uses the studies of Thonemann and Brandeau (2000) and L. L. Zhang et al. (2020) as a foundation. These studies already report models that can be adapted. The total setup costs (TC_{setup}) within a period for a specific process j and component i are calculated as:

$$TC_{setup,j} = \sum_{i=1}^{N_{PD}} n_{setup,ij} * C_{setup,ij} \tag{4.13}$$

with $n_{setup,ij}$ being the number of setups for a component i and a process j. The total setup costs for a process j ($TC_{setup,j}$) are than defined as the sum product of $n_{setup,ij}$ and the costs for a single setup ($C_{setup,ij}$). The number of setups depends on three factors. First, the lot size (N_{lot}), second, the total demand for each component ($DMD_{PD,ij}$), which goes through a process j and, finally, the scheduling method (Horngren et al., 2012; Maimon et al., 1993). If a component i does not use a specific process j_{unused} the demand is $DMD_{PD,ij_{unused}} = 0$. Using the -model by L. L. Zhang et al. (2020), the number of setups for a specific process is than given as:

$$n_{setup,j} = \sum_{i=1}^{N_{PD}} \frac{DMD_{PD,ij}}{N_{lot,ij}} * \delta(S_{jt-1}, S_{jt}) \tag{4.14}$$

Since setup costs only occur if the actual setup differs (S_t) from the previous setup (S_{t-1}), the δ operator is used, which is $\delta = 1$ if $S_{t-1} \neq S_t$ and zero otherwise. While the left factor of equation (4.14) refers to classical accounting textbooks (Horngren et al., 2012) and assumes that setups are changed each time, the second factor allows two lots of the same component to be produced without any setup changes. Component's individual lot sizes are stored in the lot size matrix ($LZM = N_{lot,ij}$) and are calculated using the economic order quantity model (e.g., Labro, 2018; Nahmias, 2011). The model is widely applied in studies investigating the effects of product design on costs (e.g., Takai & Sengupta, 2017; Thonemann & Brandeau, 2000) and is defined as:

$$LZM = N_{lot,ij} = \sqrt{\frac{2 * DMD_{PD,ij} * C_{order}}{C_{hold,i}}} \tag{4.15}$$

While C_{order} represents the costs of placing an order, $C_{hold,i}$ represents the costs of holding a specific component in stock (Russell & Taylor, 2005). According to the economic order model, it is beneficial for firms to produce smaller lot sizes if they have high holding costs and produce larger lots under high order costs. A common assumption is to use fixed order costs, which implies that the costs do not depend on the ordered quantity (e.g., Gavirneni, 2002). However, order and holding costs vary across components for several reasons. First, if components are ordered and processed in larger batches, more space in stock is required, increasing the holding costs. Although two components have the same batch size, their holding costs can differ. For example, a high-precision component cannot be stored in a cheaper outdoor stock compared to a more robust raw material. Order costs, on the other hand, can vary across suppliers. Nevertheless, for reasons of simplification, this thesis assumes that order and holding costs are constant across components and, therefore, the index i is dropped for C_{hold} and C_{order} leading to[67]:

$$LZM = N_{lot,ij} = \sqrt{\frac{2 * DMD_{PD,ij} * C_{order}}{C_{hold}}} \tag{4.16}$$

Order and holding costs are user-defined input parameter and will be used in other drivers introduced in section 4.3.3.

The scheduling is still missing and needs to be included. The scheduling decides whether setup costs occur or not. No setup is required if two lots of the same components follow each other. While Maimon et al. (1993) compare different setup methods, this work assumes a random order following Kekre (1987) for reasons of simplification. A random processing order is generated as follows:

$$TM = N_{task,ij} = \begin{cases} \left\lceil \frac{DMD_{PD,ij}}{N_{lot,ij}} \right\rceil, & N_{lot,ij} \neq 0 \\ 0, & N_{lot,ij} = 0 \end{cases} \tag{4.17}$$

TM represents the task matrix, holding information on the number of tasks necessary to fulfill the demand for each component (i) and process (j). These tasks are then executed randomly for each process where the execution vector (E_j) stores this information. For example, let TM be the task matrix defined as $TM = N_{task,ij}$ with three components ($i = 1 \dots 3$) and two processes ($j = 1 \dots 2$) given as:

$$TM = \begin{pmatrix} 0 & 1 \\ 1 & 2 \\ 2 & 1 \end{pmatrix} \tag{4.18}$$

According to the example, component two needs one batch for process PrD_1 and two for PrD_2. Possible execution vector for process one E_1 and process two E_2 are:

$$E_1 = [3 \quad 2 \quad 3] \tag{4.19}$$

$$E_2 = [3 \quad 2 \quad 2 \quad 1] \tag{4.20}$$

The example shows that components manufactured with more lots are more likely to be executed after another. In E_2, no setup between the second and third tasks is required since the same component is processed. It further shows that the execution vector has a different length depending on the selected process. L. L. Zhang et al. (2020) note that setups occur if a setup change is required. Therefore, the total number of setups for a process j is calculated as:

$$n_{setup,j} = 1 + \sum_{t=1}^{t_{max,j}} \delta_j\left(E_{jt-1}, E_{jt}\right), \ \ \delta_j = \begin{cases} 1, & E_{jt} \neq E_{jt-1} \\ 0, & E_{jt} = E_{jt-1} \end{cases} \tag{4.21}$$

[67] Several simplifications are made by using this model. According to the more detailed model by Thonemann and Brandeau (2000), dependencies between holding costs are more complicated.

In this example, the number of setups for E_1 and E_2 is three due to an initial setup.

The second part of equation (4.13) requires defining the setup costs $C_{setup,ij}$. These costs can vary across processes and components. Even for the same process, a simple component may require less time compared to a complex component. For reasons of simplification, the model assumes that the setup costs vary only across processes ($C_{setup,j}$)[68]. Under the assumption of fixed setup costs per process ($C_{setup,j}$) the equation (4.13) turns into:

$$TC_{setup,j} = \sum_{t=1}^{t_{max,j}} \delta_j(E_{jt-1}, E_{jt}) * C_{setup,j} *, \delta_j = \begin{cases} 1, & E_{jt} \neq E_{jt-1} \\ 0, & E_{jt} = E_{jt-1} \end{cases} \tag{4.22}$$

A summation of $TC_{setup,j}$ across all processes leads to the total setup costs. Empirical data from a manufacturing firm is used to get valid assumptions for the setup costs C_{setup}. The firm produces large gears that transform rotational speed and torque from large machines such as gas turbines. A gear consists of four main parts: an input shaft with a mounted cogwheel and an output shaft with another cogwheel. In sum, 17 input shaft variants and 16 component variants for both cogwheels and output shafts were analyzed regarding their production time, separated into setup and processing time. Table 4.5 reports the results and shows that the setup costs are responsible for 20 - 40% of the total manufacturing costs on average. There are also some cases where the setup costs are more than 50% of component manufacturing costs, indicating a time-consuming setup. It is important to note that the data may be biased since the firm produces large (1-4 m) components in batch size one. For less complicated components produced in larger batches, setup costs may lay at the lower end (5% - 20%).

Table 4.5:
Proportion of setup costs on total manufacturing costs for a single lot across different variants.

Component	min	$Q_{5\%}$	median	mean	$Q_{95\%}$	max	sd	N
Input Shaft	5.5%	18.1%	33.2%	34.1%	48.0%	49.0%	0.108	17
Input Cogwheel	19.0%	23.2%	38.5%	39.9%	53.7%	60.5%	0.097	16
Output Cogwheel	15.0%	15.9%	19.6%	20.2%	25.0%	31.7%	0.036	16
Output Shaft	13.3%	14.7%	18.6%	21.2%	33.7%	48.0%	0.080	16

Modeling setup costs may seem complicated; however, the thesis reuses the models by Thonemann and Brandeau (2000) and L. L. Zhang et al. (2020). A more detailed explanation for several micro assumptions is provided in these studies. Nevertheless, a small example is provided in thesis' online appendix to visualize the modeling of setup costs[69]. Although the implementation relies on two widely used models, it makes certain assumptions. First, the setup time and, therefore, costs are averaged across components

[68] From a perspective of modelling, an inclusion of component dependency is possible with the given notation. However, the thesis omits this aspect for reasons of simplification.

[69] The interactive example can be accessed by running *utils::vignette('productionCostEffects', package = 'EAD')*

for each process. Thus, the effect of different setup activities for individual components is not considered. The model further assumes that setup time depends only on the setup of the current component and not the previous setup, indicating that the machine is fully unloaded and prepared. In practice, firms can gain advantages if they cluster components that use similar setups. Other but more sophisticated scheduling methods consider this (e.g., see Maimon et al., 1993).

4.3.2.2 Tooling Costs

Tooling costs describe the efforts to produce tooling and additional manufacturing support equipment necessary to produce a specific component. Literature notes that components can differ in size, shape, materials, processes, or assembly, requiring distinct tooling equipment (M. D. Johnson & Kirchain, 2010). Constructing, manufacturing, and managing these manufacturing support equipment result in costs that are summarized as tooling costs (Kota et al., 2000). The EAD does not differ components regarding their properties, such as size and shape. Components are only characterized by the processes they use. Therefore, this model assumes that distinct tooling equipment is needed for each component-process combination. In doing so, the process variety matrix ($PVM \in \mathbb{R}^{N_{PD} x N_{PrD}}$) is defined as:

$$PVM = DMM_{PD,PrD} + DMM_{PD,PrD} DSM_{PrD} \tag{4.23}$$

This matrix describes the transformation between the physical and process domains as it holds information on which components (rows) require which processes (columns). Each non-zero entry in PVM represents a process variant. PVM is binarized since the information on whether a component uses a specific process is relevant rather than the absolute process consumption.

$$PVM_{bin} = \begin{cases} 1, & PVM > 0 \\ 0, & PVM = 0 \end{cases} \tag{4.24}$$

The total number of process variants (N_{proVar}) is than calculated as:

$$N_{proVar} = \sum PVM_{bin} \tag{4.25}$$

Following M. D. Johnson and Kirchain's (2010) work, it is assumed that each process variant requires its own tooling equipment. The tooling cost matrix $C_{tooling} \in \mathbb{R}^{N_{PD} x N_{PrD}}$ defines the tooling costs for each process variant. Since tooling costs occur if and only if a process variant is used, rows in PVM_{bin} for zero-demand components are also zero. Therefore, equation (4.24) turns into:

$$PVM_{bin} = \begin{cases} 1, & PVM_{ij} > 0 \ \wedge \ DMD_{PD,ij} > 0 \\ 0, & PVM_{ij} = 0 \end{cases} \tag{4.26}$$

Element-wise multiplication (denoted as $\odot$) with the tooling costs matrix and subsequent summation than leads to the total tooling costs:

$$TC_{tooling} = \sum C_{tooling} \odot PVM_{bin} \tag{4.27}$$

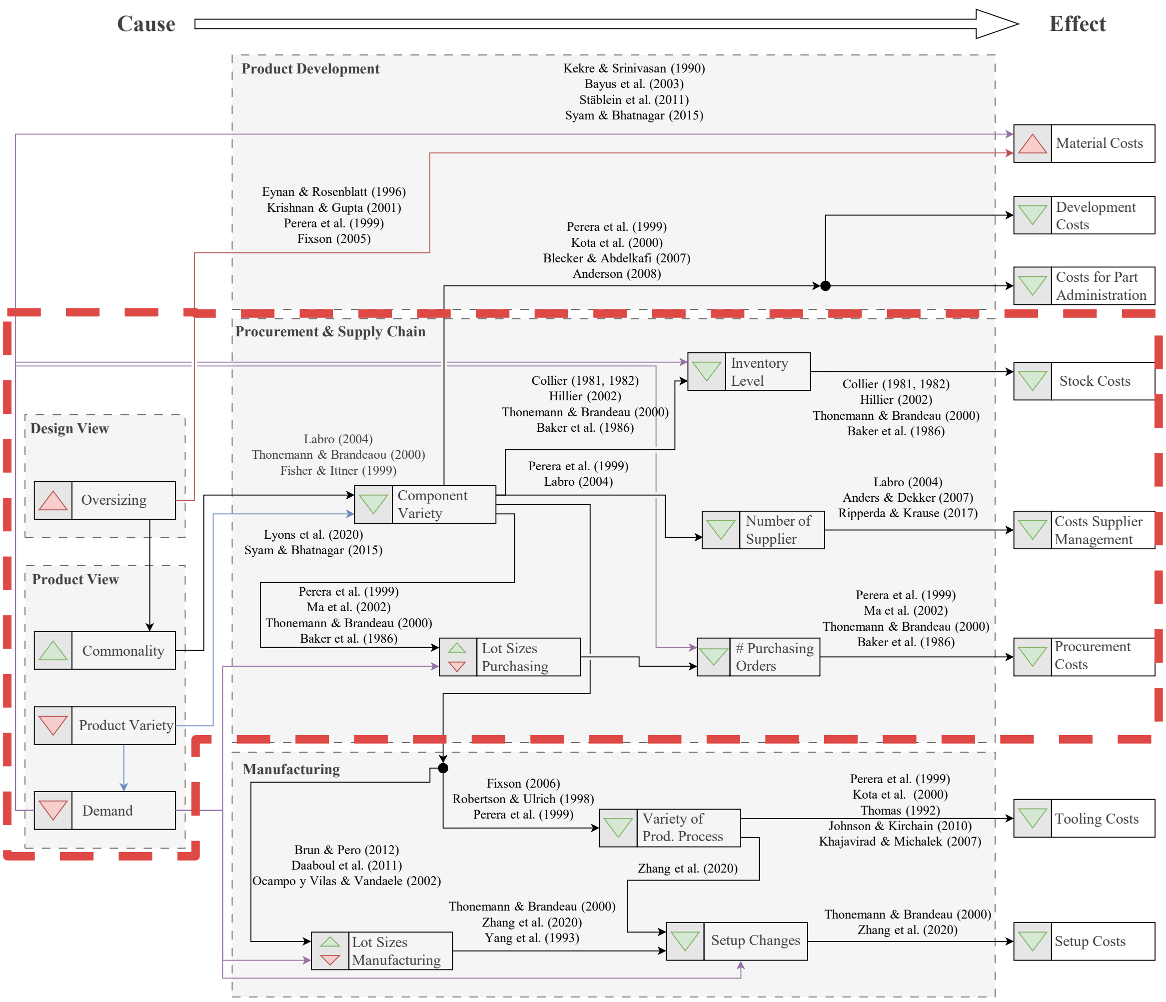
The model assumes that the combination of components and processes requires an individual tool. In the numerical experiment entries of $C_{tooling}$ are uniformly distributed. Again, a small working example is provided in thesis online appendix[70].

4.3.3 Cost Effects in Purchasing

The impact model mentions three cost effects in procurement. These complexity-induced costs are all caused by component variety, resulting from communal design and product variety. According to the impact model, decreasing component variety reduces the number of purchasing orders as the lot size increases, reduces the number of suppliers needed and, therefore, the supplier management costs, and, finally, reduces the stock needed. Figure 4.8 provides an overview of the cause-effect relationships in the purchasing and supply chain department.

[70] The interactive example can be accessed by running *utils::vignette('productionCostEffects', package = 'EAD')*

Figure 4.8:
Economic consequences of internal complexity within the purchasing department.



Management science and production economics differ between order and holding costs. While the first refers to all costs caused by placing orders at the suppliers, the latter summarizes all costs for holding inventory in the stock (Russell & Taylor, 2005). This work differs between order placing and supplier management costs as they have different causes. While supplier management costs depend on the number of suppliers, order costs depend on the number of purchasing orders. Models for each cost driver are introduced in the following subsections.

4.3.3.1 Order Costs

Costs for ordering raw materials are modeled using input from the previous drivers. This thesis assumes that purchasing and manufacturing lot sizes are equal. Therefore, components' lot sizes $N_{lot,i}$, defined in the previous section, and their demand are used. In doing so, the total number of order ($N_{order} \in \mathbb{N}^{N_{PD}}$) is defined as:

$$N_{order,i} = \left\lceil \frac{DMD_{PD,i}}{N_{lot,i}} \right\rceil \quad (4.28)$$

Each purchasing order is associated with certain costs as it requires time. Following the common assumption of fixed costs per order (e.g., Gavirneni, 2002; Thonemann & Brandeau, 2000), costs for placing an order are defined as a user-input C_{order}. This formulation assumes that the costs of each order are independent of the component and their batch size. The total order costs (TC_{order}) are then defined as:

$$TC_{order} = \sum_{i=1}^{N_{PD}} TL_i * C_{order} \tag{4.29}$$

4.3.3.2 Supplier Management Costs

Costs caused by supplier management are another driver, depending on the number of materials firms must handle. Authors note that these costs occur as firms search and acquire suppliers to find the best supplier as a trade-off between quality and price (S. W. Anderson & Dekker, 2009; Labro, 2004). Claycomb and Frankwick (2004) further note the costs of maintaining the supplier relationship. Sharing information between firm and supplier on expected component demand gains advantages for both, as Gavirneni (2002) adds. Component commonality allows for component pooling, which has pros and cons. On the one hand, overdesigned components are more complex, reducing the number of suppliers. This increases the risk in cases where vendors deliver poor quality. On the other hand, it reduces lead time uncertainty (Benton & Krajewski, 1990).

Supplier management costs are modeled as follows. The model assumes that each component is sourced from an individual supplier. If a component is used at least once, costs for the supplier management (TC_{supply}) are added to the total costs. These costs are defined as:

$$TC_{supply} = \sum_{i=1}^{N_{PD}} \zeta_i * C_{supply,i}\ ,\ \ \zeta_i = \begin{cases} 1, & \sum TCC_i > 0 \\ 0, & \sum TCC_i = 0 \end{cases} \tag{4.30}$$

$C_{supply,i}$ are the summed costs caused by supplier search, acquisition, and maintaining the relationship for a component i. As a simplification, this model assumes constant costs for each supplier, which allows for dropping the index from C_{supply}. In practice, however, costs can vary across suppliers. While some suppliers are easy to handle, firms may face lengthy contract negotiations with others, resulting in higher costs. Additionally, not all components are sourced from individual suppliers. Labro (2004) notes that firms prefer sourcing new components from existing suppliers to avoid cost-intensive search and acquisition for new ones. Supplier costs occur if a component is used by at least one product within the product mix. Total component consumption ($TCC \in \mathbb{N}^{N_{PD}}$) is calculated as:

$$TCC = DMD * P_{PD} \tag{4.31}$$

As for development and part administration costs, the model assumes that suppliers are acquired on an as-needed basis. In practice, supplier selection is a costly and time-intensive process (e.g., S. W. Anderson & Dekker, 2009), done prior to manufacturing where later demands are fuzzy.

4.3.3.3 Inventory Costs

The effect of component variety on inventory costs has been investigated by many studies over the recent decades (Baker, 1985; Baker et al., 1986; Collier, 1982; Dogramaci, 1979; Eynan & Rosenblatt, 1996; Ma et al., 2002; U. Rao, Swaminathan, & Zhang, 2004; Wan & Sanders, 2017). Studies find that increasing commonality reduces safety and cycle stock costs (e.g., Thonemann & Brandeau, 2000). While cycle stock inventory is used under normal conditions, inventory from safety stock is used when a specific inventory is not available in cycle stock anymore. Such situations are called stockouts (e.g., Baker et al., 1986). For two reasons, the model to calculate stock costs is adapted from Thonemann and Brandeau (2000). First, their study is closely related to component commonality, and second, the central concept is similar across the studies mentioned above (Baker et al., 1986; Collier, 1982). According to Thonemann and Brandeau (2000), inventory costs (TC_{inv}) are the sum of safety stock costs (TC_{safety}) and cycle stock costs (TC_{cycle}) defined as:

$$TC_{inv} = TC_{safety} + TC_{cycle}. \tag{4.32}$$

Let $C_{hold,i}$ be the costs for holding a unit of component i in stock. According to Thonemann and Brandeau (2000), cycle stock costs are than calculated as:

$$TC_{cycle} = \sum_{i=1}^{N_{PD}} C_{hold,i} * \frac{N_{lot,i}}{2} \tag{4.33}$$

Equation (4.33) assumes a linear demand for a component i. Half of the component's lot size is in stock on average. The calculation of safety stock levels is non-straightforward as it requires demand variability analyses. As a simplification, this work assumes that each component has a safety stock of one lot size.

$$TC_{safety} = \sum_{i=1}^{N_{PD}} C_{hold,i} * N_{lot,i} \tag{4.34}$$

Therefore, the total inventory costs are given as:

$$TC_{inv} = \sum_{i=1}^{N_{PD}} C_{hold,i} * \frac{3N_{lot,i}}{2} \tag{4.35}$$

Although holding costs differ across components in practice, this work assumes constant holding costs across all components. Equation (4.35) than turns into:

$$TC_{inv} = C_{hold} * \sum_{i=1}^{N_{PD}} * \frac{3N_{lot,i}}{2} \tag{4.36}$$

Since component holding costs are constant across all components, equation (4.36) is further simplified to:

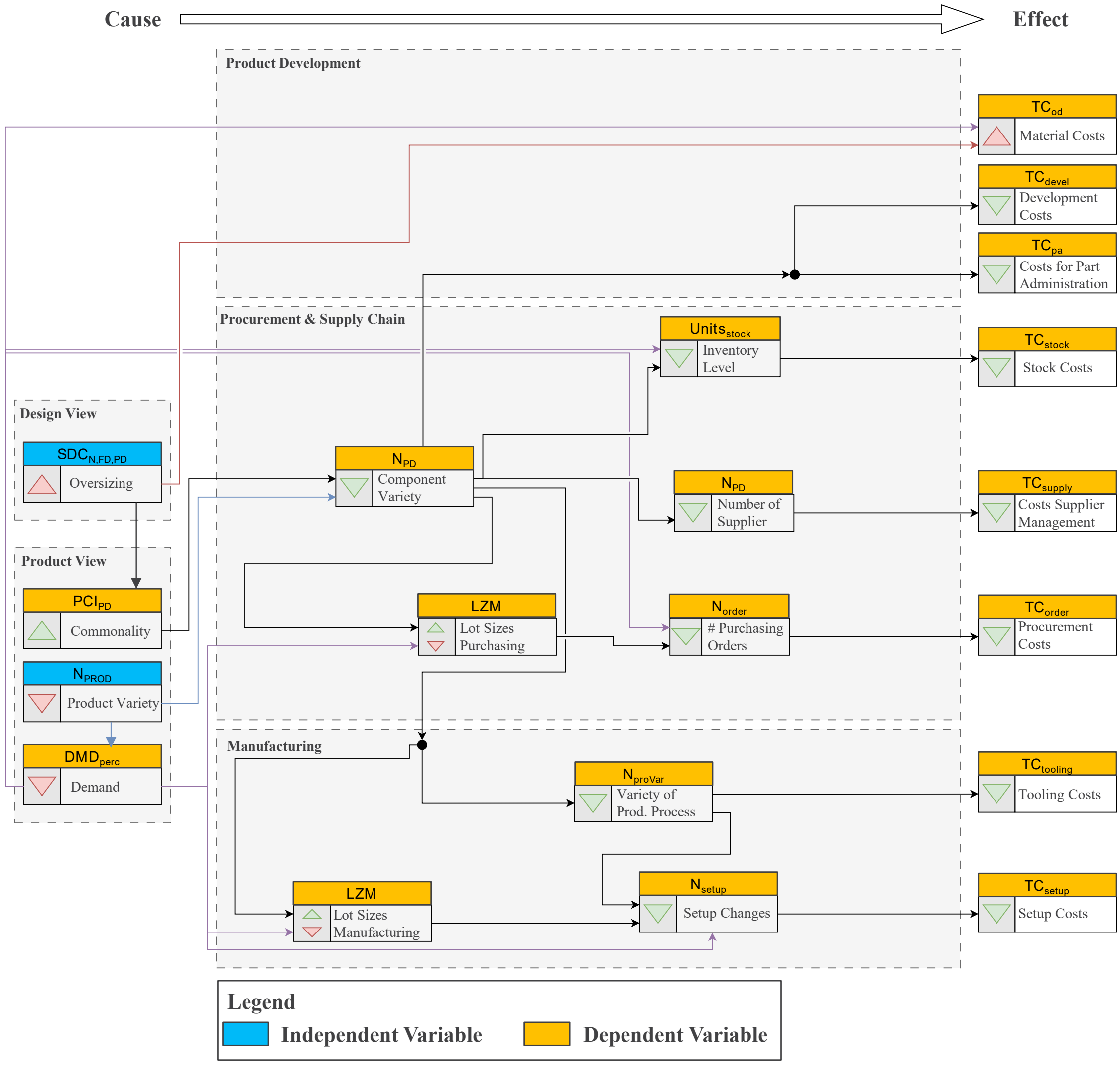
$$TC_{inv} = C_{hold} * Units_{stock} \quad (4.37)$$

4.3.4 Integration of complexity-induced Costs Models

Partial models introduced in the previous sections are integrated into the numerical EAD model. This step connects measures of internal complexity with their economic consequences and is the foundation for conducting numerical experiments in the next step. Figure 4.9 shows the impact model with variables defined in the previous section's models. These variables are separated into two independent and 18 dependent variables. Overdesign (SDC_N) and product variety (N_{PROD}) are the independent variables in this model[71]. As already argued, overdesign is a firm's reaction to offering a specific product variety cost-efficiently. This model, however, defines overdesign as an independent variable. That allows for analyzing a firm's reaction (overdesign) under different levels of product variety. Overdesign is measured by the normalized system design complexity (SDC_N) in line with findings in section 3.6. This variable reflects the coupling within the domain mapping matrix ($DMM_{FD,PD}$). Lower values indicate less coupled designs ($SDC_N \to 0$), whereas high values ($SDC_N \to 1$) strongly coupled designs. Product variety is measured via the number of products (N_{PROD}).

[71] To be precise, product variety also drives the degree of component commonality. The degree of component commonality is increased by dropping exotic products and introducing new products similar to existing ones, assuming a constant design. This work operationalizes product variety solely via the number of products for simplification, neglecting the aspect of product hetero- and homogeneity (see Buchholz, 2012).

Figure 4.9:
The operationalized impact model.



The remaining variables are influenced by at least one of the independent variables ($SDC_{N,FD,PD}, N_{PROD}$). An increased overdesign (SDC_N increases) will lead to an increased component commonality (PCI_{PD}). A fully standardized product mix where each product uses the same components is characterized by $PCI_{PD} = 1$, whereas no commonality ($PCI_{PD} = 0$) is present if products do not share any components. According to Hackl et al. (2020), decreased component variety (N_{PD}) is a result of increased component commonality. However, this model argues that overdesign directly impacts component variety since an overdesigned component replaces two individual ones. As a result of increased overdesign and reduced component variety and increases component commonality. It is further assumed that each component is sourced from an individual supplier

($N_{PD} = N_{supply}$)[72]. According to the production model in section 4.3.2, component and process variety (N_{proVar}) are positively associated.

It is further assumed that each product variant is sold at least once. A product mix with a total demand (TD) distributed across N_{PROD} distinct products is generated. DMD_{perc} then measures the proportion of actual demand ($\sum DMD$) on total demand and is defined as:

$$DMD_{perc} = \frac{\sum DMD}{TD} \quad (4.38)$$

Since each product has a non-zero demand, a change in product variety (N_{PROD}) influences DMD_{perc}. Several variables depend on the realized demand, such as the average lot size (LZM), the number of setups (N_{setup}), the number of units in stock ($Units_{stock}$), and orders (N_{order}). The right side of the impact model shows the economic consequences in terms of total costs for each effect (denoted as TC). Since N_{PD} and N_{proVar} do not depend on the total demand, costs for part administration (TC_{pa}), development (TC_{devel}), supplier management (TC_{supply}), and tooling ($TC_{tooling}$) are seen to be fixed. The remaining costs are variable as they depend on the realized demand. The following section uses the operationalized impact model to conduct numerical experiments.

4.4 Numerical Experiment

4.4.1 Data Generation

The numerical experiment aims to generate a variety of product family designs that differ in their degree of overdesign and product variety. In doing so, a three-step approach is used to generate the data set. The following pseudo-code visualizes the procedure.

```
For each EAD [i]
        Do: overdesign
                Do: increase product variety
                        increase product variety
                        calculate complexity-induced costs
                        calculate product family complexity measures
                        increase overdesign
                If full product variety: break
        If N_PD < (N_PD/2 -1 ): break
End i
```

An initial EAD is generated in the first step according to the control flow chart (Appendix A1). Each design represents the initial product family design (S_0) and is characterized by a certain structural complexity ($SDC_{N,S0}$) and degree of commonality ($PCI_{PD,S0}$). Components are stepwise overdesigned starting with S_0 according to the following procedure. For each overdesign step ($S_s,\ s = 1 \ldots S_{max}$), two components are substituted by an overdesigned one. The replacement is repeated until the number of components is

[72] Future studies, for example, can model the economic consequences of supplier commonality or standardization if the component-supplier relationships are varied.

reduced by 50%. Therefore, the overdesign loop is repeated exactly $S_{max} = \lceil N_{PD,S_0}/2 - 1 \rceil$ times[73]. Overdesign is modeled by increasing the coupling in $DMM_{FD,PD}$. Therefore, the normalized system design complexity is strictly increasing ($SDC_{N,S_s} < SDC_{N,S_{s+1}}$). Components are replaced as follows. In the first step, the component with the minimum demand is selected. Second, its most similar component[74] is selected as the second component. Using the most similar component ensures that components that already share similarities are preferred to be unified. This reflects settings where firms, for example, standardize a small and large beam instead of a small beam and a cabinet, which are characterized by completely different functional requirements. An example is used to illustrate this procedure. Let DMM_{FD,PD,S_0} a domain mapping matrix representing the initial design (S_0) defined as:

$$DMM_{FD,PD,S_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{4.39}$$

In this example, PD_2 is selected since it is assumed it has the minimum demand. Its nearest neighbor (PD_3) is calculated by using the Euclidean distance. Both components are unified by taking the elementwise maximum of the second and third columns, leading to the new domain mapping matrix in the next step (S1).

$$DMM_{FD,PD,S_1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \tag{4.40}$$

While the functional requirement FR_2 is fulfilled by components PD_2 and PD_3 in S_0, it is now solely fulfilled by the new overdesign component PD_{2*}. Since PD_3 is now substituted, its column vector is zero. The procedure is repeated once again. After two repetitions, the number of components has halved, meeting the break criteria. An interactive example is provided in the simulation models' online documentation[75].

In the third step, product variety is increased stepwise. The simulation starts with a certain number of products (N_{PROD,V_0}) and introduces $N_{PROD,step}$ new products in each variety step until all products are introduced. In maximizing profits, firms prefer to introduce high-volume products first to gain the benefits of economies of scale. Low-volume products are introduced later to address niches. However, some firms solely focus on addressing niches by offering highly customized products. For those product families, the demand vector is almost homogenous ($DMD_{T10\%} \rightarrow 0.1$), and product quantities are close to one. The increase in product variety is modeled using weighted integer sampling according to the following pseudo-code:

---

[73] $\lceil x \rceil$ represents the ceiling operator, which rounds a number to the next larger integer.

[74] Using the Euclidean distance as a similarity metric.

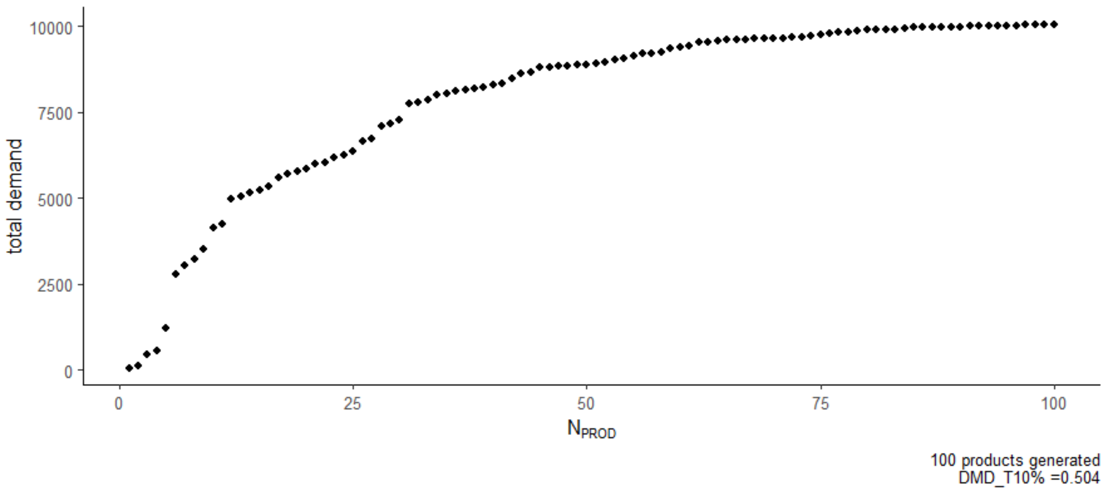
[75] The EAD library includes an interactive version. After installing the library, the example can be viewed by running *utils::vignette('developmentCostEffects',package = 'EAD')* in R.

```
number_of_products = [PROD_1,PROD_2,…,PROD_NPROD]

products_excluded = number_of_products
while: length(products_excluded) > 0
       prob = DMD[products_excluded] / sum(DMD[products_excluded])
       smp = sample one product from products_excluded with probability prob
       products_included = products_included + smp
       products_excluded = products_excluded - smp
       If full overdesign: break
end while
```

In each variety step, products from the list of excluded products are added to the actual product mix, where the introduction probability for each product is proportional to its demand. High-volume products, therefore, have a higher chance of being introduced first, whereas low-volume products are more likely to be sampled at later points. Figure 4.10 shows the increase in demand under increasing product variety as generated by the procedure described above. Complexity-induced costs are calculated for each variety step according to the models introduced in section 4.3. The output data set contains all variables, as shown in Figure 4.9.

Figure 4.10:
Total demand as a probabilistic function of product variety.



4.4.2 Design of Experiments

The design of experiments aims to create a large variety of environments in terms of product design, product mixes, cost structures, and demand distributions. In doing so, empirical boundaries, as reported in section 3.5.3, are used. While a complete overview of the design of experiments is provided in Appendix A4, this section provides a detailed explanation of variables and their boundaries. The product mix is created based on empirical data as reported in Table 3.22. The total demand is set to ($TD = 9000$) and the number of products to $N_{PROD} = 200$ as it leads to an average re-use ratio of:

$$RU = \frac{TD}{N_{PROD}} = \frac{4200}{200} = 21 \tag{4.41}$$

matching the mean value of those eight product mixes from Table 3.22. The demand distribution is varied (uniform distribution) within the bounds of $0 \leq Q_{var} \leq 1.7$, again, matching the empirical data. Things are getting more complicated for the number of products N_{PROD} and the number of functional requirements N_{FR}. Empirical data shows a ratio of sold products to possible products under free combination ($2^{N_{FR}} - 1$), which is between $1 * 10^{-20} - 1 * 10^{-117}$. This indicates that firms sell only a small proportion of theoretical possible product variants. Large models and high computational efforts are needed to create product families with such ratios. Therefore, the model sets the number of functional requirements to $N_{FR} = 30$, resulting in a ratio of realized to possible products of:

$$\frac{N_{PROD}}{2^{30} - 1} = 1.86 * 10^{-7} \qquad (4.42)$$

Product design parameters are defined within the bounds, reported in Table 3.21. In doing so, the inter-domain coupling is varied between $0 \leq SDC_N \leq 0.05$ and the intra-domain coupling within the bounds of $0 \leq DNS_{DSM} \leq 0.14$. This setup allows the creation of uncoupled ($SDC_N = 0 \wedge DNS_{DSM} = 0$) and more coupled product family designs ($SDC_N = 0.1 \wedge DNS_{DSM} = 0.14$). During the simulation, the degree of coupling is increased since components are overdesigned, leading to higher values for SDC_N and DNS_{DSM} in later analyses.

Total costs (TC) are the sum of fixed (TC_{fix}) and variable (TC_{var}) costs (see section 2.6.1), where R_{fix} defines the proportion of fixed costs to total costs for the initial design (no overdesign, S_0). Table 4.6 provides an example of the cost structure used in this experiment. Fixed costs are further separated into development (TC_{dvl}), part management (TC_{pa}), supplier management costs (TC_{supply}), and costs for the development, purchasing, and manufacturing of tolling equipment ($TC_{tooling}$). Those costs are defined as ratios ($R_{dvl}, R_{pa}, R_{supply}, R_{tooling}$) on fixed costs for the initial design (no overdesign, S_0) according:

$$TC * R_{fix} = TC_{fix} * (R_{dvl} + R_{pa} + R_{supply} + R_{tooling} + R_{fix,nonCC}) \qquad (4.43)$$

$$with\ TC_{fix} = TC_{dvl} + TC_{pa} + TC_{supply} + TC_{tooling} + TC_{fix,nonCC} \qquad (4.44)$$

$$and\ 0 \leq R_{dvl}, R_{pa}, R_{supply}, R_{tooling}, R_{fix,nonCC} \leq 1 \qquad (4.45)$$

$TC_{fix,nonCC}$ represents the remaining fixed costs unaffected changes of the internal complexity. The same logic is used for variable costs. These costs are separated into costs caused by purchasing orders (TC_{order}, R_{order}), setup costs (TC_{setup}, R_{setup}), and inventory holding costs (TC_{hold}, R_{hold}). The remaining unassigned costs are non-complexity-induced variable costs ($TC_{var,nonCC}$).

Table 4.6:
Exemplarily cost structure. Values are randomly sampled according to the boundaries defined by the design of experiments (see Appendix A4).

				Cost Type	
Description	**Variable**	**Value**	**Total Costs**	**variable**	**fix**
Total non-complexity-induced costs	TC		1,000,000		
Proportion of fixed costs	R_{fix}	30%			
variable Costs	TC_{var}			700,000	
fixed Costs	TC_{fix}				300,000
Proportion of development costs on fixed costs	R_{dvl}	40%			120,000
Proportion of part management costs on fixed costs	R_{pa}	7%			21,000
Proportion of tooling costs on fixed costs	$R_{tooling}$	10%			30,000
Proportion of supplier management costs on fixed costs	R_{supply}	7%			21,000
Remaining fixed costs not affected by complexity cost driver	$TC_{fix,nonCC}$				108,000
Proportion of order costs on variable costs	R_{order}	5%		35,000	
Proportion of holding costs on variable costs	R_{hold}	7%		49,000	
Proportion of total setup costs on total variable costs	R_{setup}	20%		140,000	
Remaining variable costs not affected by complexity cost driver	$TC_{var,nonCC}$			476,000	

Except for $TC_{var,nonCC}$ and $TC_{fix,nonCC}$, these cost types are used as inputs to calculate the cost rates (C) for each complexity-induced cost effect under the initial design (S_0). The following examples demonstrates the calculation of the order cost rate (C_{order}).

$$C_{order} = \frac{R_{order} * TC_{var}}{N_{order,S_0}} \quad (4.46)$$

The same is done for the remaining complexity cost types, as introduced in section 4.3. Cost rates are calculated for the initial design state (S_0) and remain constant when iterating over product variety and overdesign. Although cost rates remain constant, variable complexity-induced costs change since the cost driver (e.g., N_{order} for the order costs) change. The total complexity costs are then given as:

$$TC_{CC} = TC_{od} + TC_{dvl} + TC_{pa} + TC_{tooling} + TC_{supply} + TC_{order} + TC_{hold} + TC_{setup} \quad (4.47)$$

TC_{od} are the total additional material costs caused by component overdesign. These costs do not have a distinct cost rate since their value depends on component material costs. For the initial design (S_0), $TC_{od} = 0$ since no overdesign is assumed. Under increasing overdesign, these costs begin to rise. The same applies to increasing product variety as it increases the chance of an overdesigned component being used in more products. The number of runs is estimated as $N_{RUN} = 5{,}000$. At this number of repetitions, the correlations between variables are stable, and effects due to small sample sizes for specific subgroups within the data are disappeared.

4.5 Analyzing complexity-induced Cost Effects

The generated data set includes 1.029.910 observations, representing 4986 unique product family designs under ten different levels of product variety ($N_{PROD} = 20,40,\ldots,200$)[76]. The first part (section 4.5.1) analyzes the individual complexity-induced cost effects, while the second part (section 4.5.2) the total cost effects. Designs are randomly created and, therefore, have different absolute variable values. For a fair comparison of designs among each other, relative differences rather than absolute differences are reported. This is done by normalizing each variable (X) as follows:

$$\delta X_{S_x} = \frac{X_{S_x} - X_{S_0}}{X_{S_0}} \tag{4.48}$$

Normalized variables (δX_{S_x}) measure the percentage differences between the current absolute value (X_{S_x}) and its absolute value for the initial design (X_{S_0}). From a complexity theoretical perspective, X_{S_0} represents the reference simplicity for each product family and X_{S_x} the actual complexity. For example, component commonality is normalized as follows:

$$\delta\ PCI_{PD,S_x} = \frac{PCI_{PD,S_x} - PCI_{PD,S_0}}{PCI_{PD,S_0}} \tag{4.49}$$

A value of $\delta\ PCI_{PD,Sx} = .3$ indicates a component commonality increase of about 30% in design step S_x compared to the initial design (S_0). This procedure is repeated for all variables except for TC_{od}, where normalization is not required since $TC_{od,S_0} = 0$.

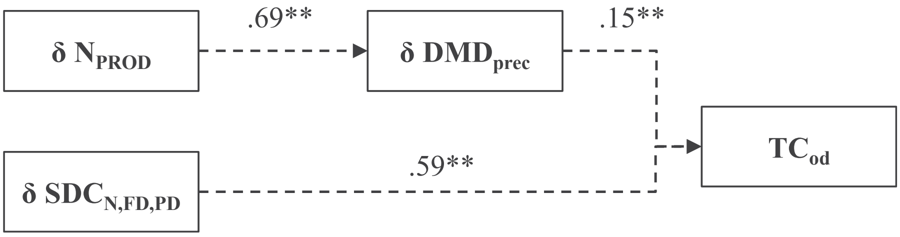
4.5.1 Individual Cost Effects

4.5.1.1 Costs of Overdesign

Two variables drive the increase in material costs caused by component overdesign ($\delta\ TC_{od}$). First, components are substituted by increasing the coupling in $DMM_{FD,PD}$, measured by $\delta SDC_{N,FD,PD}$. Therefore, a positive association between $\delta SDC_{N,FD,PD}$ and δTC_{od} is expected. Since material costs are variable, the proportion of realized demand to total demand (δDMD_{perc}) is a second driver. An increased demand increases the chance of reusing overdesign components in more products, leading to higher material costs. Therefore, a positive association between the number of products (δN_{PROD}) and δDMD_{perc} is expected. Figure 4.11 shows the cause-effect paths where coefficients represent Pearson correlation coefficients. Model statistics are reported in Table 4.7.

[76] For the remaining 17 designs, the EAD generation was aborted since no valid EAD design was found after a certain number of iterations.

Figure 4.11:
Cause-effect relationships for the additional material costs caused by overdesign. Values along the path represent Pearson correlation coefficients.



Note. ** indicates *p* < .01

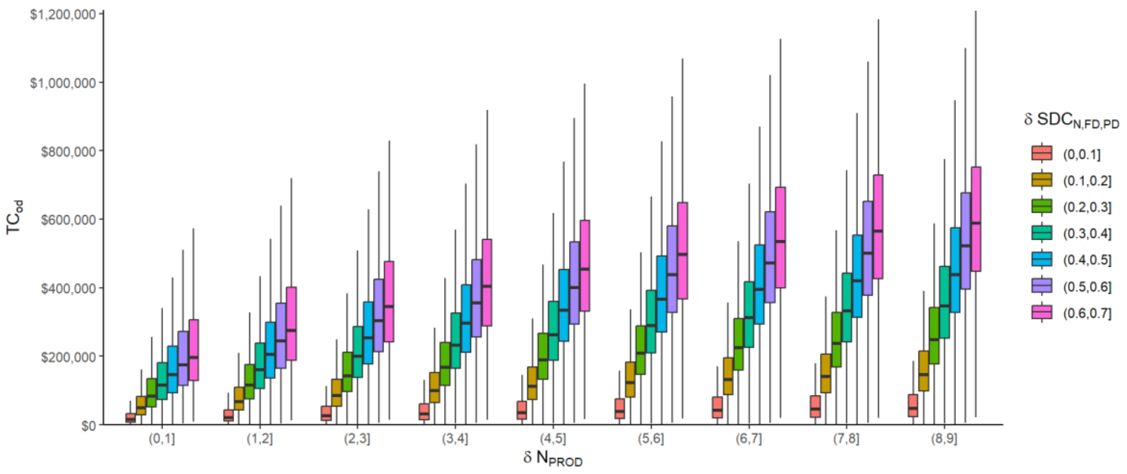
Table 4.7:
Pearson correlation coefficients, showing the association between design variables and additional material costs caused by overdesign.

Variable	δ $SDC_{N,FD,PD}$	δ DMD_{perc}	δ N_{PROD}	TC_{od}
δ $SDC_{N,FD,PD}$	1.00			.59
δ DMD_{perc}		1.00	.69	.15
δ N_{PROD}			1.00	.31
TC_{od}				1.00

Note. All values shown are significant at p<.01

The model confirms the positive association between overdesign and material costs ($c = .59, p < .01$), as well as the positive association between demand and overdesign costs ($c = .15, p < .01$). The latter is further driven by product variety ($c = .69, p < .01$). The positive effect size of $\delta SDC_{N,FD,PD}$ is expected and in line with the hypothesized directions as stated in Hackl et al.'s (2020) impact model. However, results show that questions of overdesign cannot be reduced to a coupling problem solely, and demand needs to be considered. Nevertheless, the effect sizes indicate that intra-domain coupling is more important as the demand. Figure 4.12 provides a graphical representation of total overdesign costs depending on product variety and the degree of overdesign.

Figure 4.12:
The increase of total overdesign costs under different levels of overdesign and product variety.



Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

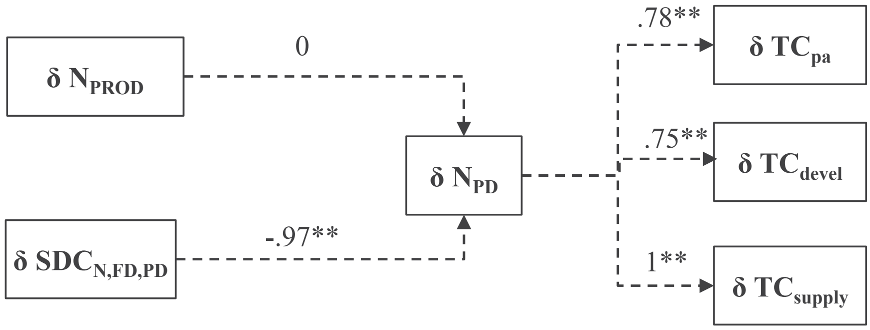
The graph confirms the positive association between overdesign ($\delta SDC_{N,FD,PD}$) and additional material costs (δTC_{od}). While differences across designs (e.g., $\delta SDC_{N,FD,PD} = (0,0.1]$ vs. $\delta SDC_{N,FD,PD} = (0.6,0.7]$) are small under a low level of product variety, they increase with increasing product variety. This steeper increase is because, under overdesign, the chance of reusing already overdesigned components increases. The results confirm the conceptual impact model and indicate the downside of increased overdesign. The remaining cost types all show cost-reducing effects with an increasing level of overdesign.

4.5.1.2 Development, Part Management and Supplier Management Costs

Component variety (δN_{PD}) is the cost driver for development (δTC_{devel}), part management (δTC_{pa}), and supplier management costs (δTC_{supply})[77]. Therefore, these effects are analyzed using one model. Engineering can influence the number of components by changing the degree of component overdesign ($\delta SDC_{N,FD,PD}$). A high degree of overdesign means that each component fulfills several functional requirements simultaneously, resulting in fewer components needed to realize a product mix. Therefore, a negative association between $\delta SDC_{N,FD,PD}$ and δN_{PD} is expected. Product variety (δN_{PROD}) is assumed to be a second cause of increasing costs. Dropping all products using a specific component reduces component variety (δN_{PD}). Conversely, it can increase component variety if new products require new components. Figure 4.13 shows the causal model, and Table 4.8 reports model statistics.

[77] This model assumes that each component is sourced from an individual supplier; therefore, the number of suppliers equals the number of components.

Figure 4.13:
Cause-effect relationships, driving the total development, part administration, and supplier management costs.



Note. ** indicates p < .01

Table 4.8:
Pearson correlation coefficients to estimate the effects driving development, part administration and supplier management costs.

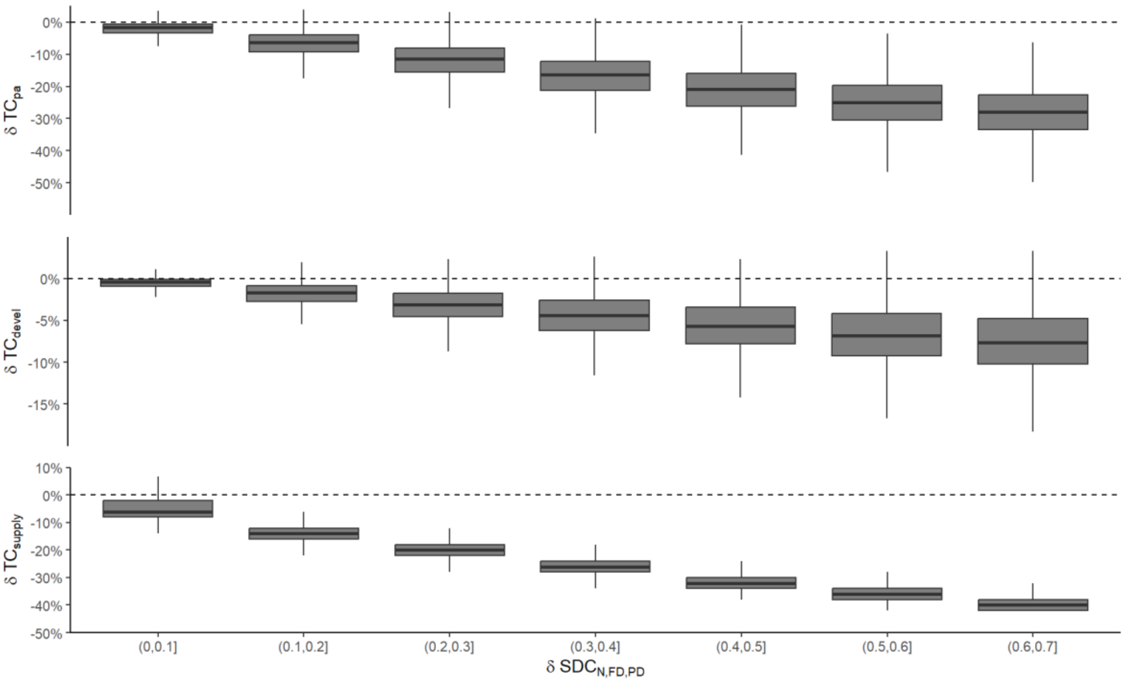
Variable	δ $SDC_{N,FD,PD}$	δ PCI_{PD}	δ N_{PROD}	δ N_{PD}	δ TC_{devel}	δ TC_{pa}	δ TC_{supply}
δ $SDC_{N,FD,PD}$	1.00	.69		-.97	-.72	-.76	-.96
δ PCI_{PD}		1.00		-.66	-.44	-.46	-.66
N_{PROD}			1.00				
N_{PD}				1.00	.75	.78	1.00
TC_{devel}					1.00	.66	.75
TC_{pa}						1.00	.78
TC_{supply}							1.00

Note. All values shown are significant at p<.01

As assumed, increasing overdesign has a significant negative association with component variety ($c = -.97, p < .01$). Since component variety is a cost driver for the development, supplier management, and part management costs, a positive association is observed for all cost types ($c_{pa} = .78$; $c_{supply} = 1.$; $c_{devel} = .75, p < .01$). Supplier management costs show a perfect association with component variety since the model assumes that each component is sourced from one and only one supplier. In practice, however, firms have several reasons for multi-scouring components (e.g., risk pooling), reducing the correlation between supplier management costs and component variety. In sum, overdesign shows a strong negative association with these costs (e.g., $c_{devel} = -.72,, p. < 01$).

Interestingly, product variety does not affect the number of components ($c = .0, p > .01$). An explanation is that under configurable product families, new products are more likely the result of combining existing components instead of introducing new ones. A detailed analysis shows some cases where an increased product variety leads to an increase in component variety. However, this happens only if new products introduce new functional requirements. Of course, this might be a caveat of the existing EAD model as it represents only linear relationships and does not reflect that combining new functional requirements may result in new components. Nevertheless, under these assumptions, product variety does not affect component variety ($\delta\ N_{PD}$) and, therefore, neither development, part management, nor supplier management costs. Figure 4.14 visualizes the decrease in development, part administration, and supplier management costs.

Figure 4.14:
The decrease of total development, part management, and supplier management costs with an increasing degree of overdesign.



Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

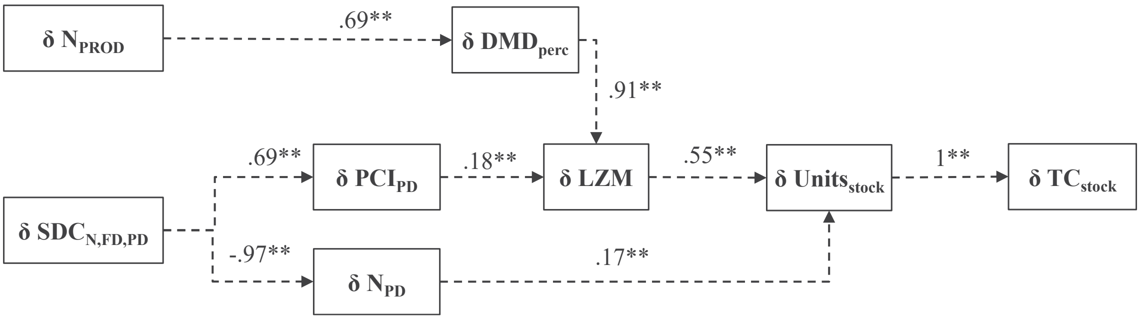
4.5.1.3 Inventory Holding and Purchasing Order Costs

While the previous complexity-induced cost effects were straightforward, the cause-effect relationships leading to inventory holding (δTC_{stock}) are more complex. Inventory holding costs are driven by the number of units in stock ($\delta Units_{stock}$), which depends on the number of distinct components (δN_{PD}) and the average lot size (δLZM). Remember, the model assumes that firms hold 1.5 lots in stock for each component. Since the model uses an economic order model:

$$\delta LZM \sim \sqrt{\delta DMD_{PD}} \quad (4.50)$$

where δDMD_{PD} are the component demand rates. Therefore, the average lot size is driven by two variables. First, an increased demand (δDMD_{perc}) increases the average component demand rates, leading to an increase of δLZM. Second, component commonality (δPCI_{PD}) increases the average demand rates due to component reuse. Figure 4.15 shows the cause-effect diagram for the inventory holding costs.

Figure 4.15.
Cause-effect relationships driving the inventory holding costs.



Note. ** indicates $p < .01$

The correlation results (Table 4.9) show a positive association between overdesign and component commonality ($c = .69, p < .01$). In contrast to Hackl et al. (2020), this work argues that there is no causal relation between component commonality and the number of components. Component commonality and component variety are effects of an increased/decreased overdesign rather than causing each other. Component commonality, for example, does not necessarily reduce the number of components as it depends also on the product mix. Adding products that share components with already existing products will increase component commonality. Although results show a medium negative correlation between δPCI_{PD} and δN_{PD} ($c = -.66, p < .01$), there is no causality between both variables. This contrasts with existing studies (e.g., Baker et al., 1986), which define component commonality as a dependent variable resulting in a causal relation, whereas this work defines overdesign as an independent variable influencing component variety and commonality.

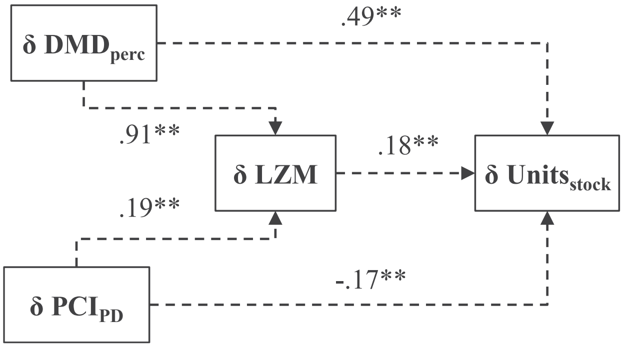
Table 4.9:
Pearson correlation coefficients to estimate the effects on total inventory holding costs.

Variable	δ $SDC_{N,FD,PD}$	δ DMD_{perc}	δ PCI_{PD}	δ N_{PROD}	δ N_{PD}	δ LZM	δ $Units_{stock}$	δ TC_{stock}
δ $SDC_{N,FD,PD}$	1.00		.69		-.97	.31	-.17	-.17
δ DMD_{perc}		1.00		.69		.91	.65	.65
δ PCI_{PD}			1.00		-.66	.18	-.13	-.13
δ N_{PROD}				1.00		.67	.46	.46
δ N_{PD}					1.00	-.32	.17	.17
δ LZM						1.00	.55	.55
δ $Units_{stock}$							1.00	1.00
δ TC_{stock}								1.00

Note. All values displayed are significant at p<0.01

As expected, the average lot sizes are driven by the total demand ($c = .91, p < .01$) and component commonality ($c = .18, p < .01$). While an increased product demand affects the demand rates of many components, an increase in component commonality is limited to some components. Therefore, large differences in correlation coefficients are observed. Increased lot sizes further lead to an increased number of units in stock ($c = .55, p < .01$). Since the average lot sizes are driven by total demand and component commonality, the observed correlation is a composite effect consisting of a demand and commonality effect. For a further investigation of these effects, a path analysis using a structural equation model is conducted (Figure 4.16).

Figure 4.16:
Path model to investigate the direct and indirect effects of total demand and component commonality on inventory levels.



Note. Values indicate standardized regression effect sizes. They are significant at p<0.01

The test statistics (Table 4.10) indicate a weaker association ($\beta = .18, p < .01$) between average lot sizes and inventory levels than the correlation analysis suggests. The reason is the existence of indirect effects. While the effect of demand via lot sizes is small ($\beta = .16, p < .01$), a medium direct effect is observed ($\beta = .49, p < .01$). Even smaller is the indirect effect of component commonality on inventory levels ($\beta = .03, p < .01$). However, component commonality shows a negative direct effect on inventory levels ($\beta = -.17, p < .01$), leading to a total negative effect estimated as $\beta = -.13$ $(p < .01)$. The analysis shows the composite effects of component commonality driving lot sizes and inventory levels. Additionally, it confirms the inventory-reducing effect of increased component commonality (Baker et al., 1986; Collier, 1982; Dogramaci, 1979; Hillier, 2002; Thonemann & Brandeau, 2000). For example, Baker et al. (1986) show that the total number of units in stock reduces under increased commonality. This effect occurs since the safety stock level for an overdesigned component (PD_*) is smaller compared to those of the individual ones (PD_1, PD_2)[78].

[78] Property 2 in Baker et al. (1986). See also: Perera et al. (1999)

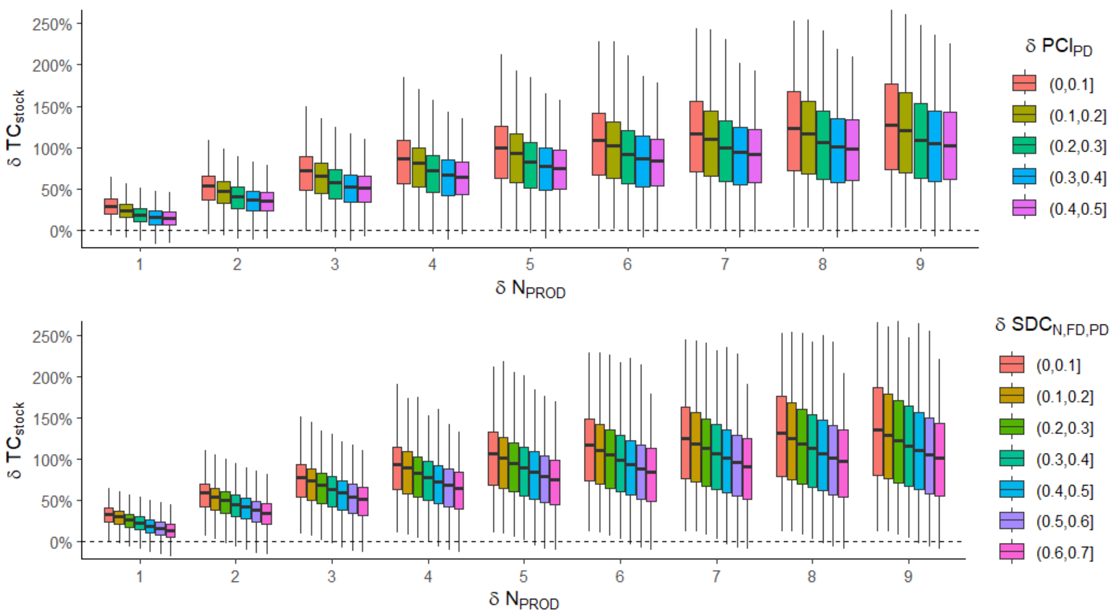
Table 4.10:
Regression betas for the path analysis.

Predictor	Outcome	*b*	*z-value*	*p-value*
	Regression			
δ PCI_{PD}	δ LZM	.18	491.40	<.01
δ DMD_{perc}	δ LZM	.91	2468.46	<.01
δ PCI_{PD}		-.17	-567.31	<.01
δ LZM	δ $Units_{stock}$	.18	247.68	<.01
δ DMD_{perc}		.49	843.93	<.01
	Indirect Effect			
δ PCI_{PD} * δ LZM	δ $Units_{stock}$	.03	221.17	<.01
δ DMD_{perc} * δ LZM		.16	246.44	<.01
	Total Effect			
δ PCI_{PD} + δ PCI_{PD} * δ LZM	δ $Units_{stock}$	-.13	-495.83	<.01
δ DMD_{perc} * δ LZM + δ DMD_{perc}		.65	2166.22	<.01

Note. CFI = 1; RMSEA = 0 and SRMR = 0, indicating a good model fit according to L. Hu and Bentler (1999); $R^2_{LZM} = 86.1\%$; $R^2_{Unitsstick} = 92.8\%$;

Figure 4.17 visualizes the inventory costs for different levels of component commonality and product variety (upper panel). Under all levels of product variety, increased component commonality leads to lower inventory costs. For example, firms can increase their product variety (e.g., $\delta N_{PROD,S0} = 6, \delta N_{PROD,S1} = 7$) while reducing the inventory costs ($\delta TC_{stock,S0} = 1.04, \delta TC_{stock,S1} = 0.96$) if they increase the level of component commonality (via overdesign) by $20\% - 30\%$ ($\delta PCI_{PD,S0} = 0, \delta PCI_{PD,S1} = (0.2,0.3]$). Nevertheless, they must consider the cost-increasing effects of overdesign, such as increased material costs. However, the figure highlights the power of overdesign and component commonality to be used as a lever for providing product variety cost-efficiently.

Figure 4.17:
Total inventory costs for different levels of component commonality (upper panel) and different degrees of overdesign (lower panel).

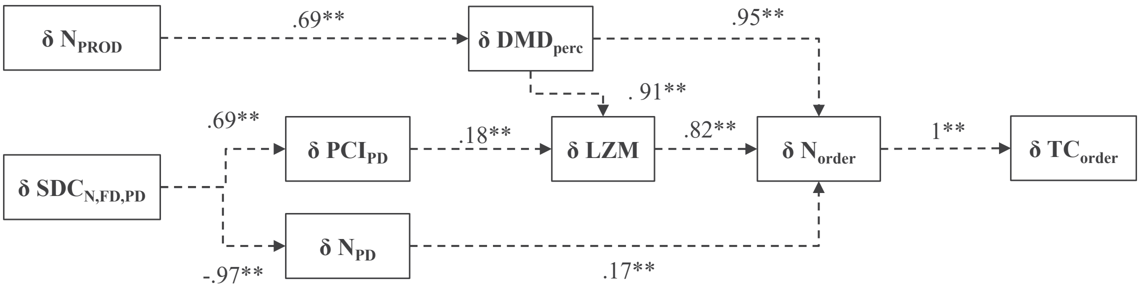


Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

Since a lot-per-lot order model is assumed, the total number of orders shows an almost similar cause-effect diagram (Figure 4.18). The only difference is a direct effect association between demand and the number of purchasing orders. Correlation results (Table

4.11) indicate a strong correlation between both variables ($c = .95, p < .01$), while only a medium correlation is reported for the association between demand and inventory levels. All remaining coefficients stay constant compared to the inventory model.

Figure 4.18:
Causes of total order cost. Numbers along the path represent Pearson correlation coefficients.



Note. ** indicates $p < .01$

A second path analysis, according to the model in Figure 4.16, estimates the direct effect of demand on purchasing orders as $\beta = .79$ ($p < .01$), leading to a total effect of $\beta = .95$ ($p < .01$)[79]. Due to the increased effect sizes of total demand, product variety shows a slightly higher correlation on purchasing orders ($c = .66, p < .01$) than on inventory levels ($c = .46, p < .01$).

Table 4.11:
Pearson correlation coefficients for the analysis of the total order costs

Variable	δ $SDC_{N,FD,PD}$	δ DMD_{perc}	δ PCI_{PD}	δ N_{PROD}	δ N_{PD}	δ LZM	δ N_{order}	δ TC_{order}
δ $SDC_{N,FD,PD}$	1.00		.69		-.97	.31	-.17	-.17
δ DMD_{perc}		1.00		.69		.91	.95	.95
δ PCI_{PD}			1.00		-.66	.18	-.12	-.12
δ N_{PROD}				1.00		.64	.66	.66
δ N_{PD}					1.00	-.32	.17	.17
δ LZM						1.00	.82	.82
δ N_{order}							1.00	1.00
δ TC_{order}								1.00

Note. All values displayed are significant at p<.01

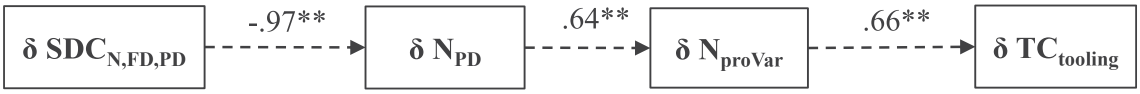
4.5.1.4 Tooling Costs

Tooling costs occur for each process variant. A process variant is defined as the combination of component and process. Therefore, the total tooling costs ($\delta TC_{tooling}$) are driven by process variety (δN_{proVar}), which is caused by component variety (δN_{PD}). Since tooling costs are fixed, and tooling equipment is required only at the component level rather than the assembly level, these costs are independent of product variety. Figure 4.19 shows the cause-effect relationships driving the tooling costs. Pearson coefficients are reported in Table 4.12.

[79] Model statistics are available upon request. The model shows the following fit criteria $CFI = 1, RMSEA =, SRMR = 0$

Figure 4.19:

The causes of total tooling costs. Numbers along the path represent Pearson correlation coefficients.



Note. ** indicates p < .01.

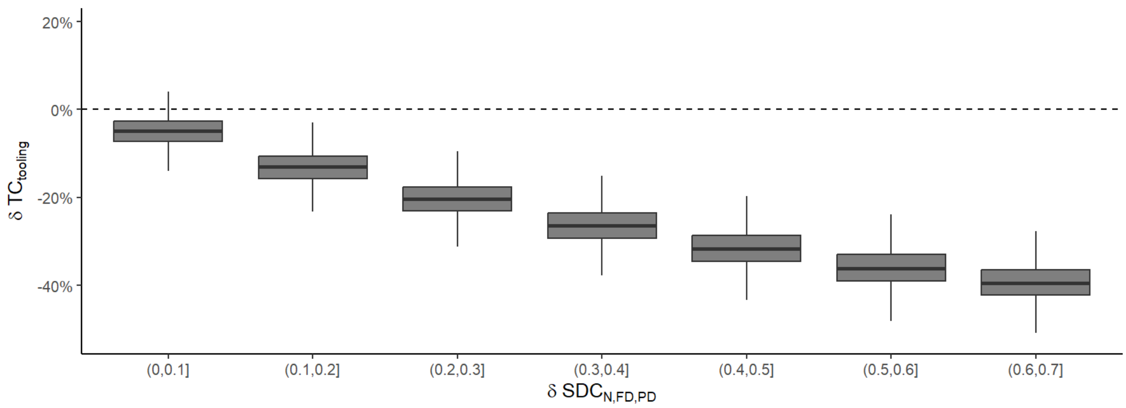
The analysis confirms the positive association between component and process variety ($c = .64, p < .01$), as reported in the literature (e.g., M. D. Johnson & Kirchain, 2010; Kota et al., 2000; Perera et al., 1999). Process variety is further positively associated with tooling costs ($c = .66, p < .01$). In sum, a medium negative association between overdesign and total tooling costs is observed ($c = -.44, p < .01$). Figure 4.20 illustrates the negative relationship, highlighting the cost-reducing effect of overdesign.

Table 4.12:
Pearson correlation coefficients for design measures with the total tooling costs

Variable	$\delta\, SDC_{N,FD,PD}$	$\delta\, N_{PD}$	$\delta\, N_{proVar}$	$\delta\, TC_{tooling}$
$\delta\, SDC_{N,FD,PD}$	1.00	-.97	-.61	-.44
$\delta\, N_{PD}$		1.00	.64	.98
$\delta\, N_{proVar}$			1.00	.66
$\delta\, TC_{tooling}$				1.00

Note. All values displayed are significant at p<.01

Figure 4.20:
A negative association between overdesign and total tooling costs is observed.

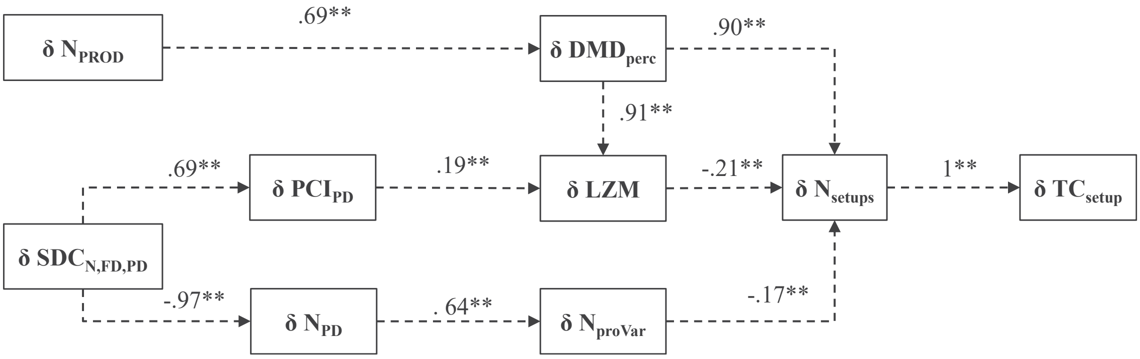


Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

4.5.1.5 Setup Costs

The total setup costs (δTC_{setup}) are driven by the total number of setups $\left(\delta N_{setups}\right)$ and are investigated as the last complexity-induced cost driver. According to the discussion in section 4.3.2.1, the total number of setups depends on several factors, such as the lot size (δLZM), the total demand (δDMD_{perc}), and process variety (δN_{proVar}). Figure 4.21 summarizes the relationships with the corresponding Pearson correlation coefficients.

Figure 4.21:
Causes of total setup costs. Numeric values along the path represent the Pearson correlation coefficients.



Note. ** indicates *p* < .01.

Pearson correlation coefficients are reported in Table 4.13. As assumed, demand and product variety are positively associated with the number of setups ($c = .68; c = .90, p < .01$). It is further observed that increasing lot sizes (δLZM) reduces the number of setups ($c = -.21, p < .01$). While these relationships confirm Hackl et al.'s (2020) impact model, the correlation analysis indicates opposite effect sizes for the effect of component commonality($c = .15, p < .01$), overdesign ($c = .29, p < .01$), component variety ($c = -.30$), and process variety ($c = -.17, p < .01$) on the number of setups. This is unexpected since the literature, for example, notes a negative association between component commonality and the number of setups.

Table 4.13:
Pearson correlation coefficients for design measures with the total setup costs.

Variable	δ $SDC_{N,FD,PD}$	δ DMD_{perc}	δ PCI_{PD}	δ N_{PROD}	δ N_{PD}	δ LZM	δ N_{proVar}	δ N_{setups}	δ TC_{setup}
δ $SDC_{N,FD,PD}$	1.00		.69		-.96	.31	-.61	.29	.29
δ DMD_{perc}		1.00		.69		.91		.90	.90
δ PCI_{PD}			1.00		-.66	.19	-.37	.15	.15
δ N_{PROD}				1.00		.67		.68	.68
δ N_{PD}					1.00	-.32	.64	-.30	-.30
δ LZM						1.00	-.21	.79	.79
δ N_{proVar}							1.00	-.17	-.17
δ N_{setups}								1.00	1.00
δ TC_{setup}									1.00

Note. All values displayed are significant at p<.01

To investigate these effects in more detail, Figure 4.22 shows two designs. Based on an initial design ($S0$), component PD_1 and PD_2 are replaced by an overdesign component (PD_{2*}) in design $S1$. To fulfill the FR_1, PD_{2*} now requires the first process (PrD_1) where all remaining variables remain unchanged.

Figure 4.22:
Two designs (S0, S1) to demonstrate the effect of increased overdesign (and component commonality) on the total number of tasks to perform.

S0

	FR_1	FR_2	FR_3	PD_1	PD_2	PD_3	PD_4	PrD_1	PrD_2	PrD_3	PrD_4	DMD
FR_1				1	0	0	0					
FR_2				0	1	1	0					
FR_3				0	0	1	1					
PD_1				0	0	0	0	1	0	0	0	
PD_2				0	0	0	0	0	1	1	0	
PD_3				0	0	0	0	1	2	1	0	
PD_4				0	0	0	0	1	0	0	1	
P1	1	0	0	1	0	0	0	1	0	0	0	10
P2	0	1	0	0	1	1	0	1	3	2	0	50
P3	0	0	1	0	0	1	1	2	2	1	1	100
P4	0	1	1	0	1	2	1	3	5	3	1	30

S1

	FR_1	FR_2	FR_3	PD_1	PD_2*	PD_3	PD_4	PrD_1	PrD_2	PrD_3	PrD_4	DMD
FR_1				0	1	0	0					
FR_2				0	1	1	0					
FR_3				0	0	1	1					
PD_1				0	0	0	0	0	0	0	0	
PD_2*				0	0	0	0	1	1	1	0	
PD_3				0	0	0	0	1	2	1	0	
PD_4				0	0	0	0	1	0	0	1	
P1	1	0	0	0	1	0	0	1	1	1	0	10
P2	0	1	0	0	1	1	0	2	3	2	0	50
P3	0	0	1	0	0	1	1	2	2	1	1	100
P4	0	1	1	0	1	2	1	4	5	3	1	30

According to the defined model (section 4.3.2.1), component demand is estimated as:

$$DMD_{PD,S0} = (10 \quad 80 \quad 210 \quad 130) \tag{4.51}$$

$$DMD_{PD,S1} = (0 \quad 90 \quad 210 \quad 130) \tag{4.52}$$

With an assumed ratio of component order to holding costs of $R = 3$[80], component lot sizes are calculated as:

$$LZM_{S0} = (8 \quad 22 \quad 36 \quad 28) \tag{4.53}$$

$$LZM_{S1} = (0 \quad 23 \quad 36 \quad 28) \tag{4.54}$$

As suggested by the literature, increasing overdesign increases the average lot sizes due to component demand-pooling. Remember, since component PD_1 is not available in $S1$, zero entries are not used for calculating the average lot size. Based on these values, the task matrices (TM), defining the number of necessary tasks (entries) for each process (columns) and component (rows), are estimated as:

$$TM_{S0} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 4 & 4 & 0 \\ 6 & 6 & 6 & 0 \\ 5 & 0 & 0 & 5 \end{pmatrix}, TM_{S1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 4 & 4 & 4 & 0 \\ 6 & 6 & 6 & 0 \\ 5 & 0 & 0 & 5 \end{pmatrix} \tag{4.55}$$

Under the initial design ($S0$), process one has $\sum TM_{S0,i,1} = 13$ tasks. By increasing the overdesign, the remaining 80 units of component PD_{2*} now require PrD_1, leading to an increased number of tasks ($\sum TM_{S1,i,1} = 15$) under scenario $S1$. In this example, increased overdesign increased the number of manufacturing tasks. However, if PD_{2*} and PD_3 are replaced by another overdesigned component in $S2$, the task matrix turns into:

$$TM_{S2} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 7 & 7 & 7 & 0 \\ 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 5 \end{pmatrix} \tag{4.56}$$

[80] The actual value order to holding costs is not relevant as the qualitative results remain the same.

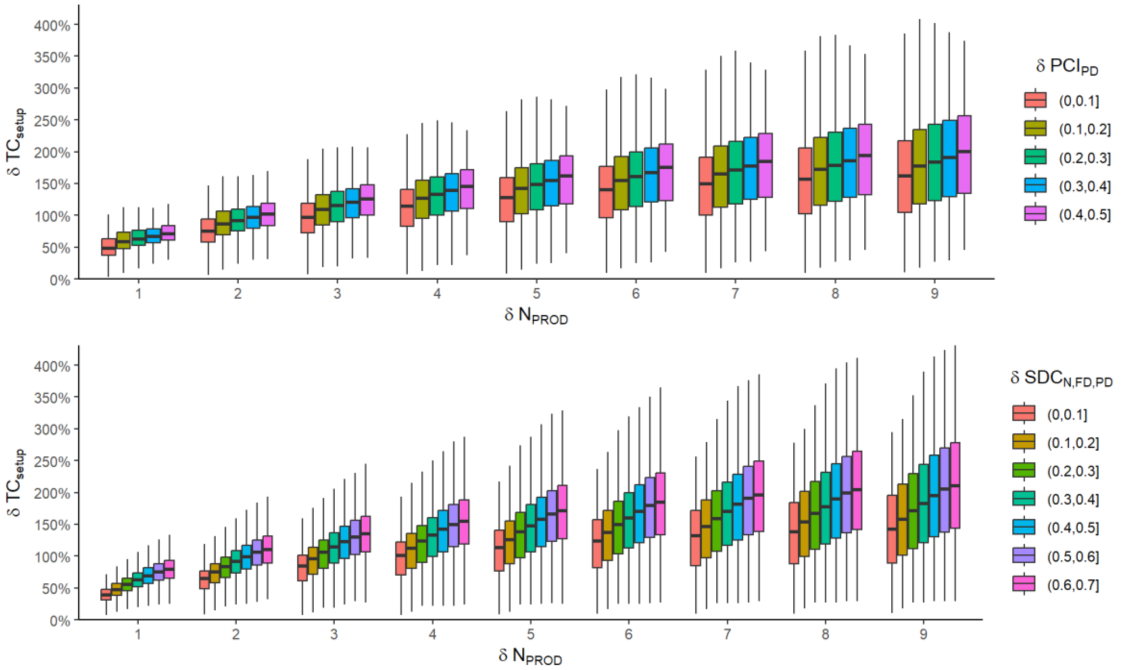
Under scenario $S2$, the number of tasks for process one is reduced ($\sum TM_{S2,i,1} = 12$). Table 4.14 summarizes the design measures and the total number of production tasks for each design step. The example demonstrates that increasing overdesign does not necessarily reduce the number of production tasks (and, therefore, setups). A reduction is only achieved if an overdesigned component replaces components, sharing at least some processes (e.g., as in $S2$). This finding is crucial as it indicates a much more complex cause-effect relationship, as suggested by the impact model. It further highlights that questions of component overdesign cannot be answered without process domain considerations.

Table 4.14:
Design variables and the number of production tasks to perform.

Design	$SDC_{N,FD,PD}$	N_{PD}	PCI_{PD}	N_{proVar}	∑TM
S0	.11	4	.44	4	42
S1	.21	3	.65	3	49
S2	.25	2	.74	2	31

Figure 4.23 visualizes the total setup costs for different levels of overdesign, component commonality, and product variety. As suggested by the correlation analysis, a positive association between component commonality and setup costs, as well as overdesign and setup costs are observed. Although previous studies note a negative association between overdesign and setup costs (e.g., Hackl et al., 2020), these findings are not completely in contrast. Increased overdesign reduces setup costs if the two components share at least certain processes that lead to a reduction in the total number of tasks. Nevertheless, the analysis highlights that cause-effect relationships are much more complex as suggested by recent studies.

Figure 4.23:
Effects of component overdesign (lower panel) and component commonality (upper panel) on total setup costs for different levels of product variety.



Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

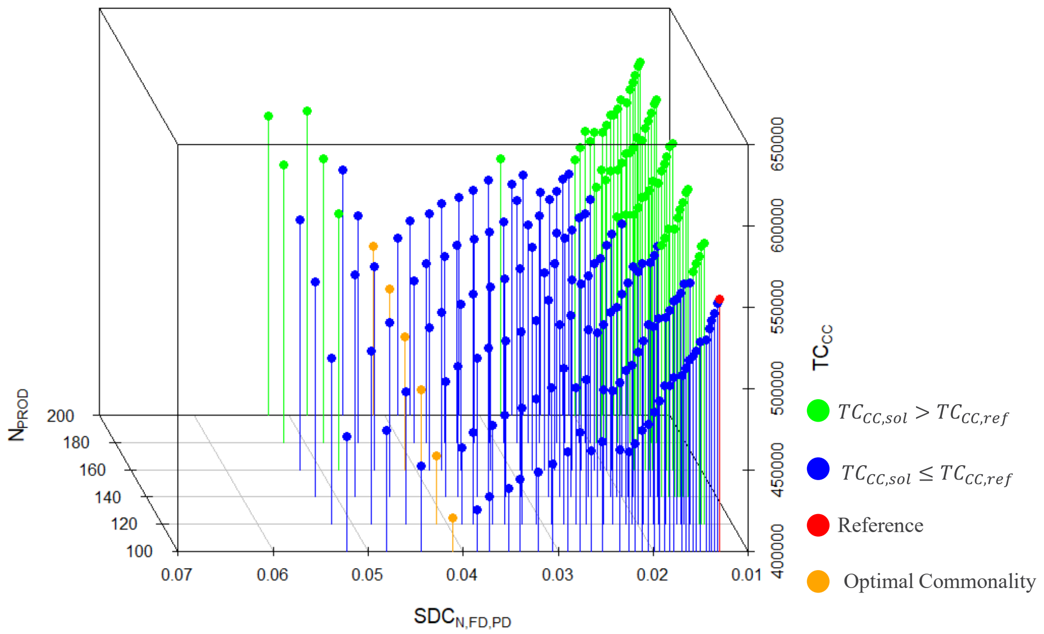
4.5.2 Effect on Total Complexity Costs

In the second part of this experiment, the effects of internal complexity on total costs in terms of product variety and overdesign are analyzed. In doing so, the total costs for each complexity cost type are summarized where TC_{CC} represents the total complexity costs defined as:

$$TC_{CC} = TC_{od} + TC_{devel} + TC_{pa} + TC_{supply} + TC_{stock} + TC_{order} + TC_{tooling} + TC_{setup} \tag{4.57}$$

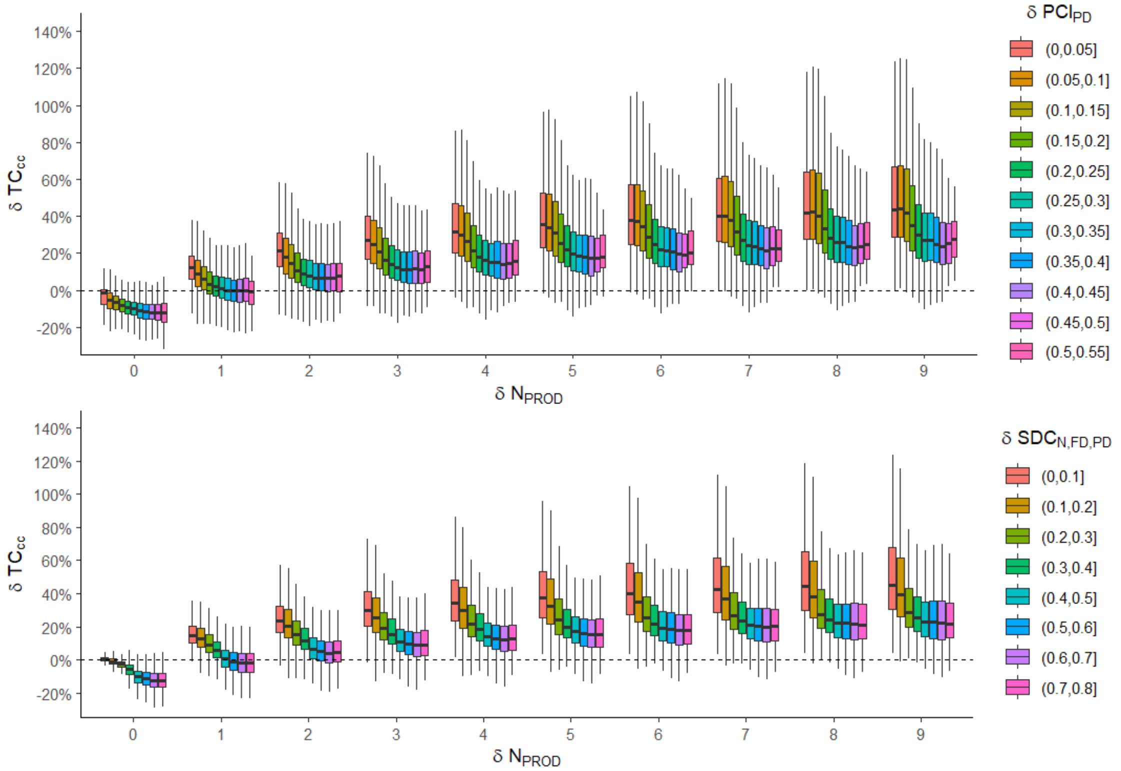
While some of the costs decrease with increasing overdesign (or increasing component commonality) such as (TC_{devel}, TC_{pa}, TC_{supply}, TC_{stock}, TC_{order}, $TC_{tooling}$), others increase (TC_{od}), and others show both increasing and decreasing effects (TC_{setup}). The result is a two-sided effect, as demonstrated by the case study (see section 3.2.5), where firms gain economic advantages as long as the cost-reducing effects compensate for the cost-increasing effects. To reduce costs, firms search for a cost-minimizing product family design for a given level of product variety. Figure 4.24 shows the graphical representation of this problem. The two independent model variables (product variety and overdesign) defined the design space in the xy-plane. This design space is projected into a solution space (z-axis), representing the total complexity-induced costs. Each point in Figure 4.24 represents an individual product family design, where the red point indicates an initial starting design without an overdesign and a medium level of product variety ($N_{PROD} = 100$). All solutions with lower total complexity costs and at least as many products as the reference costs are colored in blue. Starting from this reference design, an increase in component commonality leads to a reduction in total complexity costs in the first step. The orange points indicate the optimal degree of overdesign (component commonality) for each level of product variety, characterized by the minimum total complexity costs. Beyond this point, costs begin to rise when overdesign is further increased. At a certain point, the additional material costs become so high that the decreasing cost effects cannot be compensated, resulting in higher total complexity costs than the reference (green points).

Figure 4.24:
Exemplarily visualization of the two-sided effect of overdesign (component commonality) for a randomly chosen product family under different levels of product variety.



The example highlights several essential aspects. First, the reference (starting) design is important since increasing overdesign does not necessarily reduce costs. Starting from a loosely coupled design (red point), a moderate level of overdesign ($SDC_{N,FD,PD} \approx 0.04$) leads to fewer costs. If overdesign is further increased, costs begin to rise. However, if firms start with a more coupled design ($SDC_{N,FD,PD} > 0.05$) a further increase in overdesign results in even higher costs. In this case, the best approach is to reduce the level of overdesign. A second observation is that there exist local (e.g., $SDC_{N,FD,PD} \approx 0.02$) and global minima ($SDC_{N,FD,PD} \approx 0.04$). For example, if firms start with the reference design (red point) they may stop increasing overdesign at $SDC_{N,FD,PD} \approx 0.02$ as they think that a further increase will lead to higher costs. However, the figure shows that a global minimum is present at $SDC_{N,FD,PD} \approx 0.04$. These local minima are the result of interaction effects within the model. For example, the previous section showed that a moderate increase in overdesign leads to more setup activities, while a further increase in overdesign leads to fewer setup activities (see section 4.5.1.5). The entire data set of 5000 individual product family designs is used to generalize this finding. Figure 4.25 visualizes the differences in total complexity costs (δTC_{CC}) for different levels of component commonality (δPCI_{PD}, upper panel), overdesign ($\delta SDC_{N,FD,PD}$, lower panel), and product variety (δN_{PROD}).

Figure 4.25:
Differences in total complexity costs under different levels of product variety and component commonality (upper panel) as well as overdesign (lower panel).



Note. The bars indicate range between the 5%-quantile and the 95%-quantile.

The graph shows the typical U-shaped curve of increased overdesign and component commonality, where the pattern is more prominent for component commonality. Therefore, the remaining analysis focuses on component commonality. There are two additional arguments for focusing on δPCI_{PD} rather than overdesign. First, it allows for a better alignment of results with the existing literature, and second, both variables show a strong correlation with each other ($c = .69, p < .01$).

Table 4.15 reports the median complexity-induced costs for each design and level of product variety. It is observed that a moderate increase in component commonality ($30\% < \delta PCI_{PD} < 45\%$) leads to minimum complexity-induced costs under medium and high levels of product variety ($\delta N_{PROD} > 1$). For low levels of product variety, the effect is not observed. However, it is observed that complexity-induced costs decrease more under low levels of δPCI_{PD} and less under high levels. For example, an increase in component commonality by around $35\% < \delta PCI_{PD} < 40\%$ leads to an average cost decrease of $\delta TC_{CC} = 35.4\% - 17.9\% = 17.5\%$ under a medium level of product variety ($\delta N_{PROD} = 5$) compared to the initial starting design. This underlines the cost advantages firms can gain by increasing component commonality. Another way to read the results in Table 4.15 is that component commonality allows firms to offer an increased product variety by keeping complexity-induced constant or even reducing these costs. For example, a firm can increase the product variety from $\delta N_{PROD} = 2$ to $\delta N_{PROD} = 7$ while keeping the complexity-induced costs almost constant ($TC_{CC} = 21.2\%$ vs. $TC_{CC} =$

21.5%) if they increase the level of component commonality from $\delta PCI_{PD} = 0 - 5\%$ to $\delta PCI_{PD} = 40 - 45\%$.

Table 4.15:
Median total complexity ($\boldsymbol{TC_{CC}}$) costs for each level of product variety and component commonality.

	δ N_{PROD}									
δ PCI_{PD}	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0-5 %	-1.5%	12.3%	21.2%	27.2%	31.8%	35.4%	38.1%	40.4%	42.2%	43.6%
5-10 %	-5.4%	8.7%	17.9%	24.7%	29.8%	34.1%	37.5%	40.3%	42.5%	44.1%
10-15 %	-6.5%	5.8%	14.4%	20.9%	26.3%	31.0%	34.7%	37.7%	40.2%	42.3%
15-20 %	-8.1%	3.4%	10.6%	16.5%	21.3%	25.4%	28.8%	31.7%	33.7%	35.2%
20-25 %	-9.1%	1.9%	8.8%	13.9%	18.1%	21.7%	24.6%	27.0%	28.6%	30.1%
25-30 %	-9.9%	1.1%	7.5%	12.4%	16.2%	19.5%	22.2%	24.2%	25.9%	27.0%
30-35 %	-11.0%	0.1%	6.4%	11.3%	15.0%	18.6%	21.7%	24.0%	25.8%	27.1%
35-40 %	-11.6%	-0.3%	6.5%	11.3%	15.1%	17.9%	20.6%	22.4%	24.0%	25.1%
40-45 %	-11.9%	-0.3%	6.5%	11.5%	14.2%	17.6%	19.7%	21.5%	23.0%	23.9%
45-50 %	-11.9%	0.0%	6.6%	11.5%	14.7%	17.7%	19.8%	22.7%	23.6%	25.2%
50–55 %	-12.0%	0.0%	8.0%	12.6%	15.7%	18.1%	20.5%	22.8%	24.7%	27.5%

This pattern is further generalized by using a linear regression model. The model is formulated as:

$$\delta TC_{CC} = \beta_1 * \delta N_{PROD} + \beta_2 * \delta PCI_{PD} + \beta_3 * highPCI\ x\ \delta PCI_{PD} \quad (4.58)$$

Product variety is measured via the number of products (δN_{PROD}). As suggested by the previous analyses, an increasing product variety is associated with increasing costs. The second variable is component commonality, where increasing values of δPCI_{PD} are associated with decreasing costs as long as the increase is not too high. Therefore, the dummy variable $highPCI$ is introduced, which is defined as:

$$highPCI = \begin{cases} 0, & \delta PCI_{PD} < .3 \\ 1, & \delta PCI_{PD} \geq .3 \end{cases} \quad (4.59)$$

$highPCI$ allows for analyzing differences between the effect of component commonality for low values of component commonality ($highPCI = 0$) and high values ($highPCI = 1$).

Regression results (Table 4.16) indicate a significant positive standardized effect size for product variety ($\beta_1 = .54; p < .01$). A significant negative effect ($\beta_2 = -.33; p < .01$) is identified for component commonality. These results confirm that increased levels of component commonality can compensate for increased complexity costs caused by product variety. The compensation effect gets smaller the larger the product variety since a small positive effect size for the interaction term is identified ($\beta_3 = .05; p < .01$).

Table 4.16:
Standardized effect sizes to estimate the impact of product variety and component commonality on complexity-induced costs.

Predictor	*b*	*b* 95% CI [LL, UL]	sr^2	sr^2 95% CI [LL, UL]	Fit 95% CI [LL, UL]
		Model 1			
δ N_{PROD}	0.54**	[0.54, 0.54]	0.27	[.27, .27]	$R^2_{adj.}$ = .444** [.44,.45]
δ PCI_{PD}	-0.33**	[-0.33, -0.32]	0.05	[.05, .05]	
highPCI x δ PCI_{PD}	0.05**	[0.05, 0.05]	0.01	[.00, .01]	

Note. A significant *b*-weight indicates the semi-partial correlation is also significant. *b* represents unstandardized regression weights. sr^2 represents the semi-partial correlation squared. *LL* and *UL* indicate the lower and upper limits of a confidence interval, respectively. * indicates $p < .05$. ** indicates $p < .01$; VIF<1.1 for all predictors.

In sum, the analyses in this section generalize the empirical observed two-side effect of component commonality as noted by the literature (Fixson, 2005, 2006; Hackl et al., 2020; Labro, 2004; Thonemann & Brandeau, 2000). The cost-decreasing potential of a moderate increase in component commonality is demonstrated. However, too much component commonality results in higher costs since the additional material costs cannot be compensated by economies of scale, such as reduced units in stock, fewer purchasing orders, or reduced tooling equipment. The following section discusses the results.

4.5.3 Discussion

The individual complexity cost drivers are the foundation for the analyses in this section. Following the call for a more in-depth investigation of economic consequences caused by internal complexity (Hackl et al., 2020; Lyons et al., 2020; Trattner et al., 2019), this section integrates eight existing analytical models from the literature into the numerical EAD framework. In the first analyses, the effects of increased overdesign and component commonality are investigated for each cost-effect separately. Except for the setup costs, these analyses confirm the cause-effect relationships noted by the literature (e.g., Thonemann and Brandeau, 2000) and summarized by Hackl et al.'s (2020) impact model. Nevertheless, due to the power of numerical experiments, it is observed that causes are more complicated, as suggested by the impact model for some cost effects. For example, the study finds a direct and indirect effect of component commonality on inventory levels. As noted, setup costs show differences in the assumed relations. In contrast to the literature, a positive association between component commonality and the number of setups is observed. An in-depth analysis demonstrates that an increase in overdesign and, therefore, component commonality reduces the number of setups (and, therefore, setup costs) only under a certain condition. If two replaced components share a process, the total number of tasks decreases due to volume-pooling effects. If they do not share processes the total number increases since the overdesigned component requires, prior independent, processes but at a higher demand rate. It is important to note that this finding results from the underlying assumptions that the processes of the overdesign component are the union of the component processes it replaces. A second important aspect is the potential effects of the production order. This work uses a random order model and assumes that firms have no information on future demand. In practice, however, firms can look ahead (at least to some degree) and cluster batches of identical components to reduce the number of setups (e.g., Kekre 1987).

All analyses reveal a degressive cost curve, reflecting the economies of scope under increasing product variety (e.g., Panzar & Willig, 1977). The higher the product variety, the less the increase in total costs, as new products have a higher chance of reusing already existing resources. This effect is stronger the higher the level of component commonality. In contrast, studies focusing on complexity-induced costs note a progressive cost increase (e.g., Ripperda & Krause, 2017; Schuh, 2005). As argued in the prior considerations, such a cost curve occurs only if cost rates depend on product variety or higher-order effects occur, driving resource consumption. The work of Labro (2004) reports studies highlighting the effects of internal complexity on cost rates as well as higher-order effects. For example, an increased product variety or decreased component commonality increases process variety, requiring better-trained workers with higher hour rates. Setup times are an example of higher-order effects. Product variety and commonality affect the number of repetitions, causing learning and forgetting effects (Adler & Clark, 1991; Globerson & Levin, 1987; Yelle, 1979). The average setup time for more common machine setups is lower than for exotic setups, as they rarely occur. However, these effects are not only limited to the production. Increasing product and component variety drives the development time since engineers must check their dependencies with other components (S. Gupta & Krishnan, 1999; Perera et al., 1999). Nevertheless, the literature argues that the development costs of an overdesign component are less than those of the individual (non-overdesigned) ones (Eynan & Rosenblatt, 1996; Perera et al., 1999). For reasons of simplification, this work does not include all noted effects. For example, cost rates are assumed to be constant while the decreased development costs of overdesigned components are considered. Including these additional effects will impact the cost curve and may result in a higher cost increase under high levels of product variety. However, the question arises whether these effects result in a disproportionately cost increase.

The second part analyzes the effects of increased overdesign and product variety on total complexity-induced costs. The results generalize the two-sided effect of component commonality (Labro, 2004; Thonemann & Brandeau, 2000; Trattner et al., 2019) as a U-shaped cost curve is observed. The apex of this curve indicates the optimal degree of component commonality. At this point, complexity-induced costs reach their absolute minimum, where a further decrease or increase results in higher costs again. Reaching the optimal point of component commonality is important for firms as they face the challenge of choosing the most economical product family design out of many alternatives (Kulak et al., 2010). While, in theory, locating the optimal degree of component commonality seems simple, it leads to challenges in practice for several reasons. First, it requires the cost estimation of product family design alternatives at an early stage where uncertainty is high (Fixson, 2006; Ripperda & Krause, 2017; Skirde et al., 2016). Due to limited information, estimated costs are biased, and firms may erroneously choose the wrong alternative. A second challenge is the large design space. In practice, limited resources allow the creation of only a few concepts. Therefore, only a small proportion of the theoretical design space is explored, while the theoretical number of potential design alternatives (NDA) is much higher. Even in the simplest case with 50 components, where firms replace two components with an overdesigned one, there are over 1200 alternatives, as the following equation shows.

$$NDA = \binom{50}{2} = 1225 \qquad (4.60)$$

However, design decisions are much more complex and go beyond these simple cases. Therefore, Ripperda and Krause (2017) note the importance of case-by-case decisions made by engineers. Finally, a third problem is the existence of local minima since not all cost effects are strictly increasing or decreasing. As shown for the setup costs, an increase in component commonality is associated with increasing setup costs under some conditions. In the belief that the optimal degree of commonality is reached, firms stop searching for alternative designs and may oversee even better solutions.

4.6 Conclusion

This section investigates the economic consequences of internal complexity. In doing so, a large variety of 5000 product families is created by conducting numerical experiments. For each product family, product variety and the degree of component overdesign (and, therefore, component commonality) are stepwise increased, leading to over 1 million unique designs. Resulting complexity-induced costs are estimated for each design using eight analytical models adapted from the literature and integrated into the numerical EAD framework. Analyses are separated into two main parts. The first analyses investigate the mechanisms behind each cost effect individually. The second part investigates the effects of component commonality and product variety on total costs.

This section confirms the cost-reducing effects of overdesign and component commonality (both show a strong correlation) on several costs, as summarized in Table 4.17. In line with the literature, increasing overdesign increases material costs and decreases development, part administration, supplier management, and tooling costs. A decrease is also observed for inventory holding and purchasing order costs, which are variable. However, the causal effects for the last two cost types are more complex, as suggested by the impact model. For example, increasing overdesign reduces the number of distinct components, leading to a cost reduction on the one hand and, on the other hand, increases the average lot sizes, leading to a cost increase. In contrast to existing studies, overdesign and component commonality show a mixed effect on setup costs. If two replaced components share processes, the setup costs decrease due to volume-pooling effects. If they do not share a process, setup costs increase since the overdesigned component requires the process of the component it replaces but at a higher quantity.

Table 4.17:
The effects of increased overdesign and component commonality on complexity-induced costs for each cost type.

Cost Type	Complexity-induced Costs
additional material costs	increasing
development costs	decreasing
part administration costs	
supplier management costs	
inventory holding costs	
purchasing order costs	
tooling costs	
setup costs	decreasing and increasing

The effects of total complexity-induced costs are investigated in the second part of this section. In doing so, the total complexity-induced costs are defined as the sum of the complexity-induced costs for each cost type. Since some costs decrease while others increase under increasing overdesign and component commonality, the two-sided face of component commonality is highlighted. Specifically, component commonality shows a U-shaped curve where the apex indicates the point of optimal commonality. Due to the complex interactions within the model, representing already simplified assumptions, local minima exist, making identifying optimal component commonality challenging in practice. The analyses in the second part further show that the right level of overdesign and component commonality enables firms to offer a larger product variety and, therefore, increase sales opportunities by keeping total complexity-induced costs constant.

This work has several implications for research and practice, summarized in Figure 4.26. From a researcher's perspective, this work follows recent calls for operationalizing the cause-effect relationships leading to complexity-induced costs (Hackl et al., 2020; Lyons et al., 2020; Trattner et al., 2019). It sheds light on the vague concept of complexity-induced costs (Meßerschmidt et al., 2020) and reveals that causes are more complicated, as the literature suggests (e.g., Hackl et al., 2020; Perera et al., 1999). However, the analyses just scratched the surface. Therefore, the first implication calls for further in-depth analyses to detail the cause-effect relationships. This includes model extensions where certain assumptions are relaxed and partial models are extended. A second implication refers to the generation of design alternatives. This work increases overdesign by replacing a low-demand component and its nearest neighbor with an overdesigned component. However, this is only one heuristic to increase overdesign. More advanced heuristics may result in even higher cost savings under identical conditions. Evaluating the heuristic's performance using the numerical EAD framework allows for identifying specific rules of thumb to guide practitioners in the future.

Figure 4.26:
Implications for research and practice.

Research

- This work indicates a first step towards operationalizing the complicated and interacting cause-effect relationships that lead to complexity-induced costs. More in-depth investigations are needed.
- The design space for generating alternative product family designs is large. Advanced heuristics will ensure efficient design space exploration and allow deriving rules of thumb to guide practice.

Practice

- An increase in component commonality is not always the best case. A reduction in complexity-induced is only achieved if the optimum point has not yet been exceeded.
- Increase the transparency of complexity-induced costs to make cost-optimal decisions.
- Simulation models allow for a systematic, quick, and cheap design space exploration under uncertainty.

Three practical implications are derived based on the results. First, an increase in component commonality is not always the best choice. Complexity-induced costs are only reduced if the current product design has not exceeded the optimal degree of commonality. While collecting empirical observations (section 3.5.3), multiple examples where firms got beyond the optimal point were observed. For example, one company faced the question of whether it should standardize the frame for its products or create two variants. Believing that the cost-savings can compensate for the increased material costs (around 7%), they pick the standardized option. However, a detailed ex-post analysis revealed that the two-frame alternative would have been around 3% cheaper after three years. Three percent sounds little; however, they represent a low six-digit savings. This example shows that too much overdesign is a practice problem.

The second practical implication notes that transparency in complexity-induced costs is necessary for cost-optimal decision-making. Remember, costs are effect indicators rather than causes. Without knowledge of these causes, engineers and management face a tough challenge to estimate the complexity-induced effects of a design alternative, leading to an under- or over-estimation of cost savings. While overestimating cost-savings might be the rarer case, underestimation is more common. The reason that the major cost-increasing effect of component commonality, the additional material costs, are direct costs, which are much more prominent than the savings in indirect costs. This work is a further step towards increased transparency as it sheds light on the cause-effect relationships between product family design, product variety, and costs. The numerical EAD further allows for estimating the total costs of product design alternatives.

A last practical implication is related to the large design space. While design alternatives are almost endless, firms have limited resources to explore the design space. Simulation models can support this process as they allow for quick and cheap design space exploration. Several studies demonstrate the power of numerical models in this context (e.g., Fujita, 2002; Kashkoush & ElMaraghy, 2017; Sinha & Suh, 2018). They further allow locating their current product family design in the design space (e.g., below or beyond the optimal degree of component commonality). Finally, simulations support identifying design regions of interest and allow for modeling uncertainty in terms of 'what if'. Several authors note that high uncertainty characterizes decision-making during the product development process (Chen et al., 2022; Fixson, 2006; Geng, Chu, & Zhang, 2010; Skirde et al., 2016). For example, material costs of an overdesigned component are not available a priori. Only rough estimations exist, such as those based on the engineer's experience. Simulation models such as the EAD allow determining under which conditions and at which probability a design alternative may be the most economical. For example, they can estimate the maximum material costs of an overdesign component to achieve cost savings. Besides these implications, it is essential to note that the numerical EAD framework supports the decision-making process among many other methods and tools. In line with Ripperda and Krause (2017), this work agrees that product development decisions are highly individual, driven by many (observable and latent) factors, and, therefore, must be made on a case-by-case basis.

This work has several limitations, which are briefly discussed. First, several simplifications and assumptions are made. For example, a random production order is assumed where literature notes many other models, such as the look-ahead rule (e.g., Kekre, 1987). Additionally, cost rates are assumed to be constant in the model, whereas Labro (2004) notes that cost rates can vary under different levels of component commonality. For example, learning and forgetting effects are not modeled yet. A second limitation is related to the numerical EAD model itself. It is observed that the chance of new products requiring new components is relatively low. Therefore, the impact of product variety is limited to variable complexity-induced costs. Speaking of costs, the model assumes the absence of cost stickiness. Economic benefits, as suggested by the model, may not fully appear in practice. For example, increased component commonality can reduce the units in stock. However, it does not necessarily reduce inventory costs since firms cannot reduce warehouse building costs or the number of intra-logistic employees in the short- and medium-run. Remember, such resources are only variable in the long run (section 2.4.1).

Besides these limitations, there are several open issues for future research. For example, this work focuses on the direct economic consequences of overdesign and commonality in terms of costs. However, there are several other consequences of component commonality, such as flexibility or agility (Cormier et al., 2009; Fuchs & Golenhofen, 2019; Greve et al., 2020; Orton & Weick, 1990) and its ability to reduce risk (Baker et al., 1986; Labro, 2004). Subramanian, Ferguson, and Beril Toktay (2013) further note the positive effects of component commonality on the supplier side. A second potential field for future research is the operationalization of product variety. This work operationalizes product variety by the number of distinct products. However, in her theoretical work, Buchholz (2012) notes that product variety is also related to product differences. For example, product variety is lower if products are more similar and vice versa. Adding new measures for product variety would allow future research to identify the costs of adding more heterogeneous (exotic) product configurations to the product mix. Such an extension adds to the work of Trattner et al. (2019), who discuss potential differences in firm's operational performance by adding related (similar) or unrelated (dissimilar) configurations to the product mix.

5 Effects on Product Costing System Accuracy

5.1 Introduction

The impact of variety-induced complexity is not limited to increased total costs, as shown in the previous section. It further affects the accuracy of product costing systems (PCSs). Section 2.6.3 highlighted the importance of product costing system accuracy since costing information is used for a variety of decisions within firms, such as resource acquisition or supply and product mix decisions (V. Anand et al., 2017; Balakrishnan & Sivaramakrishnan, 2002; M. Gupta & King, 1997; Homburg et al., 2018; Innes et al., 2000; Labro, 2018). Prior studies investigated the drivers of product costing systems accuracy, mainly using numerical experiments (V. Anand et al., 2019; Balakrishnan et al., 2011; Cardinaels & Labro, 2008; Labro & Vanhoucke, 2007, 2008; Mertens, 2020; Schmidt et al., 2023). Numerical experiments allow the observation of the true resource consumption necessary for estimating the true (benchmark) product costs (PC_B). Although these studies gained valuable insights, they have some limitations. First, resource consumption patterns (P_{RD}) are created via a correlation-based approach rather than an underlying product family design. Hence, patterns used in prior studies may not fully reflect the resource consumption of configurable products present in practice. Related to this issue, cost accounting faces criticism for being too far away from practice (Brierley et al., 2001) or other fields (Labro, 2015a), which is a second motivation for this study. For example, numerical studies mentioned above operationalize the resource consumption patterns via the degree of resource sharing (DNS_{RD}) and average correlation among resources (COR). These variables are difficult to observe in practice as they require knowledge of product resource consumption. Another example is the strong focus of cost accounting studies on activity-based costing (Mertens, 2020). Empirical studies, however, indicate that only a small proportion of firms (15% - 40%) use ABC systems, while the others use traditional (volume-based) costing systems (see Table 5.1). Literature notes several reasons for the low adoption rate, such as the costs of implementing and managing ABC systems (Kaplan & Anderson, 2007; Labro & Vanhoucke, 2007; Schoute, 2011). However, Ittner et al. (2002) note that firms using ABC can improve their performance by reducing cycle times and manufacturing costs as well as increasing quality. Even though most of the firms rely on traditional PCS, there is only a small body of literature comparing traditional (volume-based) and sophisticated (ABC) product costing systems (exceptions are: Mertens, 2020; Schmidt et al., 2023). Focused on ABC, these studies further assume the typical ABC cost hierarchy of unit-, batch-, product- and facility-level costs, where volume-based systems take the traditional perspective as they differ solely between variable and fixed costs. Again, empirical studies show an unclear picture regarding the ABC cost hierarchy in practice, as S. W. Anderson and Sedatole (2013) summarize. A third motivation comes from the practical perspective on errors in product costing systems. In firms, product managers make product mix decisions using PCS information (Banker & Hughes, 1994). They estimate whether product target prices exceed their total costs plus an additional contribution margin. In doing so, they require knowledge of the product-level error rather than an overall system-wide error. Although there are studies investigating product-level errors (Datar & Gupta, 1994; M. Gupta, 1993; Labro & Vanhoucke, 2007; Mertens, 2020; Schmidt et al., 2023), they note that the effects are still fuzzy and non-straightforward.

Table 5.1:
Adoption rate of ABC across different empirical studies

Study	Sample	ABC Adoption Rate
Krumwiede (1998)	225 US manufacturing firms	17%
Ittner et al. (2002)	2789 US manufacturing firms	26%
Cagwin and Bouwman (2002)	204 US firms	23%
Hughes and Gjerde (2003)	130 US manufacturing firms	38%[1)]
Bjørnenak (1997)	75 Norwegian firms	40%
Cinquini, Collini, Marelli, and Tenucci (2015)	132 Italian firms	18%
Al-Sayed and Dugdale (2016)	152 UK manufacturing firms	37%
Al-Omiri and Drury (2007)	147 UK manufacturing firms	29%
Schoute (2011)	191 Dutch manufacturing firms	15%
Neumann and Cauvin (2007)	Survey among 2500 management accountants (duplications possible)	23%
Gosselin (1997)	161 Canadian manufacturing firms	30%

Note. [1)] summarizes firms using ABC or ABC with volume-based allocation heuristics

This section tackles these challenges. In doing so, the numerical EAD framework is used to create 5000 unique product family designs. Differences in the resource consumption patterns are compared to those generated by V. Anand et al.'s (2019) framework (ABL). Generating resource consumption patterns by using the EAD has several advantages compared to the ABL. First, it allows for generating resource consumption patterns in which products share certain similarities as they are derived from the same underlying product family design. Therefore, EAD-generated resource consumption patterns represent product variants rather than individual and unrelated products. Buchholz (2012) notes that products are variants of each other if they share more similarities than dissimilarities. The second advantage is that EAD comes with a set of new measures. As noted above, the ABL operationalizes resource consumption patterns via two variables, which are difficult to observe in practice. The EAD, however, comes with several additional measures which are better to observe. Observability, however, is related to efforts for information gathering and the time when information becomes available. For example, the product architecture is defined in the early development stages, whereas decisions on the production technology are made in later stages. Therefore, the EAD allows for analyzing the effects of product family design on product costing system accuracy, increasing the practical relevance of cost accounting research. By combining the generated product family design with 33 product costing systems, differences between the reported (PC_H) and true product costs (PC_B) are measured on system- and product-level errors. Error metrics are defined in line with prior studies (e.g., Balakrishnan et al., 2011; Labro & Vanhoucke, 2007, 2008; Mertens, 2020;

Schmidt et al., 2023) and work as dependent variables. In contrast to existing studies, this work does not consider the ABC cost hierarchy (since most firms rely on volume-based costing) and differs solely between non-unit level (fixed) and unit-level (variable) costs. Analyzes are separated into system- and product-level analyses. While the system-level analysis investigates the total costing error, the product-level analysis investigates the cross-subsidization of products in terms of under- and over-costing. This experiment is related to the third research objective (RO III).

Results indicate that resource consumption patterns are more homogenous under the EAD, except for two anomalies. This study identifies more variations in total resource consumption across resources (TRC_{cv}) and products ($PROD_{RNG}$), increasing challenges for volume-based PCS as product differences are further manifested. Results suggest that ABC systems show less than 47% of the error of volume-based product costing systems. Volume-based systems are more robust to product design changes, and errors are better to predict. However, they are highly sensitive to changes in the static demand distribution and the introduction of new products. This is important since firms have limited control over static demand distribution. Product-level results confirm the typical pattern of high-volume products being over-costed and low-volume products under-costed if firms use a volume-based allocation. Additionally, it is observed that a product's exoticness is a good proxy for under-costing. This pattern is not observed under ABC systems where the under-/over-costing pattern is non-straightforward. Nevertheless, it is observed that products with high intra-product heterogeneity tend to be prone to errors.

The remainder of this section is structured as follows. Prior considerations on error metrics and the current research in the context of product costing system accuracy are made in the following section (5.2). Section 5.3 contains the numerical experiment. The first analysis investigates differences in resource consumption patterns between the EAD framework and the ABL model. The following two sections discuss system-level (5.3.3) and product-level errors (5.3.4). Results are discussed in section 5.3.5. Finally, this chapter ends with implications in section 5.4.

5.2 Prior Considerations

5.2.1 Errors in Product Costing Systems

Following recent cost accounting studies (e.g., V. Anand et al., 2019; Balakrishnan et al., 2011; Mertens, 2020), the resource consumption matrix (P_{RD}) represents a full information setting. Full information settings allow the observation of true product costs (PC_B), whereas, under incomplete information, a product costing system (PCS) measures the reported (heuristic) product costs (PC_H). The difference between true and reported product costs is known as costing error. Section 2.6.4 provides an overview of system- and product-level error metrics. According to the model of V. Anand et al. (2019), true indirect product costs ($PC_{B,in}$) are calculated as[81]:

[81] The model assumes that all costs are variable on unit-level.

$$PC_{B,in} = P_{RD} * RCU_{in} \tag{5.1}$$

With RCU_{in} being the indirect unit resource costs. The index indirect is added since direct costs are assumed to be error free. Indirect resource unit costs RCU_{in} are calculated as:

$$RCU_{in} = \frac{RC_{var,in}}{TRC} \tag{5.2}$$

where $RC_{var,in}$ are the variable resource costs and TRC the total resource consumption defined as:

$$TRC = P_{RD}{}^{T} * DMD \tag{5.3}$$

True product costs (PC_B) are than given as the sum of indirect true costs and the direct product costs (PC_{direct})[82]:

$$PC_B = PC_{B,in} + PC_{direct} \tag{5.4}$$

Error-free product costing systems ($PC_{B,in} = PC_{H,in}$) exist only in theory. In practice, firms know the unit resource costs (RCU) and the total resource consumption (TRC). However, they do not have full information about resource consumption (entries of P_{RD}) for each cost object (V. Anand et al., 2019). Reasons are information collection efforts or the inability to observe the consumption of some resources at all. The latter occurs, for example, when cost assessment and the incurrence of costs are delayed, such as when a product variant is introduced long after product families' initial entry into the market. Due to these facts, firms have to deal with indirect costs and, therefore, face a trade-off between *"accuracy and the cost of accuracy"* (Labro & Vanhoucke, 2007, p. 940) or the assumed benefits of more accurate product costs versus the costs of information collection (Balakrishnan et al., 2011). Technically, limited information in resource consumption patterns means that not all values of P_{RD} are known, resulting in biased product costs (PC_H) reported by the PCS. To allocate the indirect costs onto cost objects, PCSs use heuristics. These heuristics contain rules on how to allocate indirect costs to products. Under ABC, a two-stage allocation procedure, according to Figure 2.16, is assumed. In the first stage, resources (RCP) are grouped into activity cost pools (ACP). In the second stage, the costs for each activity cost pool are then allocated to cost objects using activity drivers (AD). Table 5.2 provides an overview of first-stage allocation heuristics, as reported by Balakrishnan et al. (2011). Heuristics are sorted in ascending order of informational need. While the size random method requires only information on the desired number of activity cost pools (ACP), the correlation-size heuristic additionally requires information on resources' similarity and resource costs.

[82] A distinction between true (index B) and reported (index H) costs is not necessary for direct costs since they are error-free as PCS traces such costs directly onto cost objects.

Table 5.2:
First-stage allocation heuristics according to Balakrishnan et al. (2011) base line experiment.

#	Name	Description	Informational Need		
			ACP[1)]	RC[2)]	CONS[3)]
1	random	Resources are randomly assigned to activity cost pools.	x		
2	size-random	Focus on the resources with the largest monetary value. First, the ACP largest resources are assigned to individual cost pools. The remaining resources are randomly distributed across ACP.	x	x	
3	size-misc	Focus on the resources with the largest monetary value. First, the ACP-1 largest resources are assigned to individual cost pools. The remaining resources are assigned to a miscellaneous cost pool.	x	x	
4	correlation-random	First ACP resources are randomly assigned to cost pools. The remaining resources are assigned to the ACP with the highest correlation.	x	x	x
5	correlation-size	Such as heuristic four, except the initial assignment is done by resource's size rather than random choice.	x	x	x

Note. 1) ACP= information on activity consumption by products; 2) RC= information on resource costs; 3) information on resource consumption pattern to estimate the similarity between resources.

According to the literature, two main heuristics are used in the second stage. The 'big-pool' heuristic uses the largest resource as a driver for each activity cost pool ($ACT\_CONS\_PAT_j, j = 1 \ldots N_{ACP}$). Indexed heuristics are more information-demanding and use a composite driver, averaging resource consumption over the i-largest resources assigned to ACP_j. The number of averaged resources is a user-specific parameter, and therefore, the method is always reported with an additional index, such as $AD_{indexed,i}$ where i indicates the number of averaged resources. The result of this second allocation step is the activity consumption pattern (ACT_CONS_PAT), representing the limited information environment. Columns represent the individual activity cost pools and rows the individual products. Entries define the proportion of activity consumption for each product and activity cost pool. Each column of ACT_CONS_PAT adds up to one since an activity needs to be fully used (V. Anand et al., 2019). The indirect, reported product costs are calculated as follows:

$$PC_{H,in} = ACT\_CONS\_PAT * ACP \tag{5.5}$$

The vector ACP holds information on the activity cost pool size, defined as the total resource costs for resources assigned to this pool. Remember, firms have information on the resource unit costs and know the heuristics of their PCS (V. Anand et al., 2019). Therefore, they know which resources are grouped into which activity cost pool and can calculate the activity cost pool size (ACP). For a more detailed discussion on allocation heuristics, see Balakrishnan et al. (2011).

Recent literature uses these heuristics under various firm environments (resource consumption patterns and cost structure) to analyze the accuracy of PCS. However, these heuristics assume an ABC system where empirical studies indicate that only 15% to 50% of firms use ABC systems while the other half uses simple costing systems that rely on traditional costing view (e.g., Drury & Tayles, 2005; Hughes & Gjerde, 2003). The reason is that complex PCS are expensive to implement and manage (Kaplan & Anderson, 2007). Under such simple costing systems, indirect costs are allocated using volume-based drivers such as the total production units ('DIV'), as Mertens (2020) summarizes. In doing so, all indirect costs are grouped into one cost pool in the first step. The cost driver is calculated by dividing those costs by the total number of products. The first step is similar to an ABC system with only one activity cost pool. Differences occur in the second stage. Traditional costing systems assume a constant driver in the second stage, whereas ABC systems have individual activity drivers for each product. While an averaged driver across all products is a source of error compared to ABC, it is less information-demanding and, thus, reducing efforts.

5.2.2 Relevant Literature

Over recent years, several studies have investigated the drivers of product costing systems' accuracy. This section provides an overview of the relevant findings gained by these studies (Table 5.3). Findings are separated into system and product-level results. Since later numerical experiments do not analyze the effect of measurement errors in PCS, these studies are not included in the table (e.g., Cardinaels & Labro, 2008; Datar & Gupta, 1994; Labro & Vanhoucke, 2007, 2008; Mertens & Meyer, 2018). The literature notes four main drivers of errors in PCSs: cost system sophistication, demand diversity, cost diversity, and diversity in resource consumption. Related to the level of cost system sophistication, Balakrishnan et al. (2011) find that information-intensive ABC-heuristics (e.g., correlation-based) perform better compared to simpler, less-information demanding ones (e.g., random). This also applies to second-stage heuristics, where indexed heuristics show fewer errors than the big-pool method. Another lever to increase cost system sophistication is by increasing the number of activity cost pools, where more ACP increase the resolution of the PCS. Balakrishnan et al. (2011) further find that the increase in accuracy is non-linear, meaning that some activity cost pools are already sufficient to reduce the total error by a large amount. Mertens (2020) compares complex ABC systems with more simple volume-based allocation (traditional) heuristics. He finds that traditional costing systems perform better under low information environments while complex PCS outperform these systems as ACP increases. He further observes that an increasing demand skewness (some high-volume products and many low-volume products) is associated with more errors in product costing systems under volume-based heuristics. However, Al-Omiri and Drury (2007) find that product mix and demand heterogeneity are not associated with the degree of cost system sophistication in practice[83]. Their findings do not necessarily mean that both variables are not associated with errors. However, they highlight that the perceived impact of product mix and demand heterogeneity is small since firms do not invest in more sophisticated costing systems. The third driver of errors in PCS is the diversity in the resource consumption patterns. Literature notes that more homogeneity (e.g., increased resource sharing or similarity among

[83] Other studies find that product diversity and the level of ABC adoption are positively associated (Malmi, 1999) or show a negative association (Bjørnenak, 1997)

resources) reduces product costing errors (Balakrishnan et al., 2011; Homburg et al., 2018; Mertens, 2020). Finally, the literature notes cost diversity as a fourth driver of errors in product costing systems. Mertens (2020) notes that an increase in indirect fixed costs leads to larger errors under volume-based PCS compared to an increase in indirect variable costs. Balakrishnan et al. (2011) further find that increasing disparity in resource cost pools (resource cost heterogeneity) is associated with higher errors.

Table 5.3:
Drivers, influencing the accuracy of product costing systems.

Authors	Finding	Operationalization
	Cost System Sophistication	
Balakrishnan et al. (2011)	information-intensive first-stage ABC allocation heuristics (e.g., correlation-based) perform better compared to less information-demanding ones (e.g., random)	Under RD = 'correlation-random' the EUCD is smaller compared to RD = 'random'
	information-intensive second-stage ABC allocation heuristics (e.g., indexed) perform better compared to less information-demanding ones (e.g., big-pool)	EUCD is smaller when AD = 'indexed' compared to AD = 'big-pool'
	some activity cost pools are sufficient to reduce the total error by a large amount	ACP is negatively associated with EUCD where the increase is non-linear for low values of ACP
Feltham (1977)	under a small number of cost pools, the chance of dissimilar (heterogeneous) resources being grouped into one cost pool is high, leading to larger errors.	ACP is negatively associated with EUCD where the increase is non-linear for low values of ACP
Mertens (2020)	volume-based allocation methods outperform more complex ABC heuristics under low information environments. However, ABC heuristics perform better under high information environments (large number of ACPs)	If ACP<4, volume-based heuristics show a smaller average EUCD compared to complex ABC-heuristics
	Demand Diversity	
Mertens (2020)	highly skewed demand distribution increases the total costing error under volume-based PCS	$DMD_{T10\%}$ is positively associated with EUCD under volume-based systems
Banker & Hughes (1994)	knowledge of product demand is not necessary to design PCS under ABC	$DMD_{T10\%}$ is not associated with EUCD under ABC
	Cost Diversity	
Mertens (2020)	increasing the proportion of variable costs on total costs (UNIT_SHARE) decreases the total error under specific volume-based heuristics.	negative association between UNIT_SHARE and EUCD under DIV allocation methods
Balakrishnan et al. (2011); Mertens (2020)	increasing resource cost disparity increases errors; however, the effect is negligible	$RC_{T10\%}$ is positively associates with EUCD
	Diversity in Resource Consumption	
Balakrishnan et al. (2011); Homburg et al. (2018); Mertens (2020)	increasing levels of resource sharing are associated with fewer errors	DNS is negatively associated with EUCD
Balakrishnan et al. (2011); Mertens (2020)	increasing correlation among resource are associated with fewer errors	COR is negatively associated with EUCD

Literature on product-level errors focuses primarily on product cross-subsidization (see Table 5.4). Specifically, accounting textbooks (e.g., Hilton, 2008; Horngren et al., 2012), empirical studies (e.g., Cooper & Kaplan, 1988b; Gietzmann, 1991; M. Gupta, 1993), and numerical experiments (e.g., Labro & Vanhoucke, 2007; Schmidt et al., 2023) show that high-volume products tend to be over-costed (OC) while low-volume products tend to be under-costed (UC). Conducting numerical experiments, Labro and Vanhoucke (2007) show that more products are UC than OC. If a product is over-costed, the costing error is larger compared to under-costed products. Mertens (2020), however, shows that high-volume products can be both OC and UC, indicating that the cross-subsidization pattern is more complex than suggested by prior literature. He finds that complex products (high $INTER$ and $INTRA$) tend to be OC if they have a large proportion of fixed costs on total costs. Usually, low-volume products show a high proportion of fixed costs on their total costs since fixed costs are allocated across smaller volumes compared to high-volume products. Schmidt et al. (2023) find that the low-volume UC and high-volume OC pattern occurs only if non-unit level (fixed) costs are introduced and a volume-based heuristic is used. Driven by empirical observations, product management literature claims that exotic (low-volume) products tend to be UC, while standardized (high-volume) products tend to be OC (Feldhusen, Nurcahya, & Löwer, 2007; Rebentisch et al., 2016; Schuh, 2005; Schuh et al., 2014).

Early product-level studies claim a simple cross-subsidization pattern, while recent studies indicate that this behavior is much more complex and occurs only under certain conditions. It is further observed that empirical studies are rare and quite old. Reasons might be the challenges in measuring true benchmark costs in practice, further explaining the high proportion of numerical and analytical studies in this context, as they allow for observing the true costs under full information settings. Although these studies gained valuable insights, they used the correlation-based approach of the ABL framework for generating resource consumption patterns, raising several concerns. First, do these patterns reflect the resource consumption of configurable product families derived from an underlying design? If patterns differ between models, do previous results still hold? A third concern is related to the study's practical implications. Heterogeneity in resource consumption is operationalized via the degree of resource sharing (DNS_{RD}) and the resource correlation (COR). However, these measures are difficult to observe in practice, increasing the challenge to transfer findings into practice. The following numerical experiment tackles these issues. The first part analyzes how patterns generated by the EAD and ABL framework differ. By using the EAD-generated patterns, the second analysis checks whether existing findings still hold. In doing so, the EAD product design measures from section 3 are used to allow for better adoption of findings into practice.

Table 5.4:
Drivers of product-level errors as state in literature

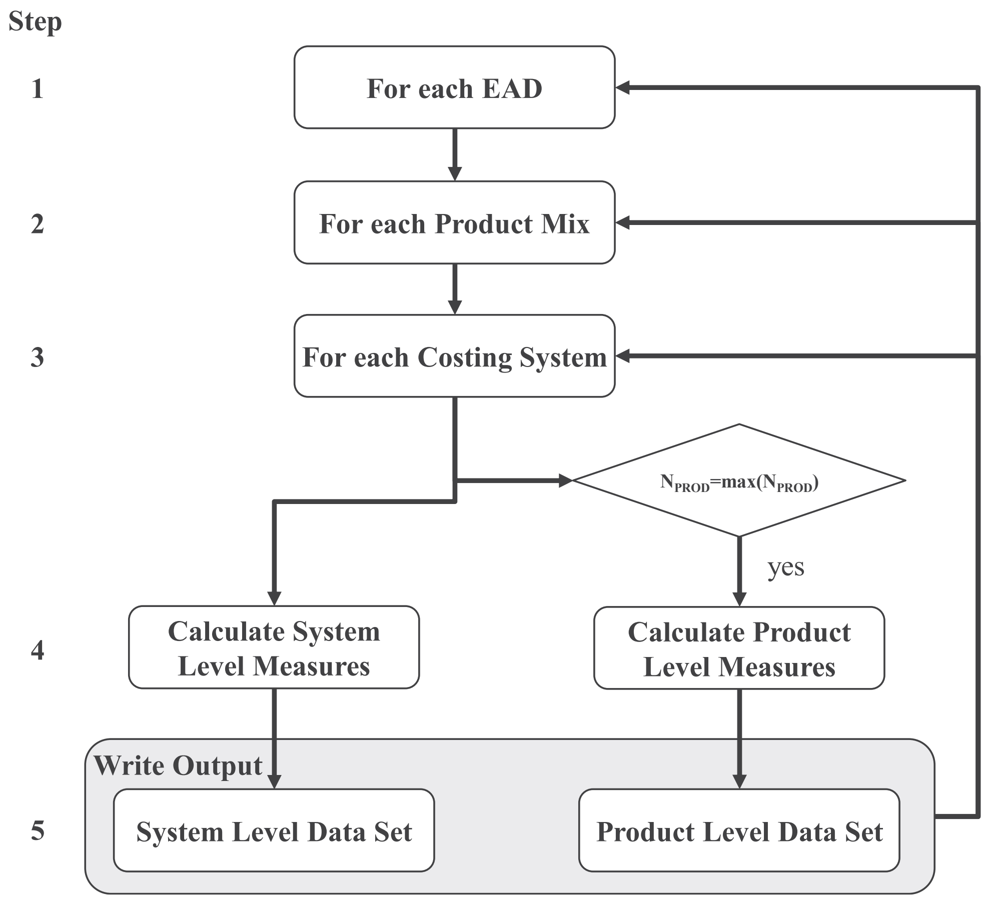
Authors	Finding	Operationalization
Labro and Vanhoucke (2007)	Under ABC systems, more products are under-costed (UC), while only some are over-costed (OC).	the proportion of under-costed products (P_{UC}) is larger than the proportion of products being over-costed (P_{OC}) ($P_{UC} > P_{OC}$)
	Since more products are UC than OC, the amount of OC is larger than the amount of UC.	the absolute percentage error is higher for OC-products compared to UC-products on average
Cooper & Kaplan (1988)	High-volume products tend to be OC, while low-volume products tend to be UC.	case study
Gietzmann (1991)	High-volume products tend to be OC, while low-volume products tend to be UC.	case study
Horngren et al. (2012); Hilton (2008)	Simple high-volume products tend to be OC. Complex low-volume products tend to be UC.	case study
Gupta (1993)	More products are UC then OC.	case study
Rezaie et al. (2008)	Volume-based costing systems report products that differ from the average product in the product mix with highly biased costs.	case study
Schmidt et al. (2023)	Under volume-based allocation heuristics, high-volume products are UC, while low-volume products are OC. However, this pattern occurs only if non-unit level costs are introduced. The pattern gets more dominant with an increasing proportion of non-unit level costs demand heterogeneity.	An increasing level of non-unit level activities (1-UNIT_SHARE) is positivly associated with the high-volume (OC) and low-volume (UC) pattern (VB_PATTERN)
Mertens (2020)	They find no evidence for the typical over-/under-costing pattern (high volume products OC, low volume products UC) under ABC.	No association between PE and product volume
	Products with a large proportion of unit-level costs (high UNIT_SHARE) tend to be OC. Products with a low proportion of unit-level costs (low UNIT_SHARE) tend to be UC.	UNIT_SHARE and PE are positively associated
	Non-trivial behavior of costing systems. Complex products can be over- and under-costed. High complexity in combination with a high cost share (high volume products) leads to OC.	UNIT_SHARE, INTER and INTRA decide whether products are UC or OC

5.3 Numerical Experiment

5.3.1 Data Generation and Measures

The experiment generates unique product family designs under varying levels of product variety, demand, and cost distributions. Figure 5.1 shows the five-step simulation procedure for this experiment. In the first step, 5000 individual EADs are created, each representing a single product family with a different product design, product mix, demand, and cost vectors. Appendix A5 shows the full design of experiments (DoE). Input parameters are chosen according to the empirical data presented in 3.5.3. Summarizing seven empirical studies, Mertens (2020) shows that 70% of a firm's total costs are direct, while 30% of the costs are indirect on average. However, case studies (e.g., Al-Omiri & Drury, 2007; Foster & Gupta, 1990; G. Miller & Vollmann, 1985; Wouters & Stadtherr, 2017) show a wide range across firms. In line with these studies, this work defines a range of $20\% \leq R_{id} \leq 80\%$ for the proportion of indirect costs on total costs. An empirical study by Ittner, Larcker, and Randall (1997) reports that 40% of firms total costs are fixed. In line with the numerical study by Schmidt et al. (2023), this work defines a range of $20\% \leq R_{fix} \leq 60\%$ for the fixed (non-unit level) costs on total costs. The last cost variable RC_{sdlog} describes the degree of resource cost disparity. Low values of RC_{sdlog} represent firms where resource costs are homogenously distributed across resources (low disparity), whereas high values represent environments where some resources are responsible for a large proportion of total costs (high disparity). In the second step, different levels of product variety are generated according to the procedure described in section 4.4.1. Remember, this procedure prefers introducing high-volume products before introducing low-volume products.

Figure 5.1:
Simulation procedure to investigate the effect of internal complexity on product costing system accuracy under different cost system designs.



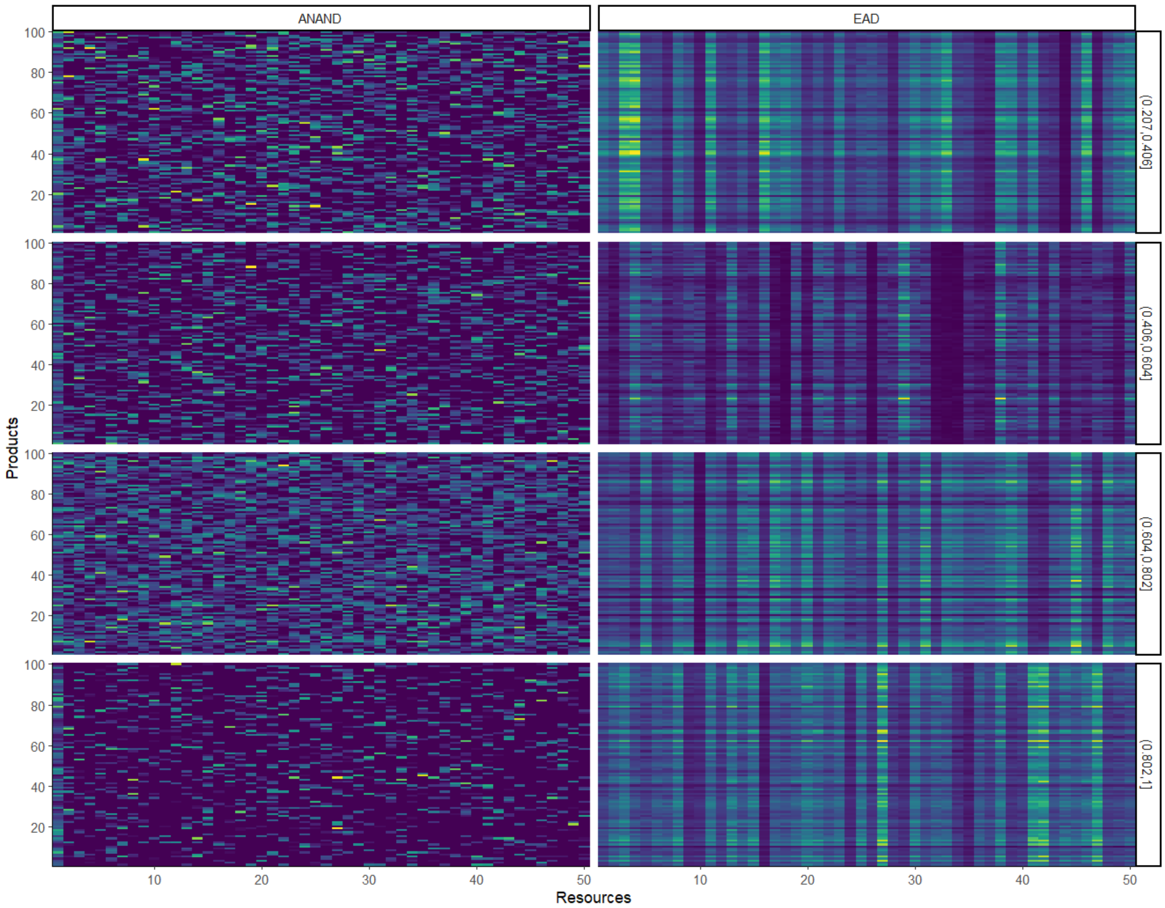
For each product family and product mix step (steps 1 and 2), the true benchmark costs (PC_B) are estimated under a full information environment. Applying different product costing systems leads to the reported costs (PC_H) for each step. In this experiment, three allocation methods are used. More complex ABC systems are modeled using the ‘random’ and ‘correl-random’ methods, as reported in Table 5.2. The third costing system represents a traditional volume-based PCS. The ‘DIV’ heuristic allocates indirect costs based on product units. In doing so, indirect costs are summed up and equally distributed across products. For the ABC-costing systems, the number of activity cost pools is varied between $2 \leq ACP \leq 30$ with a step size of two. Additionally, a PCS with $ACP = 1$ is added to compare the traditional ‘DIV’ heuristic with the ABC heuristics. The big-pool method is chosen as an activity driver for the ABC-heuristics, resulting in 33 product costing systems that are combined with 5000 product family designs. In the fourth step, measures for product costing accuracy are calculated. The total costing error ($EUCD$) and the mean absolute percentage error ($MAPE$) are calculated on the system level. At the product level, the percentage (PE) and absolute percentage errors (APE) are calculated.

5.3.2 Differences in Resource Consumption Pattern

Both the ABL and EAD models generate resource consumption patterns in different ways. Therefore, this section investigates whether differences in the generation procedure lead to differences in the resource consumption matrix. This is important since recent studies show that these patterns significantly impact product costing errors and their distribution across products (see Table 5.3). To compare the resource consumption patterns, the system-level data set, as generated in the previous section, is used to represent resource consumption patterns generated by the EAD. The ABL is used to generate correlation-based resource consumption patterns. In doing so, this work uses the design of experiments as reported by Schmidt et al. (2023). The only difference is the size of the resource consumption patterns defined by 50 resources ($N_{RD} = 50$) and 100 products ($N_{PROD} = 100$) to have matrices with identical dimensions across models. In total, 5000 unique resource consumption matrices are generated using the ABL.

The analysis is separated into two parts. In the first step, an exploratory analysis is conducted to visualize the main differences between both models. In the second step, descriptive statistics provide a more statistical and summarized description of model differences. Four resource consumption patterns are randomly selected and visualized for the exploratory analysis in Figure 5.2. Lighter colors represent a higher resource consumption, whereas darker tones indicate a small or no resource consumption. Each row in the figure shows matrices with approximately the same density (label on the right side). The ABL generates patterns by randomly sampling a uniform distribution, resulting in homogenous patterns. The EAD, however, generates patterns via an underlying design, resulting in more heterogenous (vertically oriented) patterns. Some resources show a high total resource consumption (TRC), while others show a small total resource consumption.

Figure 5.2:
Differences in resource consumption patterns. The color encodes the resource consumption.



Note. Each row compares resource consumption patterns with similar densities. The labels on the right side show the interval range of the randomly selected matrix.

Descriptive statistics (Table 5.5) highlight the differences between both models. The EAD shows higher average resource sharing than the ABL ($DNS_{RD,mean,EAD} = .989$ vs. $DNS_{RD,mean,ANAND} = .555$). This difference exists since resource sharing is an endogenous variable within the EAD model resulting from an underlying product design. Although the chance of generating low-density environments is smaller under the EAD, the model can also generate sparse resource consumption patterns ($DNS_{RD,min} = .381$). A second difference is the average correlation among resources (COR), which is higher under the EAD model ($COR_{mean,EAD} = .303$ vs. $COR_{mean,ANAND} = .044$). A high correlation indicates that if one product uses a large proportion of units for one resource, there is a high chance that it will also consume another resource by a high proportion. However, a high correlation does not necessarily imply more homogenous resource consumption patterns as it measures only relative differences across resources. Mathematically, correlation measures the angle between two vectors and neglects their absolute values. To highlight the absolute differences, the variation in total resource consumption is analyzed (TRC_{cv}). The total resource consumption (TRC) is defined as the column-wise sums of the resource consumption as:

$$TRC_j = \sum_{j=1}^{N_{RD}} P_{RD} \tag{5.6}$$

$$TRC_{cv} = \frac{sd(TRC)}{\overline{TRC}} \tag{5.7}$$

Differences in total resource consumption are observed by calculating the coefficient of variation (TRC_{cv}). The analysis shows that the variation in total resource consumption is higher under the EAD model ($TRC_{cv,EAD} = .684$ vs. $TRC_{cv,ANAND} = .190$). This result indicates that some resources are used frequently (by different products or at a low amount), while others are rarely used. The average inter- ($INTER_{RD,mean}$) and intra-product heterogeneity ($INTRA_{RD,mean}$) are reported to compare product-level differences. The analysis shows that the ABL generates more heterogeneous resource consumption patterns since the mean values are twice as high as for the EAD model ($INTER_{RD,mean,EAD} = 51.45$ vs. $INTER_{RD,mean,Anand} = 111.67$). Finally, the range of a product's total resource consumption is measured by $PROD_{RNG}$. In doing so, differences between the 95%- and 5%-quantile of products total resource consumption (row-wise sums of P_{RD}) are calculated. Although product-level measures indicate more homogeneous patterns under the EAD, product's total resource consumption range is around 40% higher under the EAD model ($PROD_{RNG,median,EAD} = 258$ vs. $PROD_{RNG,median,Anand} = 185$). This suggests that later product benchmark costs show a wider range for the EAD model, challenging volume-based allocation systems as they use a constant allocation base instead of product-individual drivers.

Table 5.5:
Descriptive statistics for both data sets

Variable	min	25%-quantile	mean	median	75%-quantile	max
			ABL			
DNS_{RD}	.212	.380	.555	.557	.725	.909
COR	.015	.023	.044	.030	.050	.277
TRC_{cv}	.045	.090	.190	.146	.262	.651
$INTER_{mean}$	32.43	53.94	111.67	86.29	152.99	333.46
$INTRA_{cmean}$	27.17	51.87	111.27	85.10	152.61	349.79
$PROD_{RNG}$	87.10	140.05	185.82	162.35	210.25	493.90
			EAD			
DNS_{RD}	.381	.995	.989	1.000	1.000	1.000
COR	.038	.200	.303	.300	.398	.638
TRC_{cv}	.440	.619	.684	.676	.739	1.147
$INTER_{mean}$	11.38	25.38	51.45	36.62	60.14	440.11
$INTRA_{cmean}$	6.26	22.34	59.78	38.97	71.97	639.68
$PROD_{RNG}$	24.10	136.15	258.51	224.18	349.21	909.65

In sum, results indicate that the EAD shows more homogenous patterns besides two exceptions. First, the variation in resources total consumption is higher, and products show a wider range in their total resource consumption. A detailed discussion of these results is done in section 5.3.5. The following two sections use the EAD-generated patterns, apply different product costing systems (PCS), and measure total costing as well as product-level costing errors.

5.3.3 System Level Analysis

In the second analysis, the costing error on the system level is analyzed for all three heuristics. Figure 5.3 shows the total costing error ($EUCD$) for each method under different numbers of activity cost pools (ACP). Since 'DIV' assigns all costs into one overhead pool, it is only available for $ACP = 1$. The exploratory analysis (Figure 5.3) shows that volume-based PCSs (DIV) cannot outperform more complex ABC systems with only one activity cost pool ($ACP = 1$), which contrasts with findings by Mertens (2020). The total costing error of the volume-based heuristic ($EUCD$) is around 47% higher compared to more complex ABC heuristics (see Table 5.6). Introducing more activity cost pools (ACP), the difference between ABC heuristics and the volume-based heuristic further increases. Table 5.6 shows that information-intensive ABC heuristics (cor-rnd) report more accurate product costs than simpler ones (rnd). However, these differences are smaller, as Balakrishnan et al. (2011) suggest. It is further observed that some cost pools are sufficient to reduce the error by a significant amount. Using eight activity cost pools reduces the total costing error by 53% for the random and 65% for the correlation-random heuristic.

Figure 5.3:
Total costing error, depending on the number of activity cost pools (ACP) for each allocation method.

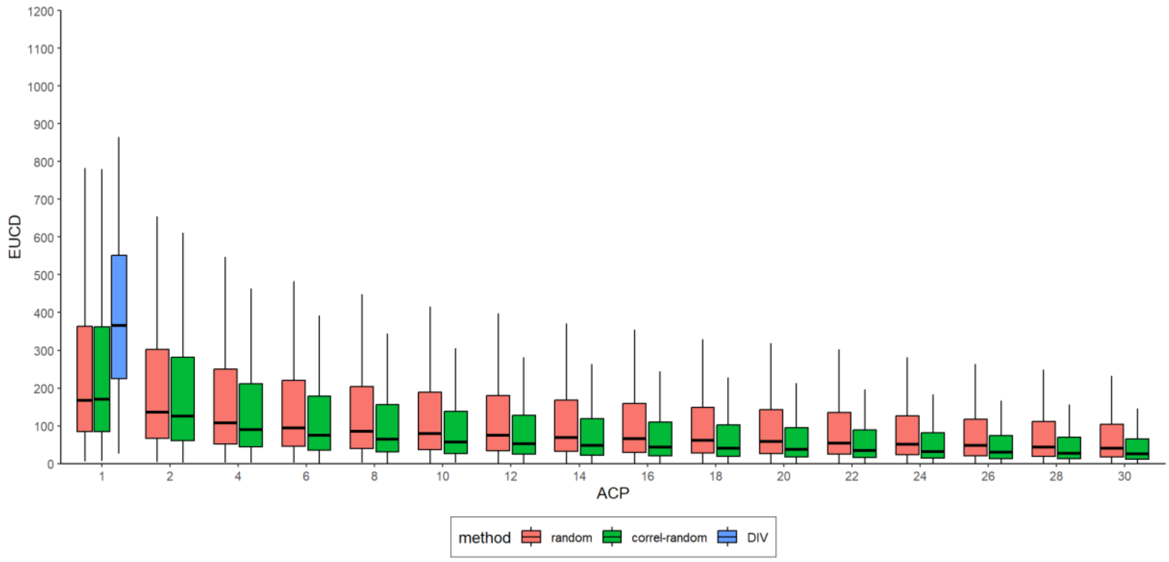


Table 5.6:
Median Error for each heuristic across different numbers of activity cost pools (ACP), the percentage differences in errors, and the percentage error for the DIV and random (rnd) heuristics compared to the correl-random (cor-rnd) heuristic.

	Median EUCD			Delta EUCD (ACP=1)			Delta DIV	
ACP	DIV	rnd	cor-rnd	DIV	rnd	cor-rnd	rnd	cor-rnd
1	366	192	193	0%	0%	0%	-47%	-47%
2		149	136		-22%	-30%	-59%	-63%
4		115	96		-40%	-50%	-69%	-74%
6		99	78		-48%	-60%	-73%	-79%
8		90	68		-53%	-65%	-75%	-82%
10		83	60		-57%	-69%	-77%	-84%
12		78	54		-59%	-72%	-79%	-85%
14		72	49		-63%	-74%	-80%	-86%
16		68	45		-65%	-77%	-81%	-88%
18		64	42		-67%	-78%	-83%	-88%
20		60	39		-69%	-80%	-84%	-89%
22		56	36		-71%	-82%	-85%	-90%
24		53	33		-73%	-83%	-86%	-91%
26		49	31		-74%	-84%	-87%	-92%
28		46	28		-76%	-85%	-88%	-92%
30		42	26		-78%	-87%	-89%	-93%

In order to generalize the effects on PCS's accuracy, three stepwise regression models are formulated for each of the three heuristics (nine models in total). These models represent different levels of available information firms can use to predict their total product costing error ($EUCD$). Table 5.7 summarizes the individual models where the first, less information-intensive model (Model 1) requires information on the product mix, cost structure, and the number of activity cost pools. The product mix is operationalized via the number of products (N_{PROD}), the demand skewness ($DMD_{T10\%}$), and the total demand ($TOTAL\_DMD$) [84]. The second model (Model 2) uses additional information on product families' physical design, operationalized via the level of component commonality (PCI_{PD}), the degree of inter-domain ($SDC_{N,FD,PD}$), and intra-domain coupling ($HIC_{N,PD}$)[85]. The third model uses additional information from the resource domain, such as the level of resource sharing (DNS_{RD}) and the similarity among resources (COR). Although section 3.6 suggests the diversification index (D_{RD}) as another resource domain measure, it is not included in the model as it causes collinearity problems[86].

[84] Mertens (2020) operationalizes demand skewness via Q_{var}. Nevertheless, Q_{var} and $DMD_{T10\%}$ are highly correlated ($c = .93, p < .05$).

[85] For a more detailed description of these measures, see section 3.4 .

[86] $VIF = 5.8$

Table 5.7:
Available information for the different regression models

Variable	Description	Model 1	Model 2	Model 3
	Product Mix			
N_{PROD}	number of products	x	x	x
$DMD_{T10\%}$	demand skewness	x	x	x
TOTAL_DMD	total demand	x	x	x
UNIT_SHARE	proportion of variable costs on total costs	x	x	x
$RC_{T10\%}$	resource cost pool disparity	x	x	x
R_{id}	proportion of indirect costs on total costs	x	x	x
	Costing System			
ACP	number of Activity Cost Pools	x[1)]	x	x
	Physical Domain			
PCI_{PD}	component commonality		x	x
$SDC_{N,FD,PD}$	degree of inter-domain coupling		x	x
$HIC_{N,PD}$	degree of intra-domain coupling		x	x
	Resource Domain			
DNS_{RD}	degree of resource sharing			x
COR	similarity among resources			x
D_{RD}	resource diversification			x

Note. 1) not for the DIV method

Table 5.8 reports the standardized effect sizes for each PCS and model. A positive effect size for product variety is observed under all models. Product variety has a small effect size ($\beta_{N_{PROD,cor-rnd}} = .167, p < .05$) under ABC systems and a medium effect size under volume-based systems ($\beta_{N_{PROD,DIV}} = .488, p < .05$). Larger effect sizes under the DIV heuristic than the ABC heuristics are also observed for the demand skewness and the total demand. Higher (static) demand diversity increases product costing errors ($\beta_{DMD_{T10,DIV}} = .673$ vs. $\beta_{DMD_{cor-rnd}} = .188$; $p < .05$), aligning with empirical observations (e.g., Cooper & Kaplan, 1988a; Horngren et al., 2012). The total demand shows a negative effect size under all PCS ($-.685 \leq \beta_{TOTAL\_DMD} \leq -.317, p < .05$). These findings suggest that volume-based costing systems are more sensitive to product mix changes compared to ABC systems, as relevant variables show higher absolute effect sizes.

A positive effect size for the proportion of indirect costs on total costs is observed ($.140 \leq \beta_{R\_ID} \leq .436, p < .05$). This matches intuition since more indirect costs must be allocated rather than traced. A second variable describing the cost structure is the proportion of variable indirect costs on total indirect costs where negative effect sizes are observed under all costing systems. ($-.362 \leq \beta_{UNIT\_SHARE} \leq -.086, p < .05$). This finding indicates that an increase in indirect variable costs is not as bad as an increase in indirect fixed costs. Again, the DIV heuristic is more sensitive to changes in these variables. However, the method is more robust to resource cost disparity. No significant effect for $RC_{T10\%}$ is observed under the DIV heuristic, whereas ABC systems show significant negative effect sizes ($\beta_{RC_{T10\%,rnd}} = 0.92; \beta_{RC_{T10\%,cor-rnd}} = 0.76;\ p < .05$). Volume-based costing systems show no effect size as they assign all resource costs into one cost pool. Finally, and confirming the study by Balakrishnan et al. (2011), the more information-intensive the ABC allocation, the more it profits from an increase in the number of activity cost pools ($\beta_{ACP,rnd} = -.148, p < .05$ vs. $\beta_{ACP,cor-rnd} = -.192, p < .05$).

Table 5.8:
Standardized regression effect sizes for the nine regression models.

	Model 1				Model 2				Model 3			
Predictor	**beta**	**std. error**	**t-value**	**p-value**	**beta**	**std. error**	**t-value**	**p-value**	**beta**	**std. error**	**t-value**	**p-value**
					DIV							
N_{PROD}	.488	.008	61.854	.000	.488	.008	62.296	.000	.488	.008	62.364	.000
$DMD_{T10\%}$	.673	.007	96.285	.000	.674	.007	97.229	.000	.675	.007	97.375	.000
UNIT_SHARE	-.362	.007	-55.103	.000	-.363	.007	-55.656	.000	-.362	.007	-55.559	.000
$RC_{T10\%}$	-.002	.006	-.275	.783	-.003	.006	-.434	.664	-.003	.006	-.442	.659
R_{id}	.436	.006	71.986	.000	.435	.006	72.341	.000	.436	.006	72.488	.000
TOTAL_DMD	-.685	.008	-84.069	.000	-.685	.008	-84.658	.000	-.686	.008	-84.877	.000
PCI_{PD}					-.152	.011	-14.055	.000	-.147	.012	-12.762	.000
$SDC_{N,FD,PD}$					.085	.009	9.325	.000	.067	.010	6.939	.000
$HIC_{N,PD}$					.055	.008	6.686	.000	.042	.009	4.910	.000
DNS_{RD}									-.031	.008	-3.861	.000
COR									.048	.009	5.346	.000
D_{RD}										VIF > 5		
$R^2_{adj.}$		51.7%				52.5%				52.6%		
Delta in R^2 to Model 3		98%				100%				100%		
					random							
N_{PROD}	.176	.002	106.470	.000	.174	.002	107.085	.000	.177	.002	111.308	.000
$DMD_{T10\%}$	.203	.002	134.654	.000	.203	.001	137.224	.000	.206	.001	141.849	.000
UNIT_SHARE	-.095	.002	-62.351	.000	-.092	.001	-61.511	.000	-.097	.001	-66.536	.000
$RC_{T10\%}$	.092	.001	63.483	.000	.092	.001	64.287	.000	.091	.001	65.178	.000
R_{id}	.151	.001	103.807	.000	.146	.001	102.346	.000	.145	.001	103.639	.000
TOTAL_DMD	-.341	.002	-207.182	.000	-.339	.002	-209.128	.000	-.339	.002	-213.839	.000
ACP	-.148	.001	-102.041	.000	-.148	.001	-103.742	.000	-.148	.001	-106.099	.000
PCI_{PD}					-.173	.003	-65.941	.000	-.067	.003	-24.382	.000
$SDC_{N,FD,PD}$					.015	.002	6.601	.000	.045	.002	19.536	.000
$HIC_{N,PD}$					-.002	.002	-.865	.387	.021	.002	10.430	.000
DNS_{RD}									-.094	.002	-48.794	.000
COR									-.163	.002	-75.896	.000
D_{RD}										VIF > 5		
$R^2_{adj.}$		16.4%				19.2%				22.7%		
Delta in R^2 to Model 3		72%				84%				100%		

Note. beta indicates the standardized regression coefficients; VIF<3 for all predictor variables

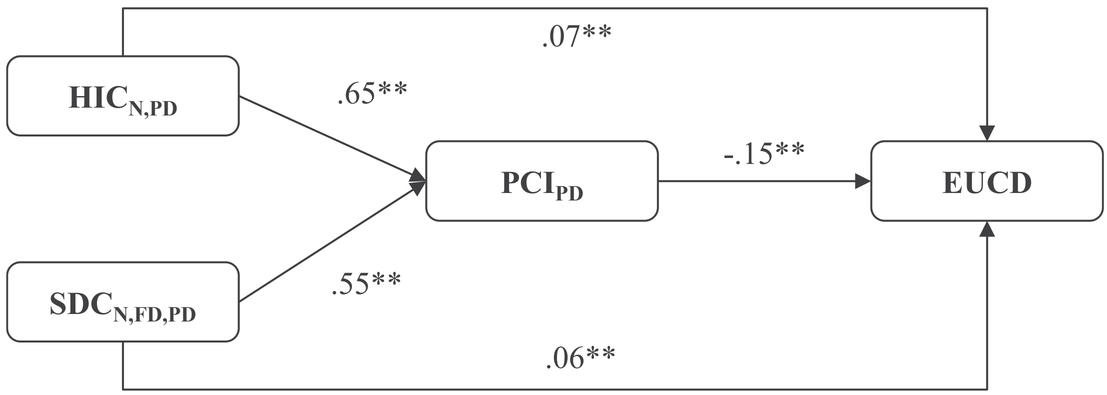
Table 5.8 (continued):
Standardized regression effect sizes for the nine regression models.

	Model 1				Model 2				Model 3			
Predictor	**beta**	**std. error**	**t-value**	**p-value**	**beta**	**std. error**	**t-value**	**p-value**	**beta**	**std. error**	**t-value**	**p-value**
					correl-random							
N_{PROD}	.167	.002	101.058	.000	.166	.002	101.488	.000	.168	.002	105.009	.000
$DMD_{T10\%}$	.188	.002	124.462	.000	.189	.001	126.658	.000	.191	.001	130.393	.000
UNIT_SHARE	-.086	.002	-56.174	.000	-.083	.002	-55.230	.000	-.088	.001	-59.554	.000
$RC_{T10\%}$	.076	.001	52.191	.000	.076	.001	52.738	.000	.075	.001	53.215	.000
R_{id}	.140	.001	96.444	.000	.136	.001	94.866	.000	.135	.001	95.703	.000
TOTAL_DMD	-.317	.002	-191.866	.000	-.315	.002	-193.345	.000	-.315	.002	-196.919	.000
ACP	-.192	.001	-132.068	.000	-.192	.001	-134.075	.000	-.192	.001	-136.580	.000
PCI_{PD}					-.165	.003	-62.683	.000	-.068	.003	-24.721	.000
$SDC_{N,FD,PD}$					.014	.002	6.102	.000	.042	.002	17.888	.000
$HIC_{N,PD}$					-.002	.002	-1.163	.245	.019	.002	9.159	.000
DNS_{RD}									-.085	.002	-43.954	.000
COR									-.149	.002	-69.002	.000
D_{RD}									VIF > 5			
$R^2_{adj.}$		15.9%				18.4%				21.3%		
Delta in R^2 to Model 3		74%				86%				100%		

Note. beta indicates the standardized regression coefficients; VIF<3 for all predictor variables

Model 2 adds the design measures to the regression models. Notable effect sizes for component commonality are observed under all PCSs. The differences in effect sizes are small across models ($-.173 \leq \beta_{PCI_{PD}} \leq -.152, p < .05$). The negative effect sizes indicate that increasing commonality decreases the product costing error. Therefore, component commonality is not only a lever to reduce costs (section 4) but also a lever to increase product costing accuracy. Product family design measured in the physical domain shows only relevant effect sizes under the volume-based heuristic ($\beta_{SDC_{N,FD,PD}} = 0.85; \beta_{HIC_{N,PD}} = 0.55;\ p < .05$). Effect sizes might indicate that volume-based costing systems are sensitive to changes in the product family design. However, this is not the case since inter- and intra-domain coupling affect component commonality. A path model is formulated to identify the direct and indirect effects of $SDC_{N,FD,PD}$ and $HIC_{N,PD}$ on the total costing error. Figure 5.4 shows the path model with the corresponding regression effects for the DIV heuristic, and Table 5.9 reports test statistics for all heuristics.

Figure 5.4:
Path model, showing the indirect effect of product design variables on the total costing error (EUCD) via the increase in component commonality (PCI_{PD}) for the random heuristic (rnd).



Note. ** indicates p<.05

The results of the path analysis underline the positive direct effect of inter- and intra-domain coupling on the total costing error under the DIV heuristic ($\beta_{HIC-EUCD,DIV} = .07, \beta_{SDC-EUCD,DIV} = .06; p < .05$). However, the effect is compensated by the indirect effect via component commonality ($\beta_{SDC,indirect,DIV} = -.079, \beta_{HIC,indirect,DIV} = -.096; p < .05$). Therefore, a neglectable total effect for design measures under the DIV heuristic is observed ($\beta_{SDC,total,DIV} = -.022, \beta_{HIC,total,DIV} = -.023; p < .05$). Under ABC heuristics, the direct effects of $SDC_{N,FD,PD}$ and $HIC_{N,PD}$ is neglectable and, therefore, a notable total effect size is observed for these heuristics (e.g., $\beta_{SDC,total,rnd} = -.102, \beta_{HIC,total,rnd} = -.101; p < .05$). In combination with an almost constant effect size for component commonality across all models, these findings allow the notion that ABC systems are sensitive to changes in design coupling, while the simpler volume-based heuristic is more robust.

Table 5.9:
Test statistics for the indirect effects of design variables on the total costing error.

Predictor	Y	DIV *b*	DIV *z-value*	DIV *p-value*	random *b*	random *z-value*	random *p-value*	correl-random *b*	correl-random *z-value*	correl-random *p-value*
					Regression					
$SDC_{N,FD,PD}$	PCI_{PD}	.547	637	<.05	.547	637	<.05	.547	637	<.05
$HIC_{N,PD}$		.647	753	<.05	.647	753	<.05	.647	753	<.05
$SDC_{N,FD,PD}$	EUCD	.057	5	<.05	-.004	-2	0.09	-.004	-2	0.07
PCI_{PD}		-.148	-9	<.05	-.180	-63	<.05	-.172	-60	<.05
$HIC_{N,PD}$		.073	6	<.05	.015	6	<.05	.014	6	<.05
					Indirect Effect					
$SDC_{N,FD,PD}$	EUCD	-.079	-9	<.05	-.099	-63	<.05	-.094	-60	<.05
$HIC_{N,PD}$	EUCD	-.096	-9	<.05	-.117	-63	<.05	-.111	-60	<.05
					Total Effect					
$SDC_{N,FD,PD}$	EUCD	-.022	-3	<.05	-.102	-65	<.05	-.098	-63	<.05
$HIC_{N,PD}$	EUCD	-.023	-3	<.05	-.101	-66	<.05	-.097	-62	<.05

Note. All models are significant at p<.05; Fit indices for the DIV model: CFI=1; TLI=1, RMSEA=.000; Fit indices for the random model: CFI=1; TLI=1, RMSEA=.000

Finally, resource domain variables are added for the last and most information-intensive model (Model 3). A negative effect size of resource sharing is observed under all heuristics ($-.094 \leq \beta_{DNS_{RD}} \leq -.031$), confirming earlier findings (Balakrishnan et al., 2011; Homburg et al., 2018). Again, ABC systems are more sensitive to changes in resource sharing. A positive effect size for the resource correlation was observed under the DIV heuristic ($\beta_{COR} = 0.048, p < .05$). Although the effect size is small, it differs from the negative effect sizes under the ABC systems (e.g., Balakrishnan et al., 2011). The positive effect size occurs since volume-based heuristics use an averaged activity driver for all products. Thus, an increased resource correlation increases the differences in resource usage among products, leading to higher total product costing errors.

The stepwise regression models show that increased levels of available information increase the explained variance ($R^2_{adj.}$). Although the volume-based system has the highest errors, it shows the best predictability ($R^2_{adj.,Model\ 3} = 52.6\%$). Interestingly, the explained variance reduces only slightly when information is dropped from the model ($R^2_{adj.,Model\ 1} = 51.7\%$). These differences are much higher under ABC systems. For example, the differences in the explained variance between Model 3 and Model 1 ($\triangle\ R^2_{adj.}$) are estimated as $\triangle\ R^2_{adj.,rnd} = 28\%$ for the random heuristic and $\triangle\ R^2_{adj.,cor-rnd} = 26\%$ for the correlation-random heuristic. These findings indicate two aspects. First, ABC systems are more difficult to predict compared to volume-based systems. Second, when using simple costing systems, information on the cost structure and the product mix is sufficient for a good prediction of the total product costing error. A discussion of this section's results is done after reporting the product-level results in the next section.

5.3.4 Product Level Analysis

In the third analysis, product-level errors are investigated. According to Labro and Vanhoucke (2007), a product is under-costed ($UC = 1$) if reported costs are below 5% of its benchmark costs ($PC_H < PC_B * 0.95$) and is over-costed ($OC = 1$) if $PC_H > PC_B * 1.05$. The proportion of under- (over-) costed products is defined as PUC (POC). Volume-based allocation (DIV) and ABC heuristics are compared for $ACP = 1$ for a fair comparison. Under one activity cost pool, there is no difference between the correlation-random and random method (see Table 5.2). Therefore, this section compares the volume-based heuristic (DIV) with the random ABC heuristic (rnd). As introduced, the product level error is measured using the percentage error (PE) and the absolute percentage error (APE). If $PE > .05$ products are under-costed and over-costed if $PE < -.05$ [87].

Descriptive statistics (Table 5.10) show that products have an average costing error of $22.2\% \leq APE_{DIV} \leq 32.3\%$ under the volume-based allocation, which is higher compared to the ABC system ($14.9\% \leq APE_{DIV} \leq 15.9\%$) with one activity cost pool. While more products are under-costed than over-costed under the volume-based systems ($PUC_{DIV} = 49.8\%$ vs. $POC_{DIV} = 32.0\%$), there is almost no difference under ABC systems ($PUC_{rnd} = 24.2\%$ vs. $POC_{rnd} = 23.1\%$). The general dominance of under-costing appears as well, although differences are small. This contrasts with earlier findings by Labro and Vanhoucke (2007), who note the dominance of under-costed products for ABC systems. Since more products are under-costed, over-costed products show a higher average product costing error under the DIV system ($APE_{DIV,OC} = 32.3\%$ vs. $APE_{DIV,UC} = 22.2\%$).

Table 5.10:
Descriptive statistics for the absolute percentage error (APE) under the volume-based allocation heuristics (DIV) and the random ABC heuristic (rnd) with one activity cost pool (ACP=1).

Bias	min	25%-quantile	mean	median	75%-quantile	max	sd	P[1)]
				DIV				
OC	.050	.117	.323	.221	.422	1.525	.290	.320
UC	.050	.108	.222	.179	.294	.878	.150	.498
				rnd				
OC	.050	.072	.159	.107	.180	1.523	.155	.231
UC	.050	.072	.149	.106	.175	.926	.123	.242

Note. [1)] indicates the proportion of products being under-costed (PUC) or over-costed (POC)

Inter- ($INTER_{PD}$) and intra-product heterogeneity ($INTRA_{PD}$) are used as measures for product characteristics. In contrast to existing studies, these measures are estimated using the physical domain. In contrast to the resource domain, the physical domain is easier to observe by analyzing the bill of materials for the products. However, section 3.4.3 argued that these measures do not always reflect the isolation of a product adequately. Therefore, the local outlier factor (LOF_{PD}) is introduced. Values of $LOF_{RD} > 1$ indicate that a product is isolated from its neighbors. Values of $LOF_{RD} < 1$ indicate that a product is more similar compared to its neighbors as the product mix average. Additionally, DMD_{perc} is defined as the proportion of products' demand on total demand. High-volume products, therefore, are characterized by high values of DMD_{perc}, whereas low-volume products by low values. Four regression models are formulated to identify characteristics of products with biased costs. The first two models analyze relevant characteristics for the percentage error (PE), and the second two analyze the average percentage error (APE). Regression statistics are reported in Table 5.11.

[87] To avoid confusion, the percentage error for a product j is defined as $PE_j = (PC_{B,j} - PC_{H,j})/PC_{B,j}$. This differs from the work by Schmidt et al. (2023), who define as $PE_j = (PC_{H,j} - PC_{B,j})/PC_{B,j}$.

Table 5.11:
Regression results to identify product-level characteristics of biased product costs.

Measure	Predictor	beta	std. error	t-value	p-value	Fit
			DIV			
	UNIT_SHARE	-.113	.001	-96.744	<.05	
	R_id	.026	.001	21.861	<.05	
	DMD_T10	.219	.001	190.065	<.05	
	DMD_perc	-.320	.001	-276.992	<.05	34.3%
	LOF_{PD}	.262	.001	202.958	<.05	
	$INTER_{PD}$	.112	.001	80.127	<.05	
	$INTRA_{PD}$	-.314	.001	-244.945	<.05	
PE			*rnd*			
	UNIT_SHARE	.001	.001	.356	1.000	
	R_id	.004	.001	2.457	.722	
	DMD_T10	.000	.001	.147	<.05	
	DMD_perc	-.002	.001	-1.207	.883	0.2%
	LOF_{PD}	.005	.002	3.121	.227	
	$INTER_{PD}$	-.008	.002	-4.703	<.05	
	$INTRA_{PD}$	.047	.002	31.194	<.05	
			DIV			
	UNIT_SHARE	-.184	.001	-140.990	1.000	
	R_id	.299	.001	228.960	<.05	
	DMD_T10	.133	.001	103.606	<.05	
	DMD_perc	.053	.001	41.273	<.05	18.5%
	LOF_{PD}	-.039	.001	-27.085	<.05	
	$INTER_{PD}$	-.046	.002	-29.538	<.05	
	$INTRA_{PD}$	.252	.001	176.601	<.05	
APE			*rnd*			
	UNIT_SHARE	-.039	.001	-30.420	1.000	
	R_id	.238	.001	184.539	<.05	
	DMD_T10	.000	.001	-.266	<.05	
	DMD_perc	.007	.001	5.335	.790	19.6%
	LOF_{PD}	-.048	.001	-33.829	<.05	
	$INTER_{PD}$	.123	.001	83.320	<.05	
	$INTRA_{PD}$	.318	.001	234.517	<.05	

Note. All models are significant at p<.05; controlled for covariation VIF<3 for all predictors.

Under the volume-based heuristic (DIV), the typical pattern of high-volume products being over-costed and low-volume products being under-costed is observed ($\beta_{DMD_{PERC},DIV} = -.320, p < .05$), which aligns with the literature. (Cooper & Kaplan, 1988a; Gietzmann, 1991; Hilton, 2008; Horngren et al., 2012). For the remaining product level measures, it is observed that product distance from the center is an indicator of under-costing ($\beta_{INTER_{PD},DIV} = .112, p < .05$). An even stronger effect size is observed for the local outlier factor ($\beta_{LOF_{PD},DIV} = .262, p < .05$). This allows the notion that exotic products tend to be under-costed and confirms empirical observations in the product management literature (e.g., Schuh, 2005). By analyzing production complexity, intra-product heterogeneity shows a negative effect size ($\beta_{INTAR_{PD},DIV} = -.314, p < .05$), indicating that products with diverse resource consumption tend to be over-costed. In contrast, products with homogeneous resource consumption are more likely to be under-costed. For the ABC system, no notable effect size is detected. For example, a study by Schmidt et al. (2023) demonstrated that the typical pattern of high-volume-low-volume products does not exist for ABC systems. These findings suggest that product over- and under-costing cannot be characterized using the existing measures. While ABC systems do not show any relevant effect sizes for the percentage error, notable effect sizes for two product characteristics are observed for the absolute percentage error (APE). Both increasing inter- and intra-product heterogeneity are positively associated with APE ($\beta_{INTRA,rnd} = .123, \beta_{INTRA,rnd} = .318; p < .05$). It indicates that these measures are at least indicators for products with biased costs. However, it does not allow estimating whether these products are under- or over-costed. The following section discusses the results of this section.

5.3.5 Discussion

5.3.5.1 Analysis of Model Differences

The discussion is separated into three parts according to the three analyses. The first analysis investigates differences between the Anand framework (ABL) and the numerical EAD model. In contrast to the Anand model, the EAD produces high-density environments on average, representing shop floor rather than job shop environments (Balakrishnan et al., 2011). Under the EAD, the level of resource sharing is an endogenous variable as it is the result of product family design decisions. Therefore, the model rarely generates low-density environments. Nevertheless, the analysis has shown that the model can generate such patterns, even if the probability is low. This finding indicates that EAD-generated patterns represent job shop environments (such as mass customizing firms). The average correlation among resources (COR) is another difference between both models. It is observed that the EAD generates more correlated resource consumption patterns. The average inter- and intra-product heterogeneity is also lower in the EAD model. In sum, results suggest that the EAD generates more homogenous than those created by the ABL. However, the increased homogeneity in resource consumption under the EAD matches expectations since the product mix contains products belonging to one product family. Therefore, these products are variants of each other. Buchholz (2012) notes that products are variants of each other if they share more similarities than dissimilarities.

Although these measures indicate more homogenous patterns generated by the EAD, two variables indicate an increased heterogeneity. First, the diversity in total resource consumption (TRC_{cv}) is higher under the EAD, indicating that some resources are frequently used while others are only rarely used. This fits into the idea that the EAD represents mass customization environments where, for example, a large part of the value creation takes place along a standardized process (e.g., an assembly line) and only a tiny part through pre-assembly or after postponement[88]. Development activities are another example of resources frequently used across all products. Product families are developed as one rather than each product individually. These findings underline the idea of a base resource as used in the ABL framework. The second difference is the increased range between the 5%-least resource-consuming products and the 95%-most consuming ones. This leads to a broader range in product costs, especially challenging for volume-based allocation heuristics as they use an averaged driver rather than individual product drivers compared to ABC systems.

[88] According to Swaminathan and Lee (2003), postponement describes the point in the supply chain at which products are personalized. To increase flexibility, firms aim to move the point of personalization as late as possible in the value chain.

The system-level analysis confirms this notion by finding that the volume-based allocation outperforms ABC systems only under one activity cost pool ($ACP = 1$), whereas Mertens (2020) notes that they outperform these up to $ACP = 4$.

In sum, the analysis finds that products share more similarities under EAD-generated resource consumption patterns. Although these patterns seem more homogenous initially, products show a wider range regarding their total resource consumption, increasing the range of true benchmark costs. While the ABL framework is more general, as it also allows for producing job shop environments, the EAD model focuses on representing mass-customized shop floor environments where products are derived from an underlying design and belong to one product family. Besides the statistical differences in the resource consumption patterns, the model differs in how it models non-unit level costs. The EAD takes the traditional costing view as it differs solely between unit-level (variable) and non-unit-level (fixed) costs where the ABC-cost hierarchy differs between unit-, batch-, product-, and facility-level resources (Balakrishnan et al., 2012). The latest studies separated the resource consumption pattern into different column-wise blocks, each representing one level of the cost hierarchy. For example, the study by Schmidt et al. (2023) generates unit-level resources according to the ABL procedure for the first resources and randomly samples batch-level resource consumption for the remaining ones. However, Labro (2018) notes difficulties in the cost hierarchy, for example, when setup costs are a function of the unit-level volumes, such as under the economic order model (see section 4.3.2.1). Therefore, this work takes the simpler, traditional cost perspective as it differs solely between unit (variable) and non-unit level (fixed) costs. Another difference to the work by Schmidt et al. (2023) is that the resource consumption pattern is not separated into two parts (unit- and non-unit-level resources). Instead, two resource vectors ($\mathrm{RC_{var}}, \mathrm{RC_{fix}}$) are generated. If both vectors are highly correlated with each other, unit- and non-unit-level activities are correlated. Otherwise, they are independent.

5.3.5.2 System Level Analysis

Using the EAD framework, the system-level analysis shows a mixed picture of earlier findings. It confirms that an increased demand skewness ($DMD_{T10\%}$), increased resource cost disparity ($RC_{T10\%}$), and decreased levels of resource sharing (DNS_{RD}) drive errors in product costing systems, confirming earlier studies (Balakrishnan et al., 2011; Homburg et al., 2018; Mertens, 2020). Since demand skewness has notable effect sizes under all PCS, it indicates that knowledge of later demand is essential when designing costing systems, which contrasts with Banker and Hughes (1994). It is further observed that information-intensive ABC heuristics (e.g., correl-random) outperform simpler ones (e.g., random) and that some activity cost pools are sufficient to significantly reduce the error, which is in line with findings by Balakrishnan et al. (2011). A negative association between resource similarity (COR) and the costing error was only observed under ABC systems. Volume-based systems show a positive effect size. An explanation is that under such systems, increased resource similarity further increases product differences, leading to more errors due to the broad averaging. ABC systems, on the other hand, can profit from increased resource similarity as they use product-individual activity drivers. In contrast to Mertens (2020) and

Feltham (1977), this study cannot confirm that simple volume-based costing systems outperform more complex ABC systems. The reason for this result is that this experiment creates demand vectors that are more heterogeneous compared to Mertens's study.

An advantage of using the EAD is linking product family design with product costing errors. Under ABC systems, negative effect sizes for component commonality are observed. It allows the notion that component commonality is not only a lever to reduce costs but also a lever to increase product costing accuracy when using ABC. Although negative coefficients are observed under the volume-based system, a path analysis reveals that the effects of component commonality are compensated by inter- and intra-domain coupling since they also affect component commonality. In contrast to the existing studies, this work investigates the effect of product variety and total demand on the total product costing error. Under all PCSs, a positive effect of product variety on $EUCD$ is observed and a negative effect on the total demand as it increases the proportion of unit-level (variable) costs ($UNIT\_SHARE$) on indirect costs.

The comparison between ABC and volume-based systems highlights that ABC systems are more challenging to predict compared to volume-based systems. ABC systems are more sensitive to changes in the product family design. For example, component commonality (PCI_{PD}), the degree of resource sharing (DNS_{RD}), and the correlation among resources (COR) show larger effect sizes under these systems. Under the volume-based system, the effects are much smaller since such heuristics do not use design information during the allocation process. Remember, volume-based systems use a broad averaging instead of activity drivers. However, they are more sensitive to product mix variations (product variety, total demand, demand skewness) and cost structure, whereas ABC systems are more robust. The resource cost disparity ($RC_{T10\%}$) is an exception, where volume-based costing systems show no notable effect size. The reason is that all indirect costs are assigned to one cost pool under these systems. The sensitivity of simple PCSs is also underlined by empirical studies, which find that firms tend to use ABC under increasing product variety (Malmi, 1999; Schoute, 2011). The effect of demand skewness on the total costing error is more than three times higher under volume-based systems. This is critical as it leads to a tiptoeing decrease in PCS's accuracy over time when using volume-based systems[89]. Usually, firms launch a product family by focusing on the main markets first. The result is a product mix, mainly consisting of high-volume products (low demand skewness). Aiming to increase profitability further, they introduce more and more niche products over time. Since those products address a small market, their total volumes are homeopathic. As a result, demand skewness increases, and PCS accuracy decreases. While the product costing error was acceptable at the launch of a product family, it can become a serious issue for firms over time.

Table 5.12 connects the findings of this section with the concept of internal complexity. Except for some cost structure and cost system variables ($UNIT\_SHARE$, R_id and ACP), all remaining variables are related to one dimension of complexity. Increased diversity in

[89] To be precise, this effect also occurs under ABC systems. However, effect sizes are much smaller.

terms of demand skewness drives product costing errors under the volume-based heuristic. A mixed effect of multiplicity is observed. An increasing number of products increases the costing error since more costing objects exist. On the other hand, increasing product reuse (more demand at constant product variety) decreases the costing error as it decreases the proportion of indirect non-unit level costs. In sum, volume-based systems depend on the product mix and cost structure but not on the product family design since these effects compensate each other. ABC systems show smaller effect sizes for all variables. However, they depend on all three dimensions of complexity (diversity, multiplicity, interrelatedness) since the effects of inter- and intra-domain coupling do not compensate for the effect of component commonality. Therefore, ABC systems depend on the product mix, the cost structure, and the product design. These findings extend the notion that ABC systems are not only associated with fewer product costing errors compared to volume-based systems (e.g., Cooper & Kaplan, 1988b) but also with an increased robustness to product mix changes. However, they are more sensitive to design changes.

Table 5.12:
The effects of internal complexity on product costing system's total error (EUCD)

Variable	Description	Dimension[1)]	Volume-based	ABC	
			DIV	**random**	**correl-random**
		Product Mix			
$DMD_{T10\%}$	Demand skewness	D	+++	++	+
N_{PROD}	Product variety	M	+++	+	+
TOTAL_DMD	Total demand	M	---	--	--
		Cost Structure			
$RC_{T10\%}$	Resource cost diversity	D	*	+	+
UNIT_SHARE	Proportion of variable costs on indirect costs	-	--	-	-
R_{id}	Proportion of indirect costs on total costs	-	+++	+	+
		Product Design			
PCI_{PD}	Component commonality	D	*[2)]	-	-
$SDC_{N,FD,PD}$	inter-domain coupling	I	*[2)]	(-)	(-)
$HIC_{N,PD}$	intra-domain coupling	I	*[2)]	(-)	(-)
DNS_{RD}	Level of resource sharing	D	*	-	-
COR	Similarity among resources	D	*	-	-

Note. +/- indicates $.05 < |\beta| \leq .2$, $++/--$ indicates $.2 < |\beta| \leq .4$ and $+++/---$ indicates $|\beta| > .4$; $*$ indicates that $|\beta| < .04$; [1)] M = multiplicity, I = interrelatedness, D = diversity; [2)] the positive effect of inter- and intra-domain coupling is compensated by component commonality; (-) indicates an indirect effect.

5.3.5.3 Product Level Analysis

Four product-level variables are defined to characterize products with biased product costs. $INTER_{PD}$ and $INTRA_{PD}$ represent inter- and intra-product heterogeneity (see M. Gupta, 1993). In contrast to the original definition, inter- and intra-product heterogeneity are measured in the physical domain rather than the resource domain. The reason is that the physical domain is easier to observe by analyzing the bill of materials. In addition, the local outlier factor (LOF_{PD}) is introduced to identify products placed in isolation within the product mix. These products are called exotic. The last product measure (DMD_{perc}) differentiates whether the product is low- or high-volume.

The product level analysis confirms that more products are being under-costed than over-costed under volume-based systems. The typical pattern of high-volume products being over-costed and low-volume being under-costed is identified since a negative correlation between DMD_{perc} and the percentage error (PE) is observed. However, it cannot confirm these statements for ABC systems. These results underline recent findings by Schmidt et al. (2023). However, they contrast with studies by M. Gupta (1993) or Labro and Vanhoucke (2007). The analyses further find that volume-based systems under-cost products placed at the border of the product mix are under-costed as well as exotic products. Since the effect size for the local outlier factor is twice as high as for inter-product heterogeneity, it seems to be a better measure for the identification of under-costed products. These findings confirm empirical observations in the product management and engineering literature (e.g., Feldhusen et al., 2007; Rebentisch et al., 2016; Schuh, 2005; Schuh et al., 2014). In contrast, production complexity ($INTRA_{RD}$) indicates over-costing under volume-based systems. These results extend empirical observations by Rezaie et al. (2008), who note that volume-based allocation methods report highly biased costs for products that differ from product mixes' average. However, their study limits product differences to their demand and geometric size and does not investigate the under-/over-costing explicitly.

As the system-level results suggest, the characterization of product-level errors is much more challenging under ABC. No notable effect sizes for the percentage error (PE) are observed. It indicates that under- and over-costing appear randomly under ABC, or selected measures do not reflect the relevant product characteristics. While the existing set of measures does not allow the identification of under or over-costed products, it allows the identification of products with an absolute costing error (APE). A positive effect size for production complexity ($INTRA_{RD}$) is observed, suggesting that these products have more biased product costs than those with low production complexity. In sum, the product-level analysis underlines the system-level results. The behavior of volume-based costing systems is much better to predict, and rules of thumb exist, while the behavior of ABC systems is more complex and non-straightforward confirming Mertens (2020). This underlines studies that argue that firms refuse ABC adoption due to the complexity of these systems (Gosselin, 2006; Innes et al., 2000).

5.4 Conclusion

This section investigates the effects of internal complexity on product costing system accuracy. The EAD simulation model is used to create 5000 unique product family designs and combine them with 33 product costing systems (PCS). While one PCS represents a simple volume-based allocation heuristic, the others represent more sophisticated ABC systems with different degrees of sophistication. A central aspect of PCS experiments is the resource consumption pattern, which allows for estimating a product's true benchmark product costs under full information. In contrast to the existing ABL framework, resource consumption results from an underlying product design rather than random correlation. Due to creation procedure differences, resource consumption matrices differ across models. In the first analysis, these differences are investigated, where this study finds that the EAD generates more homogenous matrices in general. For example, EAD matrices are characterized by a higher degree of resource sharing, a higher average resource correlation, and less inter- and intra-product heterogeneity. Results underline that the EAD model represents products that are part of the same product family (variants of each other). However, there are two aspects in which the EAD produces matrices with higher heterogeneity. The variation in total resource consumption is higher under the EAD, indicating that some resources are used by a large amount (e.g., central manufacturing lines) while others are only rarely used (e.g., customizing activities). A second difference is the total resource consumption of products. It is observed that the range between products having a small total resource consumption and those that consume many units is more prominent under the EAD. The result is a broader range of true benchmark costs and increased aggregation errors under volume-based heuristics.

The error between a product's true benchmark costs and reported costs is calculated for each product costing system. This study cannot confirm that simple volume-based heuristics outperform more complex ABC systems, contrasting with Mertens (2020). Although ABC systems are more accurate, they are more sensitive to product design changes. On the other hand, volume-based systems are sensitive to changes in the underlying (static) demand distribution. It is observed that these systems report significantly higher errors if the demand vector is highly skewed with some high-volume and many low-volume products. On the product level, this study confirms the typical pattern of high-volume products being over-costed and low-volume being under-costed for volume-based allocation heuristics. Additionally, products' distance from an imaginary average product and its exoticness indicates under-costing under volume-based allocation. Products characterized by a high production complexity tend to be over-costed under such systems. These patterns occur only under volume-based systems and are absent under ABC. This study can only confirm that increasing production complexity is associated with an increasing absolute error for ABC heuristics. However, it is not possible to identify characteristics of under- or over-costed products for such systems.

Findings gained in this experiment have several implications for research and practice, where Figure 5.5 provides an overview. Labro and Vanhoucke (2007) note that, in practice, PCS's design is always a trade-off between accuracy and the costs of accuracy. Although ABC systems report accurate product costs, they come with high efforts for data acquisition and processing as well as a high level of complexity for the users. Thus, empirical studies find a low adoption rate (<40%) of ABC systems (see Table 5.1). On the other hand, volume-based costing systems are less accurate; however, they allow for a better prediction of product-level errors. Therefore, the work calls for a stronger focus on simpler PCS in cost accounting research in general. Specifically, identified rules of thumb can be used to improve the performance of these systems. For example, an improved volume-based heuristic could use information on products' exoticness to notify users of cost information that these products have a high chance of being reported with biased product costs. Adding this information is relatively simple to implement in modern ERP systems, and managers can take further actions. A second implication is related to the method of numerical experiments. Usually, empirical cost accounting studies are rather general by conducting a survey among many firms (100 -2800, see Table 5.1) or limited to a single firm and product family for in-depth analysis (Cooper & Kaplan, 1988a; Gietzmann, 1991; M. Gupta, 1993; Horngren et al., 2012; Shea, Waldrup, Xu, & Williamson, 2018; Wouters & Stecher, 2017)[90]. While many rather general studies investigate product costing system choices, hurdles, and reasons for (non-)implementation, in-depth studies investigating the error in product costing systems are rare. A reason for the low number of studies is the difficulties in data collecting, as it requires knowledge of the true resource consumption patterns. Although numerical experiments cannot replace empirical studies, they have two advantages. First, they allow for the creation of a large variety of empirical-related resource consumption patterns, and second, the analyst has complete control over the observation level (product- / system-level) and can manipulate input variables to see models' responses. Therefore, several studies investigating product costing systems using numerical experiments were published over the last 20 years (e.g., Balakrishnan et al., 2011; Labro & Vanhoucke, 2007; Schmidt et al., 2023). These studies use the correlation-based approach of the ABL framework for generating resource consumption patterns. In line with recent studies (Mertens et al., 2021; Meßerschmidt et al., 2020), this work challenges this approach by generating resource consumption patterns via an underlying design. It is argued that in mass customization environments, products are configured based on a set of basic elements (functional requirements, components). Thus, products are variants of each other, sharing more similarities than dissimilarities. The EAD-generated patterns reflect this as they show less inter- and intra-product heterogeneity compared to the ABL framework. However, they show a broader range in products' total resource consumption, resulting in a larger range of product costs. Using the EAD procedure and integrating empirical boundaries for model input variables, this work is a further step towards generating more empirical-related resource consumption patterns for product families in a mass customization environment. It is important to note that these patterns are still empirical-related as it is assumed that integrated theories (Axiomatic Design, Leontief production function) are sufficient to represent resource consumption patterns in practice. Nevertheless, using numerical simulations in combination with empirical studies to validate assumptions and verify results might be a fruitful direction for future research. By doing so, this study reduces the gap between cost accounting research and practice (e.g., Brierley et al., 2001) and follows the call for more interdisciplinary research (e.g., Labro, 2015a) as it connects the fields of engineering design and cost accounting.

[90] Not all studies investigated the errors in product costing systems, limiting the number of in-depth studies in this work's context.

Figure 5.5:
Summary of implications for research and practice gained by this experiment.

Research

- Search for improved simple volume-based heuristics so they can compete under more conditions with ABC but at significantly lower costs.
- Use more numerical experiments in cost accounting research as they allow the generation of empirical-related resource consumption patterns that are difficult to observe in practice.

Practice

- Be aware of the tiptoeing increase of product costing errors over the product life cycle when using volume-based product costing systems.
- Be aware of niche products as they tend to be under-costed under volume-based allocation systems. Verify that those products are a true business opportunity.

Besides the implications for researchers, this study has two practical implications. The first implication is related to the error-increasing effect of product variety and demand skewness and states that product managers should be aware that each product mix decision affects the product costing accuracy. Due to limited resources, a product family is launched with a limited number of product configurations, focusing on core markets. In order to address niches and react to competitors (e.g., Debruyne et al., 2002), new products are introduced after the initial launch. The result is a tiptoeing decrease in product costing accuracy when using volume-based systems as the demand skewness and product variety increase. Although the effect also occurs under ABC systems, it is much smaller. A vicious circle starts where niche products (under-costed) are cross-subsidized by high-volume products placed in the product mix center. In the belief that these products are highly profitable, new niche products are introduced, further increasing cross-subsidization. Following the first implication, the second implication states that decision-makers should carefully decide whether niche products are true business opportunities or burn money and standardized products are unidentified gold mines.

This study comes with some limitations. First, the model differs solely between unit- and non-unit-level costs, representing the traditional costing view and neglecting the ABC hierarchy. A second limitation is that the model generates a single resource consumption pattern rather than column-wise resource consumption blocks, as done by (Schmidt et al., 2023). This means that unit- and non-unit-level activities are not entirely independent. Third, the EAD mainly produces shop floor environments (high density). Low-density environments are rarely generated, limiting models' fields of application. A reason for the high density is that the model still assumes simple implications between elements. More complex interactions, for example, by using Boolean logic (e.g., if 'A' AND 'B', then NOT 'C'), need to be integrated.

Future research can tackle these limitations, bringing engineering design, product management, and cost accounting communities closer together and improving EAD's ability to generate more empirical-related patterns. Specifically, future research can add the ABC cost hierarchy to the numerical EAD model. They can extend the resource consumption pattern generation process to model resource consumption independently at each level of the cost hierarchy. However, hierarchical levels should still be the result of an underlying design.

6 Overall Conclusion

Faced with external complexity, firms respond with a specific product variety realized via a product family design, which induces internal complexity. This thesis investigates the economic effects of internal complexity. Specifically, it investigates how internal complexity affects firm's total costs and distorts product costing. In doing so, numerical experiments are chosen as a research method due to several advantages. First, they produce their own data and are not restricted to empirical observations. For example, the study creates a large variety of product family designs and product costing systems. Numerical experiments can generate observations that are challenging to get in practice. True product costs, for example, require full information settings, and firms are interested in keeping product family designs secret as they would give competitors an advantage. Finally, numerical experiments allow the observation of complex system interactions.

The thesis starts with operationalizing internal complexity in the context of product family design. Several studies build the foundation for this part, such as Mertens's (2020) conceptual Extended Axiomatic Design, the review by Trattner et al. (2019), and the numerical experiments by Hennig et al. (2022) in which the authors investigate product family complexity measures. Section 3 connects these studies. In the first step, the operationalized EAD simulation framework is introduced, enabling the generation of configurable product family designs. Similarities among product family complexity measures are investigated by evaluating the measures for each of the generated product family designs. This study confirms the scattered state in the context of product family complexity measures, as many measures are highly correlated (see Hennig et al., 2022). These findings raise the critical question of whether new complexity measures are necessary or whether research should focus on consolidating existing measures first. Another finding is that existing measures do not represent topological properties well. Although literature introduces topological measures, they show a high correlation with simple count-based measures and, therefore, have only a small amount of additional information. Finally, the work suggests a set of product family complexity measures representing the three dimensions of systemic complexity (diversity, multiplicity, interrelatedness) and checks their content and construct validity. In sum, this section connects the literature on product family complexity measures (Jung et al., 2022; Summers & Shah, 2010; Trattner et al., 2019), their validation (Briand et al., 1996; Weyuker, 1988), the power of numerical experiments to identify similarities (Hennig et al., 2022) and the conceptual EAD (Mertens, 2020; Matthias Meyer et al., 2019).

Following the call of recent research (Hackl et al., 2020; Lyons et al., 2020; Trattner et al., 2019), the economic consequences of internal complexity are investigated using the operationalized EAD model. In doing so, effects, as noted by Hackl et al.'s (2020) conceptual impact model, are operationalized by integrated analytical models from the literature (e.g., Thonemann & Brandeau, 2000; L. L. Zhang et al., 2020) into the EAD simulation model. While the impact model focuses on product design, this work adds the variables of product variety and demand to the model. EAD's ability to support decision-making is highlighted via a case study in which various product family design alternatives are systematically generated and their costs evaluated. The linkage between internal complexity and total costs is generalized by conducting large-scale numerical experiments. Results reveal non-straightforward cause-effect relationships. For example, increased component commonality decreases setup costs only under certain conditions. In sum, the model shows the two faces of component commonality, as noted by Labro (2004). Specifically, total costs induced by internal complexity are a U-shaped curve of component commonality. While too little commonality results in unused opportunities for cost savings, too much commonality increases costs since increased economies of scale cannot compensate for the additional costs (e.g., material, setup). Thus, the curve's apex indicates the optimal design. However, local cost minima exist due to the non-straightforward behavior of the individual cost effects. Based on the results, several implications for research and practice are made. From a researcher's perspective, the experiment demonstrates the systematic design space exploration by generating design alternatives via an overdesign heuristic and evaluating their costs. It, therefore, extends numerical studies, focusing on product design optimization (e.g., Fujita, 2002; Kashkoush & ElMaraghy, 2017; Sinha & Suh, 2018). The experiment further highlights that questions of product design optimization cannot be answered uncoupled from a product variety and demand perspective due to the interaction of individual cost effects. From a practical perspective, the experiment is a further step towards an increased transparency of complexity-induced costs. It is argued that the negative cost effects of increased component commonality are more prominent in firms as they affect direct costs, while cost savings affect indirect costs. Decision makers can use the numerical EAD model with the integrated cost-effects to identify cost-optimal product family designs. Specifically, they can generate a large variety of design alternatives, identify conditions under which specific designs are preferable over others, and by including uncertainty, identify the most likely winning design. This is important since several authors note the challenges in decision-making during the product development process (Chen et al., 2022; Fixson, 2006; Geng et al., 2010; Skirde et al., 2016). Besides these implications, it is essential to note that the numerical EAD framework supports the decision-making process among many other methods and tools. In line with Ripperda and Krause (2017), this work agrees that product development decisions are highly individual, driven by many (observable and latent) factors, and, therefore, must be made on a case-by-case basis.

Related to the third research objective, this work investigates the effects of internal complexity on product costing system accuracy. Product costing information is used for a variety of decisions within firms, such as resource acquisition or supply and product mix decisions (V. Anand et al., 2017; Balakrishnan & Sivaramakrishnan, 2002; M. Gupta & King, 1997; Homburg et al., 2018; Innes et al., 2000; Labro, 2018). Therefore, recent research explored the mechanisms leading to product costing errors. However, studies generate research consumption patterns based on randomly correlated numbers (e.g., Balakrishnan et al., 2011; Schmidt et al., 2023) rather than being the result of an underlying product family design. This work uses the numerical EAD framework to generate resource consumption patterns. Following the idea of cumulative science, the product costing system heuristics are adapted from the ABL framework. The further extended EAD framework allows for linking product family design decisions with the accuracy of product costing systems. Doing so is a further step towards a closer exchange between product management, engineering, and cost accounting following the call of different authors (Labro, 2015a; Mertens, 2020). Patterns generated by the ABL and EAD framework are compared in the first step. The results indicate that the EAD generates more homogenous patterns in general, reflecting that products of a product family share more similarities than dissimilarities (Buchholz, 2012). However, EAD-generated matrices show an increased heterogeneity in two aspects. First, the range of product resource consumption is larger under the EAD and, therefore, the range of product's true benchmark costs. Second, the EAD generates patterns with a high skewness in total resource consumption, indicating that some resources are used in a high quantity while others are only rarely used. These patterns reflect mass customizing firms where all products share certain resources, such as a central assembly line, while additional, less shared resources indicate the individualization of products. These findings underline the idea of the base consumption in the ABL framework and highlight its importance. EAD-generated resource consumption patterns are used in the second analysis to identify product family design characteristics, driving the total product costing error. In doing so, a simple volume-based system and two ABC heuristics are compared. The results indicate that volume-based PCS cannot outperform ABC systems under the generated patterns and are highly sensitive to product mix and cost structure changes. However, they are more robust to product family design changes than ABC systems. The last experiment investigates the characteristics of product cross-subsidization. While the literature (e.g., Horngren et al., 2012; Labro & Vanhoucke, 2007) states a clear picture regarding the over- and under-costing pattern, a more diffuse picture is identified in this work. Specifically, the dominance of under-costing is present under volume-based systems but absent under ABC systems, which confirms a recent study by Schmidt et al. (2023). By investigating the characteristics of products with biased costs, this work finds that exotic and low-volume products tend to be under-costed and high-volume products with a high production complexity over-costed under volume-based systems. The newly introduced local outlier factor seems to be a promising proxy to characterize under-costed products. Product-level characteristics are diffuse under ABC systems. This might be the result of misspecified measures or the complex behavior of these systems. At least, this work identifies that products with a high production complexity show higher absolute costing errors under ABC systems. However, the product-level measures do not allow for estimating whether these products are over- or under-costed under ABC systems.

This work is a further step toward a better understanding of design decision's consequences. Understanding these effects is important since, in practice, causes are hidden due to the complex mechanisms, and effects occur delayed. By operationalizing the conceptual impact model (Hackl et al., 2020; Hackl, 2022), practitioners have a framework to generate design solutions and evaluate their economic consequences. This is the foundation for optimizations where product design is systematically varied and economic consequences, such as costs and cost system accuracy, are objectives. The study by Sinha and Suh (2018) shows the potential of optimizing product family designs. However, their study does not consider costs and is limited to the intra-domain design. The present work presents a simulation model for optimizations beyond the physical design. This is relevant, especially in early-stage development phases where the lever on costs and the uncertainty are high (Fixson, 2006; Skirde et al., 2016). By conducting robustness simulations, firms can identify the most likely cost-efficient product family design. Remember, providing product variety cost-efficiently is a competitive advantage. In sum, the model reflects the increasing interest in research and practice, using digital twins of their product families for evaluating decisions a priori (Tao, Xiao, Qi, Cheng, & Ji, 2022).

The current model is just a first step towards these goals, and several open issues are identified throughout this thesis. For example, demand is modeled as an exogenous variable where including the discrete choice theory would allow for estimating the optimal product family design from a cost and revenue perspective. Another open topic is the current complexity of the model, requiring advanced users. Model complexity must be reduced to increase its practical relevance since several simplifications have already been made.

References

Abdelkafi, N. (2008). *Variety induced complexity in mass customization: Concepts and management*. *Operations and technology management: Vol. 7*. Berlin: Erich Schmidt.

Adler, P. S., & Clark, K. B. (1991). Behind the Learning Curve: A Sketch of the Learning Process. *Management Science*, *37*(3), 267–281. https://doi.org/10.1287/mnsc.37.3.267

Ai, X. (2017). Node Importance Ranking of Complex Networks with Entropy Variation. *Entropy*, *19*(7), 303. https://doi.org/10.3390/e19070303

Alfadel, M., Kobilica, A., & Hassine, J. (2017). Evaluation of Halstead and Cyclomatic Complexity Metrics in Measuring Defect Density. In *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)* (pp. 1–9). IEEE. https://doi.org/10.1109/IEEEGCC.2017.8447959

AlGeddawy, T., & ElMaraghy, H. (2013). Optimum granularity level of modular product design architecture. *CIRP Annals*, *62*(1), 151–154. https://doi.org/10.1016/j.cirp.2013.03.118

Al-Omiri, M., & Drury, C. (2007). A survey of factors influencing the choice of product costing systems in UK organizations. *Management Accounting Research*, *18*(4), 399–424. https://doi.org/10.1016/j.mar.2007.02.002

Al-Sayed, M., & Dugdale, D. (2016). Activity-based innovations in the UK manufacturing sector: Extent, adoption process patterns and contingency factors. *The British Accounting Review*, *48*(1), 38–58. https://doi.org/10.1016/j.bar.2015.03.004

Alsayegh, M. F. (2020). Activity Based Costing around the World: Adoption, Implementation, Outcomes and Criticism. *Journal of Accounting and Finance in Emerging Economies*, *6*(1), 251–262. https://doi.org/10.26710/jafee.v6i1.1074

Ameri, F., Summers, J. d., Mocko, G. M., & Porter, M. (2008). Engineering design complexity: an investigation of methods and measures. *Research in Engineering Design*, *19*(2-3), 161–179. https://doi.org/10.1007/s00163-008-0053-2

Anand, K., Bianconi, G., & Severini, S. (2011). Shannon and von Neumann entropy of random networks with heterogeneous expected degree. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *83*(3 Pt 2), 36109. https://doi.org/10.1103/PhysRevE.83.036109

Anand, V., & Balakrishnan, R. (2013). On the Profit Performance of Tidy Cost Systems : A Numerical Experiment. In

Anand, V., Balakrishnan, R., & Labro, E. (2017). Obtaining Informationally Consistent Decisions When Computing Costs with Limited Information. *Production and Operations Management*, *26*(2), 211–230. https://doi.org/10.1111/poms.12631

Anand, V., Balakrishnan, R., & Labro, E. (2019). A Framework for Conducting Numerical Experiments on Cost System Design. *Journal of Management Accounting Research*, *31*(1), 41–61. https://doi.org/10.2308/jmar-52057

Anderson, D. M. (2008). *Build-to-order & mass customization: The ultimate supply chain management and lean manufacturing strategy for low-cost on-demand production without forecasts or inventory*. Cambria, Calif.: CIM Press.

Anderson, S. P., Palma, A. de, & Thisse, J.-F. (1989). Demand for Differentiated Products, Discrete Choice Models, and the Characteristics Approach. *The Review of Economic Studies*, *56*(1), 21–35. https://doi.org/10.2307/2297747

Anderson, S. P., Palma, A. de, & Thisse, J.-F. (1992). *Discrete Choice Theory of Product Differentiation*. The MIT Press. https://doi.org/10.7551/mitpress/2450.001.0001

Anderson, S. W. (1995). Measuring the Impact of Product Mix Heterogeneity on Manufacturing Overhead Cost. *The Accounting Review*, *70*(3), 363–387. Retrieved from http://www.jstor.org/stable/248530

Anderson, S. W., & Dekker, H. C. (2009). Strategic Cost Management in Supply Chains, Part 1: Structural Cost Management. *Accounting Horizons*, *23*(2), 201–220. https://doi.org/10.2308/acch.2009.23.2.201

Anderson, S. W., & Sedatole, K. L. (2013). Evidence on the Cost Hierarchy: The Association between Resource Consumption and Production Activities. *Journal of Management Accounting Research*, *25*(1), 119–141. https://doi.org/10.2308/jmar-50293

Andrews, C. R., Cannon, H. M., Cannon, J. N., & Low, J. T. (2009). Beyond the Profitable Product Death Spiral: Managing Product Mix in an Environment of Constrained Resources. *Developments in Business Simulation and Experiential Learning*, *36*.

Antony, J. (2014). *Design of Experiments for Engineers and Scientists*. Elsevier. https://doi.org/10.1016/C2012-0-03558-2

Arora, J. S. (2017). Multi-objective Optimum Design Concepts and Methods. In *Introduction to Optimum Design* (pp. 771–794). Elsevier. https://doi.org/10.1016/B978-0-12-800806-5.00018-4

Asiedu, Y., & Gu, P. (1998). Product life cycle cost analysis: State of the art review. *International Journal of Production Research*, *36*(4), 883–908. https://doi.org/10.1080/002075498193444

Axelrod, R. (1997). Advancing the Art of Simulation in the Social Sciences. In P. Terna, R. Conte, & R. Hegselmann (Eds.), *Lecture Notes in Economics and Mathematical Systems: Vol. 456. Simulating social phenomena* (Vol. 456, pp. 21–40). Berlin: Springer. https://doi.org/10.1007/978-3-662-03366-1_2

Babad, Y. M., & Balachandran, B. V. (1993). Cost Driver Optimization in Activity-Based Costing. *The Accounting Review*, *68*(3), 563–575. Retrieved from http://www.jstor.org/stable/248201

Baker, K. R. (1985). Safety stocks and component commonality. *Journal of Operations Management*, *6*(1), 13–22. https://doi.org/10.1016/0272-6963(85)90031-2

Baker, K. R., Magazine, M. J., & Nuttle, H. L. W. (1986). The Effect of Commonality on Safety Stock in a Simple Inventory Model. *Management Science*, *32*(8), 982–988. https://doi.org/10.1287/mnsc.32.8.982

Balakrishnan, R., Hansen, S., & Labro, E. (2011). Evaluating Heuristics Used When Designing Product Costing Systems. *Management Science*, *57*(3), 520–541. https://doi.org/10.1287/mnsc.1100.1293

Balakrishnan, R., Labro, E., & Sivaramakrishnan, K. (2012). Product Costs as Decision Aids: An Analysis of Alternative Approaches (Part 1). *Accounting Horizons*, *26*(1), 1–20. https://doi.org/10.2308/acch-50086

Balakrishnan, R., & Penno, M. (2014). Causality in the context of analytical models and numerical experiments. *Accounting, Organizations and Society*, *39*(7), 531–534. https://doi.org/10.1016/j.aos.2013.09.004

Balakrishnan, R., Pugely, A. J., & Shah, A. S. (2017). Modeling Resource Use with Time Equations: Empirical Evidence. *Journal of Management Accounting Research*, *29*(1), 1–12. https://doi.org/10.2308/jmar-51444

Balakrishnan, R., & Sivaramakrishnan, K. (2002). A Critical Overview of the Use of Full-Cost Data for Planning and Pricing. *Journal of Management Accounting Research*, *14*(1), 3–31. https://doi.org/10.2308/jmar.2002.14.1.3

Baldwin, C., & Clark, K. B. (2000). *Design rules: Volume 1: The power of modularity*. Cambridge, Mass: MIT Press. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=27240

Banker, R. D., Chang, H., & Cunningham, R. (2003). The public accounting industry production function. *Journal of Accounting and Economics*, *35*(2), 255–281. https://doi.org/10.1016/S0165-4101(03)00021-1

Banker, R. D., & Hughes, J. S. (1994). Product Costing and Pricing. *The Accounting Review*, *69*(3), 479–494. Retrieved from http://www.jstor.org/stable/248235

Barney, J. B. (1986). Strategic Factor Markets: Expectations, Luck, and Business Strategy. *Management Science*, *32*(10), 1231–1241. https://doi.org/10.1287/mnsc.32.10.1231

Barney, J. B. (1991). Firm Resources and Sustained Competitive Advantage. *Journal of Management*, *17*(1), 99–120. https://doi.org/10.1177/014920639101700108

Bashir, H. A., & Thomson, V. (2001). Models for estimating design effort and time. *Design Studies*, *22*(2), 141–155. https://doi.org/10.1016/S0142-694X(00)00014-4

Baxendale, S. J. (2001). Activity-based costing for the small business: a primer. *Business Horizons*, *44*(1), 61–68. https://doi.org/10.1016/S0007-6813(01)80010-0

Bayus, B. L., Erickson, G., & Jacobson, R. (2003). The Financial Rewards of New Product Introductions in the Personal Computer Industry. *Management Science*, *49*(2), 197–210. https://doi.org/10.1287/mnsc.49.2.197.12741

Becker, M., & Knudsen, T. (2012). Nelson and Winter revisited. In M. Dietrich & J. Krafft (Eds.), *Handbook on the Economics and Theory of the Firm* (pp. 243–255). Edward Elgar Publishing.

Ben-Akiva, M. E., & Lerman, S. R. (1985). *Discrete choice analysis: Theory and application to travel demand*. *MIT Press Series in Transportation Studies: Vol. 9*. Cambridge, Mass, London: The MIT Press.

Ben-Arieh, D., & Qian, L. (2003). Activity-based cost management for design and development stage. *International Journal of Production Economics*, *83*(2), 169–183. https://doi.org/10.1016/S0925-5273(02)00323-7

Benavides, D., Segura, S., & Ruiz-Cortés, A. (2010). Automated analysis of feature models 20 years later: A literature review. *Information Systems*, *35*(6), 615–636. https://doi.org/10.1016/j.is.2010.01.001

Benavides, E. M. (2012). *Advanced engineering design : An integrated approach*. *Woodhead Publishing in mechanical engineering*. Cambridge, UK: Philadelphia, PA.

Benton, W. C., & Krajewski, L. (1990). Vendor Performance and Alternative Manufacturing Environments. *Decision Sciences*, *21*(2), 403–415. https://doi.org/10.1111/j.1540-5915.1990.tb01693.x

Bertalanffy, L. v. (1979). *General system theory : foundations, development, applications* (Rev. ed., 6. print). New York: Braziller.

Bjørnenak, T. (1997). Diffusion and accounting: the case of ABC in Norway. *Management Accounting Research*, *8*(1), 3–17. https://doi.org/10.1006/mare.1996.0031

Bjørnenak, T., & Helgesen, Ø. (2013). Customer relations and cost management. *The Routledge Companion to Cost Management*, 250.

Blecker, T., & Abdelkafi, N. (2006). Complexity and variety in mass customization systems: analysis and recommendations. *Management Decision*, *44*(7), 908–929. https://doi.org/10.1108/00251740610680596

Blecker, T., & Abdelkafi, N. (2007). The Development of a Component Commonality Metric for Mass Customization. *IEEE Transactions on Engineering Management*, *54*(1), 70–85. https://doi.org/10.1109/TEM.2006.889068

Blecker, T., & Kersten, W. (Eds.) (2006). *Operations and technology management*. *Complexity Management in Supply Chains, Concepts, Tools and Methods*. Berlin: Erich Schmidt Verlag.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008. https://doi.org/10.1088/1742-5468/2008/10/P10008

Boer, L. de, Labro, E., & Morlacchi, P. (2001). A review of methods supporting supplier selection. *European Journal of Purchasing & Supply Management*, *7*(2), 75–89. https://doi.org/10.1016/S0969-7012(00)00028-9

Bongulielmi, L., Henseler, P., Puls, C., & Meier, M. H. (2001). The K- A V-Matrix Method - An Approach in Analysis and Description of Variant Products. In *Proceedings of International Coneference on Engineering Design.* Symposium conducted at the meeting of ICED, Glasgow.

Bonjour, É., & Micaëlli, J.-P. (2010). Design Core Competence Diagnosis: A Case From the Automotive Industry. *IEEE Transactions on Engineering Management*, *57*(2), 323–337. https://doi.org/10.1109/TEM.2009.2036838

Borjesson, F., & Hölttä-Otto, K. (2014). A module generation algorithm for product architecture based on component interactions and strategic drivers. *Research in Engineering Design*, *25*(1), 31–51. https://doi.org/10.1007/s00163-013-0164-2

Brahm, F., Tarzijan, J., & Singer, M. (2017). The Impact of Frictions in Routine Execution on Economies of Scope. *Strategic Management Journal*, *38*(10), 2121–2142. https://doi.org/10.1002/smj.2643

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. *ACM SIGMOD Record*, *29*(2), 93–104. https://doi.org/10.1145/335191.335388

Briand, L. C., Morasca, S., & Basili, V. R. (1996). Property-based software engineering measurement. *IEEE Transactions on Software Engineering*, *22*(1), 68–86. https://doi.org/10.1109/32.481535

Brierley, J. A., Cowton, C. J., & Drury, C. (2001). Research into product costing practice: a European perspective. *European Accounting Review*, *10*(2), 215–256. https://doi.org/10.1080/09638180126635

Brosch, M. (2014). *Eine Methode zur Reduzierung der produktvarianteninduzierten Komplexität* (1. Aufl.). *Hamburger Schriftenreihe Produktentwicklung und Konstruktionstechnik: Bd. 7*. Hamburg: TuTech-Verl.

Browning, T. R. (2001). Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transactions on Engineering Management*, *48*(3), 292–306. https://doi.org/10.1109/17.946528

Brun, A., & Pero, M. (2012). Measuring variety reduction along the supply chain: The variety gap model. *International Journal of Production Economics*, *139*(2), 510–524. https://doi.org/10.1016/j.ijpe.2012.05.018

Buchholz, M. (2012). *Theorie der Variantenvielfalt: Ein produktions- und absatzwirtschaftliches Erklärungsmodell*. Wiesbaden: Gabler Verlag.

Bundesnetzagentur (2022). Monitoringbericht 20211. Retrieved from https://www.bundesnetzagentur.de/SharedDocs/Mediathek/Monitoringberichte/Monitoringbericht_Energie2021.pdf

Cagwin, D., & Bouwman, M. J. (2002). The association between activity-based costing and improvement in financial performance. *Management Accounting Research*, *13*(1), 1–39. https://doi.org/10.1006/mare.2001.0175

Calvo, E., & Martínez-de-Albéniz, V. (2016). Sourcing Strategies and Supplier Incentives for Short-Life-Cycle Goods. *Management Science*, *62*(2), 436–455. https://doi.org/10.1287/mnsc.2014.2138

Campagnolo, D., & Camuffo, A. (2009). The concept of modularity in management studies: A literature review. *International Journal of Management Reviews.* Advance online publication. https://doi.org/10.1111/j.1468-2370.2009.00260.x

Cannon, J. N., Cannon, H. M., & Schwaiger, M. (2012). Modeling the "Profitable-Product Death Spiral". *Simulation & Gaming*, *43*(6), 761–777. https://doi.org/10.1177/1046878111434474

Cardinaels, E., & Labro, E. (2008). On the Determinants of Measurement Error in Time-Driven Costing. *The Accounting Review*, *83*(3), 735–756. https://doi.org/10.2308/accr.2008.83.3.735

Carley, K. M. (2002). Computational organizational science and organizational engineering. *Simulation Modelling Practice and Theory*, *10*(5-7), 253–269. https://doi.org/10.1016/S1569-190X(02)00119-3

Caves, R. E. (1980). Industrial Organization, Corporate Strategy and Structure. *Journal of Economic Literature*, *18*(1), 64–92. Retrieved from http://www.jstor.org/stable/2723892

Chang, A. S.-T., Shih, J. S., & Choo, Y. S. (2011). Reasons and costs for design change during production. *Journal of Mechanical Design*, *22*(4), 275–289. https://doi.org/10.1080/09544820903425218

Chaudhuri, A., & Boer, H. (2016). The impact of product-process complexity and new product development order winners on new product development performance: The mediating role of collaborative competence. *Journal of Engineering and Technology Management*, *42*, 65–80. https://doi.org/10.1016/j.jengtecman.2016.10.002

Chen, Z., Pan, Z., Ma, Q., Hou, T., & Zhao, P. (2022). An MAGDM method for design concept evaluation based on incomplete information. *PloS One*, *17*(11), e0277964. https://doi.org/10.1371/journal.pone.0277964

Child, P., Diederichs, R., Sanders, F.-H., Wisniowski, S., & Cummings, P. (1991). Smr Forum: The management of complexity. *Sloan Management Review*, *33*, 73+.

Child, P., Diederichs, R., & Sanders, Falk-Hayo and Wisniowski, Stefan (1992). The management of complexity. *Journal of Product Innovation Management*, *9*(3), 254–255. https://doi.org/10.1016/0737-6782(92)90043-C

Chinnaiah, P. S. S., & Kamarthi, S. V. (2000). Mass Customization and Manufacturing. In P. M. Swamidass (Ed.), *Innovations in Competitive Manufacturing* (Vol. 28, pp. 283–296). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4615-1705-4_24

Chiu, M.-C., & Okudan, G. (2012). An investigation on the impact of product modularity level on supply chain performance metrics: an industrial case study. *Journal of Intelligent Manufacturing*, *25*(1), 129. https://doi.org/10.1007/s10845-012-0680-3

Christensen, J., & Hemmer, T. (2006). Analytical Modeling of Cost in Management Accounting Research. In *Handbooks of Management Accounting Research* (Vol. 2, pp. 557–571). Elsevier. https://doi.org/10.1016/S1751-3243(06)02004-9

Cinquini, L., Collini, P., Marelli, A., & Tenucci, A. (2015). Change in the relevance of cost information and costing systems: evidence from two Italian surveys. *Journal of Management & Governance*, *19*(3), 557–587. https://doi.org/10.1007/s10997-013-9275-4

Claycomb, C., & Frankwick, G. L. (2004). A Contingency Perspective of Communication, Conflict Resolution and Buyer Search Effort in Buyer-Supplier Relationships. *Journal of Supply Chain Management*, *40*(1), 18–34. https://doi.org/10.1111/j.1745-493X.2004.tb00253.x

Cokins, G. (2001). *Activity-based cost management: An executive's guide*. New York: Wiley.

Collier, D. A. (1981). THE MEASUREMENT AND OPERATING BENEFITS OF COMPONENT PART COMMONALITY. *Decision Sciences*, *12*(1), 85–96. https://doi.org/10.1111/j.1540-5915.1981.tb00063.x

Collier, D. A. (1982). Aggregate Safety Stock Levels and Component Part Commonality. *Management Science*, *28*(11), 1296–1303. https://doi.org/10.1287/mnsc.28.11.1296

Collinson, S., & Jay, M. (2012). *From complexity to simplicity: Unleash your organisation's potential* (Online-Ausg). Basingstoke: Palgrave Macmillan.

Cooper, R., & Kaplan, R. S. (1988a). How Cost Accounting Distorts Product Costs. *Management Accounting*, *69*(10), 20–27.

Cooper, R., & Kaplan, R. S. (1988b). Measure Costs Right: Make the Right Decisions. *Harvard Business Review*, *66*(5), 96.

Cooper, R., & Kaplan, R. S. (1991). Profit Priorities from Activity-Based-Costing. *Harvard Business Review*, *69*(3), 130.

Cormier, P., van Horn, D., & Lewis, K. (2009). Investigating the Use of (Re)Configurability to Reduce Product Family Cost and Mitigate Performance Losses. In *Volume 8: 14th Design for Manufacturing and the Life Cycle Conference; 6th Symposium on International Design and Design Education; 21st International Conference on Design Theory and Methodology, Parts A and B* (pp. 1089–1100). ASMEDC. https://doi.org/10.1115/DETC2009-87439

Cover, T. M., & Thomas, J. A. (2005). *Elements of Information Theory*. Wiley. https://doi.org/10.1002/047174882X

Czarnecki, K., Helsen, S., & Eisenecker, U. (2004). Staged Configuration Using Feature Models. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, . . . R. L. Nord (Eds.), *Lecture Notes in Computer Science. Software Product Lines* (Vol. 3154, pp. 266–283). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28630-1_17

Daaboul, J., Da Cunha, C., Bernard, A. [Alain], & Laroche, F. (2011). Design for mass customization: Product variety vs. process variety. *CIRP Annals*, *60*(1), 169–174. https://doi.org/10.1016/j.cirp.2011.03.093

Danese, P., & Romano, P. (2004). Improving inter-functional coordination to face high product variety and frequent modifications. *International Journal of Operations & Production Management*, *24*(9), 863–885. https://doi.org/10.1108/01443570410552090

Danilovic, M., & Sandkull, B. (2005). The use of dependence structure matrix and domain mapping matrix in managing uncertainty in multiple project situations. *International Journal of Project Management*, *23*(3), 193–203. https://doi.org/10.1016/j.ijproman.2004.11.001

Datar, S., & Gupta, M. (1994). Aggregation, Specification and Measuremenr Errors in Product Costing. *The Accouting Review*, *69*(4), 567–591.

Debruyne, M., Moenaertb, R., Griffinc, A., Hartd, S., Hultinke, E. J., & Robben, H. (2002). The impact of new product launch strategies on competitive reaction in industrial markets. *Journal of Product Innovation Management*, *19*(2), 159–170. https://doi.org/10.1111/1540-5885.1920159

Delaš, J., Škec, S., & Štorga, M. (2018). Application of Axiomatic Design principles in conceptual design. *MATEC Web of Conferences*, *223*, 1008. https://doi.org/10.1051/matecconf/201822301008

Dellaert, B. G., & Häubl, G. (2012). Searching in Choice Mode: Consumer Decision Processes in Product Search with Recommendations. *Journal of Marketing Research*, *49*(2), 277–288. https://doi.org/10.1509/jmr.09.0481

Demski, J. S. (2008). *Managerial uses of accounting information*. *Springer series in accounting scholarship: Vol. 4*. Boston: Springer. https://doi.org/10.1007/978-0-387-77451-0

Dierickx, I., & Cool, K. (1989). Asset Stock Accumulation and Sustainability of Competitive Advantage. *Management Science*, *35*(12), 1504–1511. https://doi.org/10.1287/mnsc.35.12.1504

Dietrich, M., & Krafft, J. (Eds.) (2012). *Handbook on the Economics and Theory of the Firm*. Edward Elgar Publishing. https://doi.org/10.4337/9781781002407

Diewert, W. E. (1971). An Application of the Shephard Duality Theorem: A Generalized Leontief Production Function. *Journal of Political Economy*, *79*(3), 481–507. https://doi.org/10.1086/259764

Dogramaci, A. (1979). Design of Common Components Considering Implications of Inventory Costs and Forecasting. *A I I E Transactions*, *11*(2), 129–135. https://doi.org/10.1080/05695557908974452

Drury, C., & Tayles, M. (2005). Explicating the design of overhead absorption procedures in UK organizations. *The British Accounting Review*, *37*(1), 47–84. https://doi.org/10.1016/j.bar.2004.08.003

Efatmaneshnik, M., & Ryan, M. J. (2016). A general framework for measuring system complexity. *Complexity*, *21*(S1), 533–546. https://doi.org/10.1002/cplx.21767

Ehrlenspiel, K., Kiewert, A., Lindemann, U., & Hundal, M. S. (2007). *Cost-Efficient Design*. Three Park Avenue New York, NY 10016-5990: ASME. https://doi.org/10.1115/1.802507

El Maraghy, W. H., & Urbanic, R. J. (2003). Modelling of Manufacturing Systems Complexity. *CIRP Annals*, *52*(1), 363–366. https://doi.org/10.1016/S0007-8506(07)60602-7

ElMaraghy, H., Schuh, G. [G.], ElMaraghy, W., Piller, F., Schönsleben, P., Tseng, M., & Bernard, A. [A.] (2013). Product variety management. *CIRP Annals*, *62*(2), 629–652. https://doi.org/10.1016/j.cirp.2013.05.007

Ericsson, A., & Erixon, G. (1999). *Controlling design variants: Modular product platforms*. Dearborn, MI, Fairfield: Society of Manufacturing Engineers; American Society of Mechanical Engineers.

Erumban, A. A. (2008). LIFETIMES OF MACHINERY AND EQUIPMENT: EVIDENCE FROM DUTCH MANUFACTURING. *Review of Income and Wealth*, *54*(2), 237–268. https://doi.org/10.1111/j.1475-4991.2008.00272.x

Escobar-Saldívar, L. J., Smith, N. R., & González-Velarde, J. L. (2008). An approach to product variety management in the painted sheet metal industry. *Computers & Industrial Engineering*, *54*(3), 474–483. https://doi.org/10.1016/j.cie.2007.08.009

European Commission (2021, September 23). *Verbraucherfrustration und Elektroabfällen den Stecker ziehen: Kommission schlägt einheitliches Ladegerät für elektronische Geräte vor* [Press release]. Brussles. Retrieved from https://ec.europa.eu/commission/presscorner/detail/de/ip_21_4613

Eynan, A., & Rosenblatt, M. (1996). Component commonality effects on inventory costs. *IIE Transactions*, *28*(2), 93–104. https://doi.org/10.1080/07408179608966255

Feldhusen, J., Nurcahya, E., & Löwer, M. (2007). Variant Creation Using Configuration of a Reference Variant. In *Ds 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007* (pp. 495–496). Glasgow: The Design Society.

Feltham, G. A. (1977). Cost Aggregation: An Information Economic Analysis. *Journal of Accounting Research*, *15*(1), 42. https://doi.org/10.2307/2490555

Fisher, M., & Ittner, C. d. (1999). The Impact of Product Variety on Automobile Assembly Operations: Empirical Evidence and Simulation Analysis. *Management Science*, *45*(6), 771–786. Retrieved from http://www.jstor.org/stable/2634770

Fisher, M., Ramdas, K., & Ulrich, K. (1999). Component Sharing in the Management of Product Variety: A Study of Automotive Braking Systems. *Management Science*, *45*(3), 297–315. https://doi.org/10.1287/mnsc.45.3.297

Fixson, S. K. (2005). Product architecture assessment: a tool to link product, process, and supply chain design decisions. *Journal of Operations Management*, *23*(3-4), 345–369. https://doi.org/10.1016/j.jom.2004.08.006

Fixson, S. K. (2006). A Roadmap For Product Architecture Costing. In T. W. Simpson, J. Jiao, & Z. Siddique (Eds.), *Product Platform and Product Family Design: Methods and Applications* (pp. 305–334). Boston, MA: Springer Science+Business Media LLC. https://doi.org/10.1007/0-387-29197-0_13

Fixson, S. K. (2007). Modularity and Commonality Research: Past Developments and Future Opportunities. *Concurrent Engineering*, *15*(2), 85–111. https://doi.org/10.1177/1063293X07078935

Foerster, H. von (1962). Communication amongst automata. *The American Journal of Psychiatry*, *118*, 865–871. https://doi.org/10.1176/ajp.118.10.865

Foerster, H. von (1978). The Curious Behavior of Complex Systems. In Linstone, H. A. And Clive Simmonds, W. H. (Ed.), *Future Research: New Directions*.

Foss, N. J., & Stieglitz, N. (2012). Modern resource-based theory(ies). In M. Dietrich & J. Krafft (Eds.), *Handbook on the Economics and Theory of the Firm* (pp. 256–274). Edward Elgar Publishing.

Foster, G., & Gupta, M. (1990). Manufacturing overhead cost driver analysis. *Journal of Accounting and Economics*, *12*(1-3), 309–337. https://doi.org/10.1016/0165-4101(90)90052-6

Foster, G., & Gupta, M. (1994). Marketing, Cost Management and Management Accounting. *Journal of Management Accounting Research*, 43–77.

Friedl, B. (2010). *Kostenrechnung*. München: OLDENBOURG WISSENSCHAFTSVERLAG. https://doi.org/10.1524/9783486710267

Fuchs, C. [Christoph], & Golenhofen, F. (2019). *Mastering Disruption and Innovation in Product Management*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-93512-6

Fujita, K. (2002). Product variety optimization under modular architecture. *Computer-Aided Design*, *34*(12), 953–965. https://doi.org/10.1016/S0010-4485(01)00149-X

Gavirneni, S. (2002). Information Flows in Capacitated Supply Chains with Fixed Ordering Costs. *Management Science*, *48*(5), 644–651. https://doi.org/10.1287/mnsc.48.5.644.7806

Gell-Mann, M. (1995). *Das Quark und der Jaguar: Vom Einfachen zum Komplexen, die Suche nach einer neuen Erklärung der Welt* (3. Aufl.). München: Piper.

Geng, X., Chu, X., & Zhang, Z. (2010). A new integrated design concept evaluation approach based on vague sets. *Expert Systems with Applications*, *37*(9), 6629–6638. https://doi.org/10.1016/j.eswa.2010.03.058

Gietzmann, M. (1991). Implementation issues associated with the construction of an activity-based costing system in an engineering components manufacturer. *Management Accounting Research*, *2*(3), 189–199. https://doi.org/10.1016/S1044-5005(91)70034-6

Globerson, S., & Levin, N. (1987). Incorporating Forgetting into Learning Curves. *International Journal of Operations & Production Management*, *7*(4), 80–94. https://doi.org/10.1108/eb054802

Gollop, F. M. (1997). The Pin Factory Revisited: Product Diversification and Productivity Growth. *Review of Industrial Organization*, *12*(3), 317–334. https://doi.org/10.1023/A:1007726012218

Gollop, F. M., & Monahan, J. L. (1991). A Generalized Index of Diversification: Trends in U.S. Manufacturing. *The Review of Economics and Statistics*, *73*(2), 318. https://doi.org/10.2307/2109523

Gonçalves-Coelho, A. M., & Mourão, A. J. (2007). Axiomatic design as support for decision-making in a design for manufacturing context: A case study. *International Journal of Production Economics*, *109*(1-2), 81–89. https://doi.org/10.1016/j.ijpe.2006.11.002

Gosselin, M. (1997). The effect of strategy and organizational structure on the adoption and implementation of activity-based costing. *Accounting, Organizations and Society*, *22*(2), 105–122. https://doi.org/10.1016/S0361-3682(96)00031-1

Gosselin, M. (2006). A Review of Activity-Based Costing: Technique, Implementation, and Consequences. In *Handbooks of Management Accounting Research* (Vol. 2, pp. 641–671). Elsevier. https://doi.org/10.1016/S1751-3243(06)02008-6

Götzfried, M. (2013). *Managing complexity induced by product variety in manufacturing companies : complexity evaluation and integration in decision-making.*

Gourville, J. T., & Soman, D. (2005). Overchoice and Assortment Type: When and Why Variety Backfires. *Marketing Science*, *24*(3), 382–395. https://doi.org/10.1287/mksc.1040.0109

Govindaraju, P., Achter, S., Ponsignon, T., Ehm, H., & Meyer, M. [Matthias] (2018). Comparison of two clustering approaches to find demand patterns in semiconductor supply chain planning. In *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)* (pp. 148–151). IEEE. https://doi.org/10.1109/COASE.2018.8560535

Greve, E. [E.], Fuchs, C. [C.], Hamraz, B., Windheim, M. [M.], Schwede, L.-N., & Krause, D. (2020). Investigating the Effects of Modular Product Structures to Support Design Decisions in Modularization Projects. In *2020 IEEE International Conference on Industrial*

Engineering and Engineering Management (IEEM) (pp. 295–299). IEEE. https://doi.org/10.1109/IEEM45057.2020.9309820

Grimm, R., Schuller, M., & Wilhelmer, R. (2014). *Portfoliomanagement in Unternehmen*. Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-00260-2

Guenov, M. D. (Ed.) (2002). *Complexity and cost effectiveness measures for systems design*. Retrieved from https://dspace.lib.cranfield.ac.uk/handle/1826/2620

Guenov, M. D., & Barker, S. G. (2005). Application of axiomatic design and design structure matrix to the decomposition of engineering systems. *Systems Engineering*, *8*(1), 29–40. https://doi.org/10.1002/sys.20015

Gupta, M. (1993). Heterogeneity issues in aggregated costing systems. *Journal of Management Accounting Research*, *5*, 180.

Gupta, M., & Galloway, K. (2003). Activity-based costing/management and its implications for operations management. *Technovation*, *23*(2), 131–138. https://doi.org/10.1016/S0166-4972(01)00093-1

Gupta, M., & King, R. R. (1997). An Experimental Investigation of the Effect of Cost Information and Feedback on Product Cost Decisions. *Contemporary Accounting Research*, *14*(1), 99–127. https://doi.org/10.1111/j.1911-3846.1997.tb00521.x

Gupta, S. [Saurabh], & Krishnan, V. (1999). Integrated Component and Supplier Selection for a Product Family. *Production and Operations Management*, *8*(2), 163–182. https://doi.org/10.1111/j.1937-5956.1999.tb00368.x

Hackl, J. (2022). *Wirkmodell der Eigenschaften modularer Produktstrukturen* (Vol. 22). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-65263-3

Hackl, J., Krause, D., Otto, K., Windheim, M. [Marc], Moon, S. K., Bursac, N., & Lachmayer, R. (2020). Impact of Modularity Decisions on a Firm's Economic Objectives. *Journal of Mechanical Design*, *142*(4). https://doi.org/10.1115/1.4044914

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2014). *Multivariate data analysis* (7th ed.). Upper Saddle River, NJ: Prentice Hall.

Halstead, M. H. (1979). *Elements of software science* (3. print). *The computer science library. Operating and programming systems series: Vol. 2*. New York, N.Y., Oxford: North-Holland.

Hamada, M., & Wu, C. F. J. (2009). *Experiments: Planning, analysis, and optimization* (2nd ed.). *Wiley series in probability and statistics*. Hoboken, New Jersey: John Wiley and Sons.

Han, S., Gupta, S. [Sunil], & Lehmann, D. R. (2001). Consumer price sensitivity and price thresholds. *Journal of Retailing*, *77*(4), 435–456. https://doi.org/10.1016/S0022-4359(01)00057-4

Hansen, D. R., & Mowen, M. M. (2006). *Cost management : Accounting and control* (5. ed., internat. student ed.). Mason, Ohio u.a.: Thomson/South-Western.

Hansen, S., & Magee, R. (1993). Capacity Cost and Capacity Allocation. *Contemporary Accounting Research*, *9*(2), 635–660. https://doi.org/10.1111/j.1911-3846.1993.tb00901.x

Hariprasad, T., Vidhyagaran, G., Seenu, K., & Thirumalai, C. (2017). Software complexity analysis using halstead metrics. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)* (pp. 1109–1113). IEEE. https://doi.org/10.1109/ICOEI.2017.8300883

Harland, P. E., & Uddin, Z. (2014). Effects of product platform development: fostering lean product development and production. *International Journal of Product Development*, *19*(5/6), 259. https://doi.org/10.1504/IJPD.2014.064881

Harrison, J. R., Lin, Z., Carroll, G. R., & Carley, K. M. (2007). Simulation modeling in organizational and management research. *Academy of Management Review*, *32*(4), 1229–1245. https://doi.org/10.5465/amr.2007.26586485

Hasegawa, T., Chen, S., & Duong, L. (2021). *Effects of Novel Coronavirus (COVID-19) on Civil Aviation: Economic Impact Analysis*. Retrieved from ICAO website: https://www.icao.int/sustainability/Documents/Covid-19/ICAO_coronavirus_Econ_Impact.pdf

Heese, H. S., & Swaminathan, J. M. (2006). Product Line Design with Component Commonality and Cost-Reduction Effort. *Manufacturing & Service Operations Management*, *8*(2), 206–219. https://doi.org/10.1287/msom.1060.0103

Hegge, H., & Wortmann, J. C. (1991). Generic bill-of-material: a new product model. *International Journal of Production Economics*, *23*(1-3), 117–128. https://doi.org/10.1016/0925-5273(91)90055-X

Heinen, E. (1985). *Betriebswirtschaftliche Kostenlehre: Kostentheorie u. Kostenentscheidungen* (6., verb. u. erw. Aufl., unveränd. Nachdr). Wiesbaden: Gabler.

Helson, H. (1964). *Adaptation-level Theory : An experimental and systematic approach to behavior*. New York u.a.: Harper & Row.

Henderson, R. M., & Clark, K. B. (1990). Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly*, *35*(1), 9. https://doi.org/10.2307/2393549

Hennig, A., Topcu, T. G., & Szajnfarber, Z. (2022). So You Think Your System Is Complex? Why and How Existing Complexity Measures Rarely Agree. *Journal of Mechanical Design*, *144*(4). https://doi.org/10.1115/1.4052701

Hiebl, M. R. W. (2021). Sample Selection in Systematic Literature Reviews of Management Research. *Organizational Research Methods*, 109442812098685. https://doi.org/10.1177/1094428120986851

Hillier, M. S. (2002). Using commonality as backup safety stock. *European Journal of Operational Research*, *136*(2), 353–365. https://doi.org/10.1016/S0377-2217(01)00027-3

Hilton, R. W. (2008). *Managerial Accounting*. McGraw-Hill Education.

Hodgson, G. M. (2012). Veblen, Commons and the Theory of the Firm. In M. Dietrich & J. Krafft (Eds.), *Handbook on the Economics and Theory of the Firm* (pp. 55–61). Edward Elgar Publishing.

Hölttä, K. M., & Otto, K. N. (2005). Incorporating design effort complexity measures in product architectural design and assessment. *Design Studies*, *26*(5), 463–485. https://doi.org/10.1016/j.destud.2004.10.001

Homburg, C. (2001). A note on optimal cost driver selection in ABC. *Management Accounting Research*, *12*(2), 197–205. https://doi.org/10.1006/mare.2000.0150

Homburg, C., Nasev, J., & Plank, P. (2018). The impact of cost allocation errors on price and product-mix decisions. *Review of Quantitative Finance and Accounting*, *51*(2), 497–527. https://doi.org/10.1007/s11156-017-0678-1

Horngren, C. T., Datar, S., & Rajan, M. (2012). *Cost accounting: A managerial emphasis* (14th ed.). Upper Saddle River, N.J.: Pearson/Prentice Hall.

Hounshell, D. A. (1989). *Science and corporate strategy : Du Pont R & D, 1902 - 1980* (Repr). *Studies in economic history and policy the United States in the twentieth century*. Cambridge u.a.: Cambridge Univ. Press.

Hu, L., & Bentler, P. M. (1999). Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, *6*(1), 1–55. https://doi.org/10.1080/10705519909540118

Hu, S. J., Zhu, X., Wang, H., & Koren, Y. (2008). Product Variety and Manufacturing Complexity in Assembly Systems and Supply Chains. *CIRP Annals*, *57*(1), 45–48. https://doi.org/10.1016/j.cirp.2008.03.138

Hughes, S., & Gjerde, K. (2003). Do Different Cost Systems Make a Difference? *Management Accounting*, *5*.

Hvam, L., Hansen, C. L., Forza, C. [Cipriano], Mortensen, N. H., & Haug, A. (2020). The reduction of product and process complexity based on the quantification of product complexity costs. *International Journal of Production Research*, *58*(2), 350–366. https://doi.org/10.1080/00207543.2019.1587188

Hwang, Y., Evans Ill, J. H., & Hedge, V. (1993). Production Cost Bias and Selection of an Allocation Base. *Journal of Management and Cost Accounting Research*, *5*(Herbst), 213–242.

Innes, J., & Mitchell, F. (1995). A survey of activity-based costing in the U.K.'s largest companies. *Management Accounting Research*, *6*(2), 137–153. https://doi.org/10.1006/mare.1995.1008

Innes, J., Mitchell, F., & Sinclair, D. (2000). Activity-based costing in the U.K.’s largest companies: a comparison of 1994 and 1999 survey results. *Management Accounting Research*, *11*(3), 349–362. https://doi.org/10.1006/mare.2000.0135

Ittner, C. d., Lanen, W. N., & Larcker, D. F. (2002). The Association Between Activity-Based Costing and Manufacturing Performance. *Journal of Accounting Research*, *40*(3), 711–726. https://doi.org/10.1111/1475-679X.00068

Ittner, C. d., Larcker, D. F., & Randall, T. (1997). The Activity-Based Cost Hierarchy, Production Policies and Firm profatibility. *Journal of Management Accounting Research*, 143.

Ittner, C. d., & Macduffie, J. P. (1995). Explaining Plant-level differences in Manufacturing Overhead: Strutrucal and Executional Cost Drivers in the World of Auto Industry. *Production and Operations Management*, *4*(4), 312–334. https://doi.org/10.1111/j.1937-5956.1995.tb00297.x

Iyengar, S. S., & Lepper, M. R. (2000). When choice is demotivating: Can one desire too much of a good thing? *Journal of Personality and Social Psychology*, *79*(6), 995–1006. https://doi.org/10.1037/0022-3514.79.6.995

Jacobs, M. (2013). Complexity: Toward an empirical measure. *Technovation*, *33*(4-5), 111–118. https://doi.org/10.1016/j.technovation.2013.01.001

Jacobs, M., Droge, C., Vickery, S. K., & Calantone, R. (2011). Product and Process Modularity's Effects on Manufacturing Agility and Firm Growth Performance. *Journal of Product Innovation Management*, *28*(1), 123–137. https://doi.org/10.1111/j.1540-5885.2010.00785.x

Jacobs, M., & Swink, M. (2011). Product portfolio architectural complexity and operational performance: Incorporating the roles of learning and fixed assets. *Journal of Operations Management*, *29*(7-8), 677–691. https://doi.org/10.1016/j.jom.2011.03.002

Jensen, J. B., Malhotra, M. K., & Philipoom, P. R. (1996). Machine dedication and process flexibility in a group technology environment. *Journal of Operations Management*, *14*(1), 19–39. https://doi.org/10.1016/0272-6963(95)00030-5

Jiao, J., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. *Journal of Intelligent Manufacturing*, *18*(1), 5–29. https://doi.org/10.1007/s10845-007-0003-2

Jiao, J., Tseng, M. M., Duffy, V. G., & Lin, F. (1998). Product family modeling for mass customization. *Computers & Industrial Engineering*, *35*(3-4), 495–498. https://doi.org/10.1016/S0360-8352(98)00142-9

Jiao, J., & Zhang, Y. (2006). Product Family Positioning. In T. W. Simpson, J. Jiao, & Z. Siddique (Eds.), *Product Platform and Product Family Design: Methods and Applications* (pp. 91–106). Boston, MA: Springer Science+Business Media LLC.

John Paul Macduffie (2013). Modularity-as-Property, Modularization-as-Process, and 'Modularity'-as-Frame: Lessons from Product Architecture Initiatives in the Global Automotive Industry. *Global Strategy Journal*, *3*, 8–40.

Johnson, M. D., & Kirchain, R. (2010). Developing and Assessing Commonality Metrics for Product Families: A Process-Based Cost-Modeling Approach. *IEEE Transactions on Engineering Management*, *57*(4), 634–648. https://doi.org/10.1109/TEM.2009.2034642

Johnson, N. F. (2007). *Simply complexity: A clear guide to complexity theory*. Oxford: Oneworld.

Jung, S., Sinha, K., & Suh, E. S. (2022). Domain Mapping Matrix-Based Metric for Measuring System Design Complexity. *IEEE Transactions on Engineering Management*, *69*(5), 2187–2195. https://doi.org/10.1109/TEM.2020.3004561

Kahraman, C., & Cebi, S. (2009). A new multi-attribute decision making method: Hierarchical fuzzy axiomatic design. *Expert Systems with Applications*, *36*(3), 4848–4861. https://doi.org/10.1016/j.eswa.2008.05.041

Kang, K. C. (Ed.) (1990). *Feature-oriented domain analysis (FODA) : feasibility study ; technical report CMU/SEI-90-TR-21 - ESD-90-TR-222*. Pittsburgh, Pa.: Software Engineering Inst., Carnegie Mellon Univ.

Kaplan, R. S. (1986). The role for empirical research in management accounting. *Accounting, Organizations and Society*, *11*(4-5), 429–452. https://doi.org/10.1016/0361-3682(86)90012-7

Kaplan, R. S., & Anderson, S. R. (2007). *Time-driven activity-based costing: A simpler and more powerful path to higher profits / Robert S. Kaplan, Steven R. Anderson*. Boston, Mass.: Harvard Business School; London : McGraw-Hill [distributor].

Kaplan, R. S., & Atkinson, A. A. (1998). *Advanced management accounting* (International ed., 3rd ed.). *Prentice-Hall International editions*. Upper Saddle River N.J. [etc.]: Prentice-Hall International.

Kaplan, R. S., & Cooper, R. (1992). Activity-based Systems: Measuring the Costs of Resource Usage. *Accounting Horizons*.

Karniel, A., & Reich, Y. (2009). From DSM-Based Planning to Design Process Simulation: A Review of Process Scheme Logic Verification Issues. *IEEE Transactions on Engineering Management*, *56*(4), 636–649. https://doi.org/10.1109/TEM.2009.2032032

Kashkoush, M., & ElMaraghy, H. (2017). Designing modular product architecture for optimal overall product modularity. *Journal of Mechanical Design*, *28*(5), 293–316. https://doi.org/10.1080/09544828.2017.1307949

Kekre, S. (1987). Performance of a Manufacturing Cell with Increased Product Mix. *IIE Transactions*, *19*(3), 329–339. https://doi.org/10.1080/07408178708975403

Kekre, S., & Srinivasan, K. (1990). Broader Product Line: A Necessity to Achieve Success? *Management Science*, *36*(10), 1216–1232. https://doi.org/10.1287/mnsc.36.10.1216

Kennedy, T., & Affleck-Graves, J. (2001). The Impact of Activity-Based Costing Techniques on Firm Performance. *Journal of Management Accounting Research*, *13*(1), 19–45. https://doi.org/10.2308/jmar.2001.13.1.19

Keuper, F. (2004). Systemkomplexität. *Die Betriebswirtschaft*. (05), 637. Retrieved from https://www.wiso-net.de/docPreview/primo/DBW__100404201?ZG_PORTAL=portal_exlibris

Khajavirad, A., & Michalek, J. J. (2007). An Extension of the Commonality Index for Product Family Optimization. In *Volume 6: 33rd Design Automation Conference, Parts A and B* (pp. 1001–1010). ASMEDC. https://doi.org/10.1115/DETC2007-35605

Kim, G., Kwon, Y., Suh, E. S., & Ahn, J. (2016). Analysis of Architectural Complexity for Product Family and Platform. *Journal of Mechanical Design*, *138*(7). https://doi.org/10.1115/1.4033504

Kim, Y.-S., & Cochran, D. S. (2000). Reviewing TRIZ from the perspective of Axiomatic Design. *Journal of Mechanical Design*, *11*(1), 79–94. https://doi.org/10.1080/095448200261199

Kleijnen, J. P. (1998). Experimental Design for Sensitivity Analysis, Optimization, and Validation of Simulation Models. In J. Banks (Ed.), *Handbook of Simulation* (pp. 173–223). Hoboken, NJ, USA: John Wiley & Sons, Inc.

Kogut, B., & Zander, U. (1992). Knowledge of the Firm, Combinative Capabilities, and the Replication of Technology. *Organization Science*, *3*(3), 383–397. https://doi.org/10.1287/orsc.3.3.383

Koolen, J. H., & Moulton, V. (2001). Maximal Energy Graphs. *Advances in Applied Mathematics*, *26*(1), 47–52. https://doi.org/10.1006/aama.2000.0705

Kor, Y. Y., & Mahoney, J. T. (2004). Edith Penrose's (1959) Contributions to the Resource-based View of Strategic Management. *Journal of Management Studies*, *41*(1), 183–191. https://doi.org/10.1111/j.1467-6486.2004.00427.x

Kota, S., Sethuraman, K., & Miller, R. (2000). A Metric for Evaluating Design Commonality in Product Families. *Journal of Mechanical Design*, *122*(4), 403. https://doi.org/10.1115/1.1320820

Krause, D., & Gebhardt, N. (Eds.) (2018). *Methodische Entwicklung modularer Produktfamilien*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53040-5

Krishnan, V., & Gupta, S. [Saurabh] (2001). Appropriateness and Impact of Platform-Based Product Development. *Management Science*, *47*(1), 52–68. https://doi.org/10.1287/mnsc.47.1.52.10665

Krumwiede, K. R. (1998). The Implementation Stages of Activity-Based Costing and the Impact of Contextual and Organizational Factors. *Journal of Management Accounting Research*, *10*, 239–277. Retrieved from https://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=2720284&lang=de&site=ehost-live

Kuksov, D., & Villas-Boas, J. M. (2010). When More Alternatives Lead to Less Choice. *Marketing Science*, *29*(3), 507–524. https://doi.org/10.1287/mksc.1090.0535

Kulak, O., Cebi, S., & Kahraman, C. (2010). Applications of axiomatic design principles: A literature review. *Expert Systems with Applications*, *37*(9), 6705–6717. https://doi.org/10.1016/j.eswa.2010.03.061

Kumar, D., Chen, W., & Simpson, T. W. (2009). A market-driven approach to product family design. *International Journal of Production Research*, *47*(1), 71–104. https://doi.org/10.1080/00207540701393171

Labro, E. (2004). The Cost Effects of Component Commonality: A Literature Review Through a Management-Accounting Lens. *Manufacturing & Service Operations Management*, *6*(4), 358–367. https://doi.org/10.1287/msom.1040.0047

Labro, E. (2015a). Hobby Horses Ridden. *Journal of Management Accounting Research*, *27*(1), 133–138. https://doi.org/10.2308/jmar-51060

Labro, E. (2015b). Using simulation methods in accounting research. *Journal of Management Control*, *26*(2-3), 99–104. https://doi.org/10.1007/s00187-015-0203-4

Labro, E. (2018). Costing Systems. *Foundations and Trends® in Accounting*, *13*(3-4), 267–404. https://doi.org/10.1561/1400000058

Labro, E., & Vanhoucke, M. (2007). A Simulation Analysis of Interactions among Errors in Costing Systems. *The Accounting Review*, *82*(4), 939–962. Retrieved from http://www.jstor.org/stable/30243483

Labro, E., & Vanhoucke, M. (2008). Diversity in Resource Consumption Patterns and Robustness of Costing Systems to Errors. *Management Science*, *54*(10), 1715–1730. https://doi.org/10.1287/mnsc.1080.0885

Ladyman, J., Lambert, J., & Wiesner, K. (2013). What is a complex system? *European Journal for Philosophy of Science*, *3*(1), 33–67. https://doi.org/10.1007/s13194-012-0056-8

Lancaster, K. (1990). The Economics of Product Variety: A Survey. *Marketing Science*, *9*(3), 189–206. https://doi.org/10.1287/mksc.9.3.189

Lapedus, M. (2017). Battling Fab Cycle Times: Why it's taking longer to manufacture chips at 10/7nm and what can be done about it. Retrieved from https://semiengineering.com/battling-fab-cycle-times/#:~:text=Generally%2C%20the%20most%20common%20metric,40%20to%2050%20mask%20layers.

Lattin, J. M. (1987). A Model of Balanced Choice Behavior. *Marketing Science*, *6*(1), 48–65. Retrieved from http://www.jstor.org/stable/183816

Law, A. M. (2015). *Simulation modeling and analysis* (5. ed.). *McGraw-Hill series in industrial engineering and management science*. New York: McGraw-Hill.

Leicht, E. A., & Newman, M. E. J. (2008). Community structure in directed networks. *Physical Review Letters*, *100*(11), 118703. https://doi.org/10.1103/PhysRevLett.100.118703

Lemon, K. N., Zeithaml, V. A., & Rust, R. T. (2001). *Driving customer equity: How customer lifetime value is reshaping corporate strategy*. [Place of publication not identified]: Free Press.

Li, Y., Ni, Y., Zhang, N., & Liu, Z. (2021). Modularization for the complex product considering the design change requirements. *Research in Engineering Design*, *32*(4), 507–522. https://doi.org/10.1007/s00163-021-00369-6

Libby, R. (1981). *Accounting and human information processing: Theory and applications*. *Contemporary topics in accounting series*. Englewood Cliffs, N.J.: Prentice-Hall.

Lindemann, U., Maurer, M., & Braun, T. (2009). *Structural Complexity Management*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-87889-6

Lippman, S. A., & Rumelt, R. P. (2003). A bargaining perspective on resource advantage. *Strategic Management Journal*, *24*(11), 1069–1086. https://doi.org/10.1002/smj.345

Liu, D., Wang, H., & van Mieghem, P. (2010). Spectral perturbation and reconstructability of complex networks. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *81*(1 Pt 2), 16101. https://doi.org/10.1103/PhysRevE.81.016101

Loasby, B. J. (1998). The organisation of capabilities. *Journal of Economic Behavior & Organization*, *35*(2), 139–160. https://doi.org/10.1016/S0167-2681(98)00056-0

Lorscheid, I., Heine, B.-O., & Meyer, M. [Matthias] (2012). Opening the 'black box' of simulations: increased transparency and effective communication through the systematic design of experiments. *Computational and Mathematical Organization Theory*, *18*(1), 22–62. https://doi.org/10.1007/s10588-011-9097-3

Luft, J., & Shields, M. D. (2014). Subjectivity in developing and validating causal explanations in positivist accounting research. *Accounting, Organizations and Society*, *39*(7), 550–558. https://doi.org/10.1016/j.aos.2013.09.001

Lyons, A. C., Um, J., & Sharifi, H. (2020). Product variety, customisation and business process performance: A mixed-methods approach to understanding their relationships.

International Journal of Production Economics, *221*, 107469. https://doi.org/10.1016/j.ijpe.2019.08.004

Ma, S., Wang, W., & Liu, L. (2002). Commonality and postponement in multistage assembly systems. *European Journal of Operational Research*, *142*(3), 523–538. https://doi.org/10.1016/S0377-2217(01)00314-9

MacCormack, A., Baldwin, C., & Rusnak, J. (2012). Exploring the duality between product and organizational architectures: A test of the “mirroring” hypothesis. *Research Policy*, *41*(8), 1309–1324. https://doi.org/10.1016/j.respol.2012.04.011

Macduffie, J. P., Sethuraman, K., & Fisher, M. (1996). Product Variety and Manufacturing Performance: Evidence from the International Automotive Assembly Plant Study. *Management Science*, *42*(3), 350–369. https://doi.org/10.1287/mnsc.42.3.350

Maheswari, J. U., & Varghese, K. (2005). Project Scheduling using Dependency Structure Matrix. *International Journal of Project Management*, *23*(3), 223–230. https://doi.org/10.1016/j.ijproman.2004.10.001

Mahoney, J. T. (1995). The management of resources and the resource of management. *Journal of Business Research*, *33*(2), 91–101. https://doi.org/10.1016/0148-2963(94)00060-R

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, *1*(4), 267–284. https://doi.org/10.1002/j.2334-5837.1996.tb02054.x

Maimon, O. Z., Dar-El, E. M., & Carmon, T. F. (1993). Set-up saving schemes for printed circuit boards assembly. *European Journal of Operational Research*, *70*(2), 177–190. https://doi.org/10.1016/0377-2217(93)90037-N

Malmi, T. (1999). Activity-based costing diffusion across organizations: an exploratory empirical analysis of Finnish firms. *Accounting, Organizations and Society*, *24*(8), 649–672. https://doi.org/10.1016/S0361-3682(99)00011-2

Marshall, A. (1920). *Industry and Trade*. *History of Economic Thought Books*. McMaster University Archive for the History of Economic Thought. Retrieved from https://ideas.repec.org/b/hay/hetboo/marshall1920.html

Marti, M. (2007). *Complexity Management*. Wiesbaden: DUV. https://doi.org/10.1007/978-3-8350-5435-6

Martin, M. V., & Ishii, K. (1996). Design for Variety: A Methodology for Understanding the Costs of Product Proliferation. In *Volume 4: 8th International Conference on Design Theory and Methodology*. American Society of Mechanical Engineers. https://doi.org/10.1115/96-DETC/DTM-1610

Martin, M. V., & Ishii, K. (2002). Design for variety: developing standardized and modularized product platform architectures. *Research in Engineering Design*, *13*(4), 213–235. https://doi.org/10.1007/s00163-002-0020-2

Maurer, M. (2007). *Structural awareness in complex product design.* Zugl.: München, Techn. Univ., Diss, 2007, München.

Maurer, M. (2017). *Complexity Management in Engineering Design – a Primer*. Berlin: Springer Vieweg. https://doi.org/10.1007/978-3-662-53448-9

McCabe, T. J. (1976). A Complexity Measure. *IEEE Transactions on Software Engineering*, *SE-2*(4), 308–320. https://doi.org/10.1109/TSE.1976.233837

McClelland, B. J. (1971). Properties of the Latent Roots of a Matrix: The Estimation of π -Electron Energies. *The Journal of Chemical Physics*, *54*(2), 640–643. https://doi.org/10.1063/1.1674889

McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, 105.

McKay, A., Erens, F., & Bloor, M. S. (1996). Relating product definition and product variety. *Research in Engineering Design*, *8*(2), 63–80. https://doi.org/10.1007/BF01607862

Medeiros, C. A. (2000). High wage-economy, sloanism and fordism: the American experience during the golden age. *Contributions to Political Economy*, *19*(1), 33–52. https://doi.org/10.1093/cpe/19.1.33

Meffert, H., Burmann, C., & Kirchgeorg, M. (2015). *Marketing*. Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-02344-7

Meijer, B. R. (2006). *Organization structures for dealing with complexity.*

Meinrenken, C. J., Sauerhaft, B. C., Garvan, A. N., & Lackner, K. S. (2014). Combining Life Cycle Assessment with Data Science to Inform Portfolio-Level Value-Chain Engineering. *Journal of Industrial Ecology*, *18*(5), 641–651. https://doi.org/10.1111/jiec.12182

Mertens, K. G. (2020). *Measure and manage your product costs right – development and use of an extended axiomatic design for cost modeling.* TUHH Universitätsbibliothek. https://doi.org/10.15480/882.2888

Mertens, K. G., & Meyer, M. [Matthias] (2018). Wie schlimm sind Messfehler für die Kostenrechnung? *Controlling & Management Review*, *18*(9). Retrieved from https://www.springerprofessional.de/en/wie-schlimm-sind-messfehler-fuer-die-kostenrechnung/16338210

Mertens, K. G., Rennpferdt, C., Greve, E. [Erik], Krause, D., & Meyer, M. [Matthias] (2022). Reviewing the intellectual structure of product modularization: Toward a common view and future research agenda. *Journal of Product Innovation Management.* Advance online publication. https://doi.org/10.1111/jpim.12642

Mertens, K. G., Schmidt, M., Yildiz, T., & Meyer, M. [Matthias] (2021). Introducing a Framework to Generate and Evaluate the Cost Effects of Product (Family) Concepts. *Proceedings of the Design Society*, *1*, 1907–1916. https://doi.org/10.1017/pds.2021.452

Meßerschmidt, O. J., Gumpinger, T., Meyer, M. [Matthias], & Mertens, K. G. (2020). Reviewing Complexity Costs - What Practice Needs and What Research Contributes. *Proceedings of the Design Society: DESIGN Conference*, *1*, 647–656. https://doi.org/10.1017/dsd.2020.152

Meyer, M. [Marc] (1997). Revitalize Your Product Lines Through Continuous Platform Renewal. *Research-Technology Management*, *40*(2), 17–28. https://doi.org/10.1080/08956308.1997.11671113

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

Meyer, M. [Matthias], Meßerschmidt, O. J., & Mertens, K. G. (2019). How much does variety-induced complexity actually cost? Linking axiomatic design with cost modelling. In M. Schröder & K. Wegner (Eds.), *Logistik im Wandel der Zeit – Von der Produktionssteuerung zu vernetzten Supply Chains* (pp. 813–827). Wiesbaden: Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-25412-4_39

Microsoft (2004, August 11). *Microsoft Announces Windows XP Starter Edition Pilot Program* [Press release]. Retrieved from https://news.microsoft.com/2004/08/11/microsoft-announces-windows-xp-starter-edition-pilot-program/

Miller, G., & Vollmann, T. (1985). The hidden factory. *Harvard Business Review*, *63*(5), 142.

Miller, J. G. (1965). Living systems: Basic concepts. *Behavioral Science*, *10*(3), 193–237. https://doi.org/10.1002/bs.3830100302

Min, G., Suh, E. S., & Hölttä-Otto, K. (2016). System Architecture, Level of Decomposition, and Structural Complexity: Analysis and Observations. *Journal of Mechanical Design*, *138*(2). https://doi.org/10.1115/1.4032091

Modrak, V., & Bednar, S. (2015). Using Axiomatic Design and Entropy to Measure Complexity in Mass Customization. *Procedia CIRP*, *34*, 87–92. https://doi.org/10.1016/j.procir.2015.07.013

Morkos, B., Shankar, P., & Summers, J. d. (2012). Predicting requirement change propagation, using higher order design structure matrices: an industry case study. *Journal of Mechanical Design*, *23*(12), 905–926. https://doi.org/10.1080/09544828.2012.662273

Muller, P. (1990). Airbus: Partners and Paradoxes. *The European Journal of International Affairs*, *8*, 25–45.

Nahmias, S. (2011). *Production and Operations Analysis* (6th. ed.). Boston [u.a.]: McGraw-Hill.

Neff, M. (2016, June 22). *Komplexität und Varianz – von Ursachen, Lösungen und deren Wirken*, Berlin.

Nelson, R. R. (1991). Why do firms differ, and how does it matter? *Strategic Management Journal*, *12*(S2), 61–74. https://doi.org/10.1002/smj.4250121006

Nelson, R. R., & Winter, S. G. (1982). *An Evolutionary Theory of Economic Change*. Cambridge, Mass. u.a.: The Belknap Press of Harvard Univ. Press.

Neumann, B., & Cauvin, E. (2007). French cost accounting methods: ABC and other structural similarities. *Journal of Cost Management*, *21*, 35–41. Retrieved from https://api.semanticscholar.org/CorpusID:54642108

Newman, M. E. J. (2006). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(23), 8577–8582. https://doi.org/10.1073/pnas.0601602103

Ocampo y Vilas, C., & Vandaele, N. (2002). A cost and operations based product heterogeneity index. *International Journal of Production Economics*, *79*(1), 45–55. https://doi.org/10.1016/S0925-5273(02)00103-2

O'Leary-Kelly, S. W., & Vokurka, R. J. (1998). The empirical assessment of construct validity. *Journal of Operations Management*, *16*(4), 387–405. https://doi.org/10.1016/S0272-6963(98)00020-5

Orton, J. D., & Weick, K. E. (1990). Loosely Coupled Systems: A Reconceptualization. *The Academy of Management Review*, *15*(2), 203–223. https://doi.org/10.5465/amr.1990.4308154

Otto, K. N., & Wood, K. L. (2001). *Product design: Techniques in reverse engineering and new product development*. Upper Saddle River, N.J.: Prentice Hall.

Page, S. E. (2010). *Diversity and Complexity*. Princeton: Princeton University Press. https://doi.org/10.1515/9781400835140

Pandremenos, J., & Chryssolouris, G. (2011). A neural network approach for the development of modular product architectures. *International Journal of Computer Integrated Manufacturing*, *24*(10), 879–887. https://doi.org/10.1080/0951192X.2011.602361

Panzar, J. C., & Willig, R. D. (1977). Economies of Scale in Multi-Output Production. *The Quarterly Journal of Economics*, *91*(3), 481. https://doi.org/10.2307/1885979

Park, K., & Okudan Kremer, G. E. (2015). Assessment of static complexity in design and manufacturing of a product family and its impact on manufacturing performance. *International Journal of Production Economics*, *169*, 215–232. https://doi.org/10.1016/j.ijpe.2015.07.036

Passerini, F., & Severini, S. (2009). Quantifying Complexity in Networks. *International Journal of Agent Technologies and Systems*, *1*(4), 58–67. https://doi.org/10.4018/jats.2009071005

Penrose, E. (1959). *The Theory of the Growth of the Firm*. New York: John Wiley and Sons.

Perera, H., Nagarur, N., & Tabucanon, M. T. (1999). Component part standardization: A way to reduce the life-cycle costs of products. *International Journal of Production Economics*, *60-61*, 109–116. https://doi.org/10.1016/S0925-5273(98)00179-0

Piattini, M., Genero, M., & Jiménez, L. (2001). A metric-based Approaches for Predicting Conceptual Data Models Maintainability. *International Journal of Software Engineering and Knowledge Engineering*, *11*(06), 703–729. https://doi.org/10.1142/S0218194001000736

Piller, F. T. (2003). *Mass customization: Ein wettbewerbsstrategisches Konzept im Informationszeitalter* (3., überarb. und erw. Aufl.). *Gabler Edition Wissenschaft*. Wiesbaden: Dt. Univ.-Verl.

Pimmler, T. U., & Eppinger, S. D. (1994). Integration Analysis of Product Decompositions. In *6th International Conference on Design Theory and Methodology* (pp. 343–351). American Society of Mechanical Engineers. https://doi.org/10.1115/DETC1994-0034

Pine, B. J., & Kotha, S. (1994). Mass Customization: The New Frontier in Business Competition. *The Academy of Management Review*, *19*(3), 588. https://doi.org/10.2307/258941

Prather, R. E. (1984). An Axiomatic Theory of Software Complexity Measure. *The Computer Journal*, *27*(4), 340–347. https://doi.org/10.1093/comjnl/27.4.340

Puls, C. (2003). *Die Konfigurations- und Vertraeglichkeitsmatrix als Beitrag zum Management von Konfigurationswissen in KMU.* ETH Zurich.

Querbes, A., & Frenken, K. (2018). Grounding the "mirroring hypothesis": Towards a general theory of organization design in New Product Development. *Journal of Engineering and Technology Management*, *47*, 81–95. https://doi.org/10.1016/j.jengtecman.2018.01.001

Rabbath, C. A., & Léchevin, N. (2014). *Discrete-Time Control System Design with Applications*. New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-9290-0

Rao, U., Swaminathan, J. M., & Zhang, J. (2004). Multi-product inventory planning with downward substitution, stochastic demand and setup costs. *IIE Transactions*, *36*(1), 59–71. https://doi.org/10.1080/07408170490247304

Rao, V. R. (2014). *Applied Conjoint Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-87753-0

Ravix, J.-L. (2012). Alfred Marshall and the Marshallian Theory of the Firm. In M. Dietrich & J. Krafft (Eds.), *Handbook on the Economics and Theory of the Firm* (pp. 49–54). Edward Elgar Publishing.

Rebentisch, E., Schuh, G. [G.], Riesener, M., Gerlach, M., & Zeller, P. (2016). Determination of a Customer Value-oriented Product Portfolio. *Procedia CIRP*, *50*, 82–87. https://doi.org/10.1016/j.procir.2016.04.165

Reich, L. S. (1985). *The making of American industrial research : science and business at GE and Bell, 1876-1926 / Leonard S. Reich*. *Studies in economic history and policy*. Cambridge: Cambridge University Press.

Rezaie, K., Ostadi, B., & Torabi, S. A. (2008). Activity-based costing in flexible manufacturing systems with a case study in a forging industry. *International Journal of Production Research*, *46*(4), 1047–1069. https://doi.org/10.1080/00207540600988121

Ripperda, S., & Krause, D. (2017). Cost Effects of Modular Product Family Structures: Methods and Quantification of Impacts to Support Decision Making. *Journal of Mechanical Design*, *139*(2), 021103-1 - 021103-12. https://doi.org/10.1115/1.4035430

Robertson, D., & Ulrich, K. (1998). Planning for Product Platforms. *Sloan Management Review*, *39*(4), 19.

Robinson, S. (2008). Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*, *59*(3), 278–290. https://doi.org/10.1057/palgrave.jors.2602368

Roethlisberger, F. J. (1977). *The elusive phenomena: An autobiographical account of my work in the field of organizational behavior at the Harvard Business School*. Boston: Division of Research, Graduate School of Business Administration, Harvard University.

Rogers, D. F. (1992). *Laminar flow analysis*. Cambridge <etc.>: Cambridge Univ.

Rothschild, E. (1974). *Paradise lost : the decline of the auto-industrial age*. New York: Vintage Books.

Roy, R., Evans, R., Low, M. J., & Williams, D. K. (2011). Addressing the impact of high levels of product variety on complexity in design and manufacture. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *225*(10), 1939–1950. https://doi.org/10.1177/0954405411407670

Ruiz-Torres, A. J., & Mahmoodi, F. (2008). Analysis of Multi-Cell Production Systems Considering Cell Size and Worker Flexibility. *International Journal Industrial Engineering*, *15*(4), 360-372.

Russell, R. S., & Taylor, B. W. (2005). *Operations management: Quality and competitiveness in a global environment* (5th ed.). Milwaukee, Wis.: Volunteer Services for the Visually Handicapped.

Salvador, F., Forza, C. [C.], & Rungtusanatham, M. (2002). Modularity, product variety, production volume, and component sourcing: theorizing beyond generic prescriptions. *Journal of Operations Management*, *20*(5), 549–575. https://doi.org/10.1016/S0272-6963(02)00027-X

Santos, V., Sampaio, M., & Alliprandini, D. H. (2020). The impact of product variety on fill rate, inventory and sales performance in the consumer goods industry. *Journal of Manufacturing Technology Management*, *31*(7), 1481–1505. https://doi.org/10.1108/JMTM-06-2019-0213

Sawai, K., Nomaguchi, Y., & Fujita, K. (2017). Case study of extended product architecture design for modularization reflecting customer needs of industrial robots. *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, *11*(4), JAMDSM0050-JAMDSM0050. https://doi.org/10.1299/jamdsm.2017jamdsm0050

Sawhney, M. S. (1998). Leveraged High-Variety Strategies: From Portfolio Thinking to Platform Thinking. *Journal of the Academy of Marketing Science*, *26*(1), 54–61. https://doi.org/10.1177/0092070398261006

Schlick, C. M., Beutner, E., Duckwitz, S., & Licht, T. (2007). A complexity measure for new product development projects. In *2007 IEEE International Engineering Management Conference* (pp. 143–150). IEEE. https://doi.org/10.1109/IEMC.2007.5235079

Schmidt, M., Mertens, K. G., & Meyer, M. [Matthias] (2023). Cost hierarchies and the pattern of product cost cross-subsidization: Extending a computational model of costing system design. *PloS One*, *18*(9), e0290370. https://doi.org/10.1371/journal.pone.0290370

Schoute, M. (2011). The relationship between product diversity, usage of advanced manufacturing technologies and activity-based costing adoption. *The British Accounting Review*, *43*(2), 120–134. https://doi.org/10.1016/j.bar.2011.02.002

Schuh, G. [Günther] (2005). *Produktkomplexität managen: Strategien - Methoden - Tools* (2nd, revised and extended edition). München, Wien: Hanser.

Schuh, G. [Günther], Riesener, M., & Rudolf, S. (2014). Identifying Preferable Product Variants Using Similarity Analysis. *Procedia CIRP*, *20*, 38–43. https://doi.org/10.1016/j.procir.2014.05.029

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, *27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sharman, D. M., & Yassine, A. A. (2004). Characterizing complex product architectures. *Systems Engineering*, *7*(1), 35–60. https://doi.org/10.1002/sys.10056

Shea, V. J., Waldrup, B. E., Xu, H. [Helen], & Williamson, S. (2018). Error Rate Impacts On Decision Efficacy: Activity-Based Costing Systems In Small Business. *QUARTERLY REVIEW of BUSINESS DISCIPLINES*, *5*(1), 59.

Sheard, S. A., & Mostashari, A. (2009). Principles of complex systems for systems engineering. *Systems Engineering*, *12*(4), 295–311. https://doi.org/10.1002/sys.20124

Shi, Q., & Blomquist, T. (2012). A new approach for project scheduling using fuzzy dependency structure matrix. *International Journal of Project Management*, *30*(4), 503–510. https://doi.org/10.1016/j.ijproman.2011.11.003

Shields, M. (1995). An empirical analysis of firms' implementation experiences with activity-based costing. *Journal of Management Accounting Research*, *7*, 148.

Shirwaiker, R. A., & Okudan, G. E. (2008). Triz and axiomatic design: a review of case-studies and a proposed synergistic use. *Journal of Intelligent Manufacturing*, *19*(1), 33–47. https://doi.org/10.1007/s10845-007-0044-6

Sickles, R. C., & Zelenyuk, V. (2019). *Measurement of Productivity and Efficiency*. Cambridge University Press. https://doi.org/10.1017/9781139565981

Simon, H. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society*, *106*(6), 467–482.

Simpson, T. W. (2004). Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *18*(1), 3–20. https://doi.org/10.1017/s0890060404040028

Simpson, T. W., & D'Souza, B. S. (2004). Assessing Variable Levels of Platform Commonality Within a Product Family Using a Multiobjective Genetic Algorithm. *Concurrent Engineering*, *12*(2), 119–129. https://doi.org/10.1177/1063293X04044383

Sinha, K. (2014). *Structural complexity and its implications for design of cyber-physical systems* (Thesis: Ph. D.). Massachusetts Institute of Technology (MIT), Boston, MA. Retrieved from http://hdl.handle.net/1721.1/89871

Sinha, K., & Suh, E. S. (2018). Pareto-optimization of complex system architecture for structural complexity and modularity. *Research in Engineering Design*, *29*(1), 123–141. https://doi.org/10.1007/s00163-017-0260-9

Sinha, K., & Weck, O. L. de [Olivier L.] (2012). STRUCTURAL COMPLEXITY METRIC FOR ENGINEERED COMPLEX SYSTEMS AND ITS APPLICATION. In M. Onishi, M. Maurer, K. Kirner, & U. Lindemann (Eds.), *Gain competitive advantage by managing complexity* (pp. 181–192). München: Carl Hanser Verlag GmbH & Co. KG. https://doi.org/10.3139/9783446434127.015

Sinha, K., & Weck, O. L. de [O. L.] (2013a). A network-based structural complexity metric for engineered complex systems. In *2013 IEEE International Systems Conference (SysCon)* (pp. 426–430). IEEE. https://doi.org/10.1109/SysCon.2013.6549917

Sinha, K., & Weck, O. L. de [Olivier L.] (2013b). Structural Complexity Quantification for Engineered Complex Systems and Implications on System Architecture and Design. In *Volume 3A: 39th Design Automation Conference*. American Society of Mechanical Engineers. https://doi.org/10.1115/DETC2013-12013

Skirde, H., Kersten, W., & Schröder, M. (2016). Measuring the Cost Effects of Modular Product Architectures — A Conceptual Approach. *International Journal of Innovation and Technology Management*, *13*(04), 1650017-1-1650017-23. https://doi.org/10.1142/S0219877016500176

Sorkun, M. F., & Furlan, A. (2017). Product and Organizational Modularity: A Contingent View of the Mirroring Hypothesis. *European Management Review*, *14*(2), 205–224. https://doi.org/10.1111/emre.12101

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2004). The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. *Management Science*, *50*(12), 1674–1689. https://doi.org/10.1287/mnsc.1040.0289

Stäblein, T., Holweg, M., & Miemczyk, J. (2011). Theoretical versus actual product variety: how much customisation do customers really demand? *International Journal of Operations & Production Management*, *31*(3), 350–370. https://doi.org/10.1108/01443571111111955

Steward, D. V. (1981). The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, *EM-28*(3), 71–74. https://doi.org/10.1109/TEM.1981.6448589

Subramanian, R., Ferguson, M. E., & Beril Toktay, L. (2013). Remanufacturing and the Component Commonality Decision. *Production and Operations Management*, *22*(1), 36–53. https://doi.org/10.1111/j.1937-5956.2012.01350.x

Suh, N. P. (1990). *The principles of design*. *Oxford series on advanced manufacturing*. New York u.a.: Oxford Univ. Press.

Suh, N. P. (1995). Axiomatic Design of Mechanical Systems. *Journal of Mechanical Design*, *117*(B), 2–10. https://doi.org/10.1115/1.2836467

Suh, N. P. (2001). *Axiomatic design: Advances and applications*. *The MIT-Pappalardo series in mechanical engineering*. New York: Oxford University Press.

Suh, N. P. (2005). *Complexity : Theory and applications*. *MIT-Pappalardo series in mechanical engineering*. Oxford: Oxford Univ. Press.

Suh, N. P., Cavique, M., & Foley, J. T. (2021). *Design Engineering and Science*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-49232-8

Summers, J. d., & Shah, J. J. (2010). Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. *Journal of Mechanical Design*, *132*(2). https://doi.org/10.1115/1.4000759

Sussman, J. M. (1999). *The New Transportation Faculty: The Evolution to Engineering Systems*. Cambridge.

Swamidass, P. M. (Ed.) (2000). *Encyclopedia of Production and Manufacturing Management*. Springer US. https://doi.org/10.1007/1-4020-0612-8

Swaminathan, J. M. (2001). Enabling Customization Using Standardized Operations. *California Management Review*, *43*(3), 125–135. https://doi.org/10.2307/41166092

Swaminathan, J. M., & Lee, H. L. (2003). Design for Postponement. In *Handbooks in Operations Research and Management Science. Supply Chain Management: Design,* *Coordination and Operation* (Vol. 11, pp. 199–226). Elsevier. https://doi.org/10.1016/S0927-0507(03)11005-5

Syam, S. S., & Bhatnagar, A. (2015). A decision support model for determining the level of product variety with marketing and supply chain considerations. *Journal of Retailing and Consumer Services*, *25*, 12–21. https://doi.org/10.1016/j.jretconser.2015.03.004

Takai, S. (2019). An Approach to Integrate Commonality and Product Family Design With Inventory Decisions. *Journal of Mechanical Design*, *141*(3). https://doi.org/10.1115/1.4042340

Takai, S., & Sengupta, S. (2017). An Approach to Evaluate the Profitability of Component Commonality. *Journal of Mechanical Design*, *139*(7). https://doi.org/10.1115/1.4036644

Tan, T. F., Netessine, S., & Hitt, L. (2017). Is Tom Cruise Threatened? An Empirical Study of the Impact of Product Variety on Demand Concentration. *Information Systems Research*, *28*(3), 643–660. https://doi.org/10.1287/isre.2017.0712

Tao, F., Xiao, B., Qi, Q., Cheng, J., & Ji, P. (2022). Digital twin modeling. *Journal of Manufacturing Systems*, *64*, 372–389. https://doi.org/10.1016/j.jmsy.2022.06.015

Teboul, J. (1991). *Managing quality dynamics*. London: Prentice Hall.

Teece, D. J., Pisano, G., & Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*, *18*(7), 509–533. https://doi.org/10.1002/(SICI)1097-0266(199708)18:7<509::AID-SMJ882>3.0.CO;2-Z

Terwiesch, C., & Loch, C. H. (1999). Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development. *Journal of Product Innovation Management*, *16*(2), 160–172. https://doi.org/10.1111/1540-5885.1620160

Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management – from its history to its new role. *International Journal of Product Lifecycle Management*, *4*(4), 360. https://doi.org/10.1504/IJPLM.2010.036489

Thevenot, H. J., & Simpson, T. W. (2009). A Product Dissection-Based Methodology to Benchmark Product Family Design Alternatives. *Journal of Mechanical Design (New York, N.Y. : 1990)*, *131*(4). https://doi.org/10.1115/1.3086789

Thomas, L. D. (1992). Functional implications of component commonality in operational systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(3), 548–551. https://doi.org/10.1109/21.155954

Thompson, D. V., Hamilton, R. W., & Rust, R. T. (2005). Feature Fatigue: When Product Capabilities Become Too Much of a Good Thing. *Journal of Marketing Research*, *42*(4), 431–442. https://doi.org/10.1509/jmkr.2005.42.4.431

Thonemann, U. W., & Brandeau, M. L. (2000). Optimal Commonality in Component Design. *Operations Research*, *48*(1), 1–19. https://doi.org/10.1287/opre.48.1.1.12445

Togay, C., Dogru, A. H., & Tanik, J. U. (2008). Systematic Component-Oriented development with Axiomatic Design. *Journal of Systems and Software*, *81*(11), 1803–1815. https://doi.org/10.1016/j.jss.2007.12.746

Tolonen, A., Shahmarichatghieh, M., Harkonen, J., & Haapasalo, H. (2015). Product portfolio management – Targets and key performance indicators for product portfolio renewal over life cycle. *International Journal of Production Economics*, *170*, 468–477. https://doi.org/10.1016/j.ijpe.2015.05.034

Toma, S. (2016). U.S. Turn Signals to Euro Style Turn Signals Conversion And Viceversa. Retrieved from https://www.autoevolution.com/news/us-turn-signals-to-euro-style-turn-signals-conversion-and-viceversa-113358.html

Trattner, A., Hvam, L., Forza, C. [Cipriano], & Herbert-Hansen, Z. N. L. (2019). Product complexity and operational performance: A systematic literature review. *CIRP Journal of Manufacturing Science and Technology*, *25*, 69–83. https://doi.org/10.1016/j.cirpj.2019.02.001

Tsafarakis, S., Grigoroudis, E., & Matsatsinis, N. (2011). Consumer choice behaviour and new product development: an integrated market simulation approach. *Journal of the Operational Research Society*, *62*(7), 1253–1267. https://doi.org/10.1057/jors.2010.70

Ulrich, K. (1994). Fundamentals of Product Modularity. In S. Dasu & C. Eastman (Eds.), *Management of Design* (Vol. 14, pp. 219–231). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-011-1390-8_12

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, *24*(3), 419–440. https://doi.org/10.1016/0048-7333(94)00775-3

Ulrich, K., & Eppinger, S. D. (2016). *Product design and development* (Sixth edition). New York, NY: McGraw-Hill Education.

Vachon, S., & Klassen, R. D. (2002). An exploratory investigation of the effects of supply chain complexity on delivery performance. *IEEE Transactions on Engineering Management*, *49*(3), 218–230. https://doi.org/10.1109/TEM.2002.803387

Vahl Davis, G. de (1986). *Numerical Methods in Engineering & Science*. Dordrecht: Springer Netherlands.

Van Mieghem, J. A. (2003). Capacity Management, Investment, and Hedging: Review and Recent Developments. *Manufacturing & Service Operations Management*, *5*(4), 269–302. https://doi.org/10.1287/msom.5.4.269.24882

Villas-Boas, J. M. (2009). Product Variety and Endogenous Pricing with Evaluation Costs. *Management Science*, *55*(8), 1338–1346. https://doi.org/10.1287/mnsc.1090.1024

Visser, M. d., Weerd-Nederhof, P. d., Faems, D., Song, M., van Looy, B., & Visscher, K. (2010). Structural ambidexterity in NPD processes: A firm-level assessment of the impact of differentiated structures on innovation performance. *Technovation*, *30*(5), 291–299. https://doi.org/10.1016/j.technovation.2009.09.008

Vogel, W., & Lasch, R. (2016). Complexity drivers in manufacturing companies: a literature review. *Logistics Research*, *9*(1). https://doi.org/10.1007/s12159-016-0152-9

Wan, X., Evers, P. T., & Dresner, M. E. (2012). Too much of a good thing: The impact of product variety on operations and sales performance. *Journal of Operations Management*, *30*(4), 316–324. https://doi.org/10.1016/j.jom.2011.12.002

Wan, X., & Sanders, N. R. (2017). The negative impact of product variety: Forecast bias, inventory levels, and the role of vertical integration. *International Journal of Production Economics*, *186*, 123–131. https://doi.org/10.1016/j.ijpe.2017.02.002

Weber, C. (2007). Looking at “DFX” and “Product Maturity” from Product and Product Development Processes. In F.-L. Krause (Ed.), *The Future of Product Development* (pp. 85–104). Berlin, Heidelberg: Springer Berlin Heidelberg.

Weber, J. (2022). regressive Kosten. In Springer Gabler Verlag (Ed.), *Gabler Wirtschaftslexikon*. Retrieved from https://wirtschaftslexikon.gabler.de/definition/regressive-kosten-44380/version-267691

Wernerfelt, B. (1984). A Resource-Based View of the Firm. *Strategic Management Journal*, *5*(2), 171–180. Retrieved from http://www.jstor.org/stable/2486175

Weyuker, E. J. (1988). Evaluating software complexity measures. *IEEE Transactions on Software Engineering*, *14*(9), 1357–1365. https://doi.org/10.1109/32.6178

Whitfield, R. I., Smith, J. S., & Duffy, A. B. (2002). Identifying Component Modules. In J. S. Gero (Ed.), *Artificial Intelligence in Design ’02* (pp. 571–592). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-017-0795-4_27

Wilson, S. A., & Perumal, A. (2010). *Waging war on complexity costs: Reshape your cost structure, free up cash flows, and boost productivity by attacking process, product, and organizational complexity*. New York: McGraw-Hill. Retrieved from http://www.loc.gov/catdir/enhancements/fy1011/2009046358-b.html

Womack, J. P. (1990). *The machine that changed the world : based on the Massachusetts Institute of Technology 5-million-dollar 5-year study on the future of the automobile*. New York, NY: Rawson Associates [u.a.].

Wouters, M., Morales, S., Grollmuss, S., & Scheer, M. (2016). Methods for Cost Management During Product Development: a Review and Comparison of Different Literatures. In M. J. Epstein & M. A. Malina (Eds.), *Advances in Management Accounting: v. 26. Advances in management accounting* (pp. 139–274). Bingley, U.K.: Emerald.

Wouters, M., & Stadtherr, F. (2017). Cost management and modular product design strategies. In E. Harris (Ed.), *The Routledge Companion to Performance Management and Control*. Title: The Routledge companion to performance management and control / edited by Elaine Harris. Description: Abingdon, Oxon, New York, NY: Routledge, 2017. | Series: Routledge companions in business, management and accounting: Routledge.

Wouters, M., & Stecher, J. (2017). Development of real-time product cost measurement: A case study in a medium-sized manufacturing company. *International Journal of Production Economics*, *183*, 235–244. https://doi.org/10.1016/j.ijpe.2016.10.018

Xu, H. [He], Zhang, L., Li, P., & Zhu, F. (2022). Outlier detection algorithm based on k-nearest neighbors-local outlier factor. *Journal of Algorithms & Computational Technology*, *16*, 174830262210781. https://doi.org/10.1177/17483026221078111

Yang, J., & Deane, R. H. (1993). Setup time reduction and competitive advantage in a closed manufacturing cell. *European Journal of Operational Research*, *69*(3), 413–423. https://doi.org/10.1016/0377-2217(93)90025-I

Yelle, L. E. (1979). The Learning Curve: Historical Review and Comprehensive Survey. *Decision Sciences*, *10*(2), 302–328. https://doi.org/10.1111/j.1540-5915.1979.tb00026.x

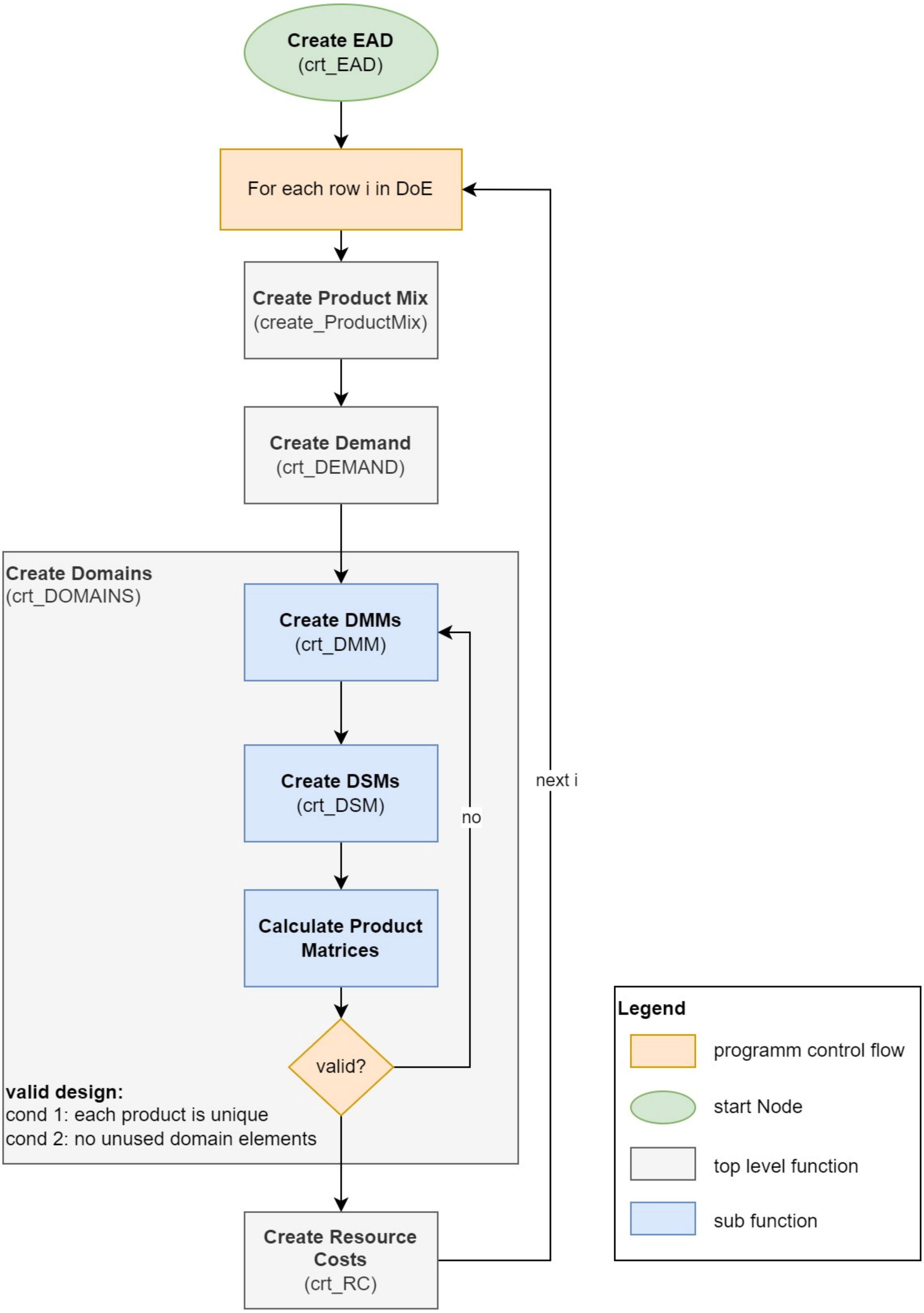
Zhang, L. L., Lee, C. K., & Akhtar, P. (2020). Towards customization: Evaluation of integrated sales, product, and production configuration. *International Journal of Production Economics*, *229*, 107775. https://doi.org/10.1016/j.ijpe.2020.107775

Zhang, M., & Tseng, M. M. (2007). A Product and Process Modeling Based Approach to Study Cost Implications of Product Variety in Mass Customization. *IEEE Transactions on Engineering Management*, *54*(1), 130–144. https://doi.org/10.1109/TEM.2006.889072

Zhang, N., Yang, Y., Zheng, Y., & Su, J. (2019). Module partition of complex mechanical products based on weighted complex networks. *Journal of Intelligent Manufacturing*, *30*(4), 1973–1998. https://doi.org/10.1007/s10845-017-1367-6

Appendix

A1 Control Flow Chart for the Generation of Product Family Designs



A2 Proof of Construct Validity for Interface Complexity (HIC)

This section proves that the interface complexity measure (HIC) as defined by Hennig et al. (2022) fulfills construct validity criteria described in section 3.3.2. The measure is defined as:

$$HIC(DSM_{bin}) = \sum E \qquad \text{(A2.1)}$$

and depends on the number of connections (E) within a binary matrix $DSM_{bin} \in \mathbb{N}_{0,1}{}^{n\,x\,n}$. Since it reflects interrelatedness the criteria C4-C6 needs to be checked to proof construct validity.

Monotony Condition (C1)
The monotony condition states that the complexity measure must be positive for all systems since there is no system that is less complex than the reference simplicity. HIC fulfills the condition since:

$$DSM_{bin} \in \mathbb{N}_{0,1}{}^{n\,x\,n} \rightarrow \; E \geq 0. \qquad \text{(A2.2)}$$

Null Value Condition (C4)
The null value condition for interrelatedness (C4) states that a system without relations ($E = 0$) has a complexity value of zero. For a system without any relations:

$$HIC(DSM_{bin}) = \sum E = 0 \qquad \text{(A2.3)}$$

which proves that C4 is fulfilled.

Monotony Condition (C5)
The monotony condition assumes that the complexity of two subsystems (P, Q) which share no relations among each other but can share elements is always less or equal to the complexity of the combined system (S). It needs to be shown that:

$$HIC(P) + HIC(Q) \leq HIC(S) \qquad \text{(A2.4)}$$

Since there are no relations between subsystems this formular can be rewritten as:

$$\sum E_P + \sum E_Q = \sum E_P + \sum E_Q = HIC(S) \qquad \text{(A2.5)}$$

Even if these two subsystems share relations (E_{shared}) the condition C5 is fulfilled since:

$$\sum E_P + \sum E_Q \leq \sum E_P + \sum E_Q + \sum E_{shared} \qquad \text{(A2.6)}$$

Disjoint Additivity (C6)
Disjoint additivity states that the complexity of two independent systems which share no elements, and no relations is identical to the complexity of the combined systems. Equation (A2.5) can be rewritten as:

$$HIC(P) + HIC(Q) = \sum E_P + \sum E_Q = \sum E_P + \sum E_Q = HIC(S) \qquad \text{(A2.7)}$$

which proves C6.

A3 Proof of Construct Validity for System Design Complexity (SDC)

The system design complexity (SDC) measures the degree of design coupling within a domain mapping matrix ($DMM \in \mathbb{N}_{0,1}{}^{N_{src} x \, N_{tgt}}$). According to Modrak and Bednar (2015), it is defined as:

$$SDC(DMM) = \sum_{j=1}^{N_{tgt}} \left[\sum_{i=1}^{N_{src}} x_{i,j} * log\left(\sum_{i=1}^{N_{src}} x_{i,j} \right) \right] \qquad \text{(A3.8)}$$

with $x_{i,j}$ being the entries of DMM. For better readability the columns sums of DMM are re-written as:

$$\Sigma_{col,j} DMM = \sum_{i=1}^{N_{src}} x_{i,j} \qquad \text{(A3.9)}$$

Since this measure represents interrelatedness the conditions C4-C6 from section 3.3.2 are checked.

Monotony Condition (C1) and Null Value Condition (C4)
The null value condition for interrelatedness (C4) states that a system without any relations ($E = 0$) has a no complexity. This case does not occur within the EAD since a DMM requires that each row has at least on entry. However, also for completely empty DMMs, SDC fulfills the null value condition since:

$$SDC(DMM_0) = \sum_{j=1}^{N_{tgt}} 0 * ln(0) \qquad \text{(A3.10)}$$

Under the assumption that:

$$0 * \log(0) = 0 \qquad \text{(A3.11)}$$

which is widely applied for the information entropy (Cover & Thomas, 2005). By using this convention, $\min(SDC) = 0$ since the logarithm $\ln(x)$ is a strictly increasing function for $x > 0$ which fulfilled given since $DMM \in \mathbb{N}_{0,1}{}^{N_{src} x \, N_{tgt}}$.

Monotony Condition (C5)
The monotony condition assumes that the complexity of two subsystems (P, Q), sharing no relations among each other but can share elements, is always less or equal the complexity of the combined system (S). It needs to be shown that the following statement is true.

$$SDC(P) + SDC(Q) \leq SDC(S) \qquad \text{(A3.12)}$$

Since both sub-systems do not share any relations, a general system is represented as:

$$S = \begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix}; \qquad \text{(A3.13)}$$

Equation (A3.12) than turns into:

$$\Sigma_{col,j}P * \log\left(\Sigma_{col,j}P\right) + \Sigma_{col,j}Q * \log\left(\Sigma_{col,j}Q\right) \leq \left[\Sigma_{col,j}P + Q\right] * \log\left(\Sigma_{col,j}P + Q\right) \tag{A3.14}$$

Therefore, the proof reduces to show that:

$$p * \ln(p) + q * \ln(q) \leq (p+q) * \ln(p+q) \;\; with \;\; p,q \in \mathbb{N} \tag{A3.15}$$

Where p are column sums of P and q the column sums of Q. Transforming the equation leads to:

$$\frac{p*\ln(p)}{\ln(p+q)} + \frac{q*\ln(q)}{\ln(p+q)} \leq p + q \; with \;\; p,q \in \mathbb{N} \tag{A3.16}$$

which is true if the following conditions are true.

$$\frac{p*\ln(p)}{\ln(p+q)} \leq p \;\wedge\; \frac{q*\ln(q)}{\ln(p+q)} \leq q \; with \;\; p,q \in \mathbb{N} \tag{A3.17}$$

Transforming this equation leads to

$$\frac{\ln(p)}{\ln(p+q)} \leq 1 \;\wedge\; \frac{\ln(q)}{\ln(p+q)} \leq 1 \; with \;\; p,q \in \mathbb{N} \tag{A3.18}$$

Since $p, q \in \mathbb{N}$, the denominator is always larger than the numerator and thus equation (A3.18) is true for all DMMs which proofs condition C5.

Disjoint Additivity (C6)

The disjoint additivity condition states that the complexity of two independent systems, sharing no elements and no relations is identical to the complexity of the combined system (S). Since there are neither shared elements nor shared relations the combined systems is given as:

$$S = \begin{pmatrix} P & 0 \\ 0 & Q \end{pmatrix}; \tag{A3.19}$$

which allows to rewrite equation (A3.12) as

$$SDC(P) + SDC(Q) = SDC(P+0) + SDC(Q+0) = SDC(S) \tag{A3.20}$$

Since $0 * \log(0) = 0$, SDC is not influenced by additional zero terms and thus, the disjoint additivity is proven for this measure.

A4 Design of Experiments for the Complexity Cost Experiment

Group	Variable	Description	low values	high values	Range[1)]
		Independent Variables			
Product Mix & Demand	DNS	Proportion of non-zero entries in P_{FD}	product mix matrix is sparse	product mix is dense	U[.07;.2]
	Q_{VAR}	Log normal standard deviation of the product demand vector	sales are equally distributed across products	some products with a high share on sales and many products with low quantity	U[0;1.7]
Product Design	SDC_N	Normalized system design complexity	uncoupled design	coupled design	U[0;0.08]
	DNS_{DSM}	density of DSM matrices	no interactions within the individual DSM matrices	interactions within the individual DSM matrices	U[0;.14]
Costs	R_{fix}	proportion of fixed costs on indirect costs	majority of total costs are variable costs	majority of total costs are fixed costs	U[.1; .5]
	RC_{sdlog}	log normal standard deviation of the direct cost vector	Homogenously distributed resource costs across resource	some resource account for most of the total costs	U[0;3]
	R_{dvl}	proportion of total development costs on fixed costs	development is responsible for only a small proportion of total costs	development is responsible for a large proportion of total costs	U[.1; .5]
	R_{pa}	proportion of total part administration costs on fixed costs	part management activities are responsible for only a small proportion of total costs	part management activities are responsible for a large proportion of total costs	U[.05;.2]
	$R_{tooling}$	proportion of total tooling costs on fixed costs	activities for tooling development, production, ... are responsible for only a small proportion of total costs	activities for tooling development, production, ... are responsible for a large proportion of total costs	U[.05;.2]
	R_{supply}	proportion of total supplier management costs on fixed costs	activities for supplier management are responsible for only a small proportion of total costs	activities for supplier management are responsible for a large proportion of total costs	U[.05;.2]
	R_{setup}	proportion of total setup costs on variable costs	setup activities are responsible for only a small proportion of total costs	setup activities are responsible for a large proportion of total costs	U[.05;.2]
	R_{order}	proportion of total order costs on variable costs	component's order activities are responsible for only a small proportion of total costs	component's order activities are responsible for a large proportion of total costs	U[.05;.2]
	R_{hold}	proportion of total inventory holding costs on variable costs	inventory holding activities are responsible for only a small proportion of total costs	inventory holding activities are responsible for a large proportion of total costs	U[.05;.2]
		Control Variables			
Product Design	N_{FR}	number of functional requirements	product mix characterized by only a few customers relevant features	product mix characterized by many customers relevant features	30
	N_{PD}	number of components	only a few components necessary to realize the product mix	many components necessary to realize the product mix	50
	N_{PrD}	number of processes	only a few processes necessary to realize the product mix	many processes necessary to realize the product mix	50
	N_{RD}	number of resources	only a few resources necessary to realize the product mix	many resources necessary to realize the product mix	50
Product Mix & Demand	N_{PROD}	number of products	low product variety	high product variety	200
	TD	total number of sales for the created product mix	product mix with low number of total sales	product mix with high number of total sales	4,200
Costs	TC	total costs			1,000,000
MISC	seed	random seed			1234
	N_{RUN}	number of runs			1,000

Note. 1) U[lb;ub] indicates a uniform distribution within the lower (lb) and upper bound (ub).

A5 Design of Experiments for the Cost System Accuracy Experiment

Group	Variable	Description	low values	high values	Range[1)]
Independent Parameter					
Product Mix & Demand	PARAM	Since $method_{PM}$="DNS" is selected it defines the desired level of feature commonality in P_{FD}	functional product mix matrix is sparse	functional product mix is dense	U[.07;.5]
	TD	Total number of sales for the created product mix	product mix with low number of total sales	product mix with high number of total sales	U[100; 12,000]
	Q_{var}	Log normal standard deviation of the product demand vector	Sales are equally distributed across products	some products with a high share on sales and many products with low quantity	U[0;1.7]
Product Family Design	SDC_n	System design complexity, representing the normalized entropy in inter domain mapping matrices	uncoupled design	coupled design	U[0;.08]
	DNS_{DSM}	Density of DSM matrices	no interactions within the individual DSM matrices	interactions within the individual DSM matrices	U[0 .14]
Costs	R_{id}	Proportion of indirect costs on total costs	majority of costs are direct costs	majority of costs are indirect costs	U[.2; .8]
	R_{fix}	Proportion of fixed costs on indirect costs			U[.2; .6]
	RC_{sdlog}	Log normal standard deviation of the direct cost vector	Homogenously distributed resource costs across resource	Some resource account for a majority of the total costs	U[0;3]
Control Parameter					
Product Mix & Demand	$method_{PM}$	Method for generating the product mix.			DNS
	N_{PROD}	Number of Products	Small number of products in the product mix	Large number of products in the product mix	100
Product Family Design	N_{FR}	Number of Elements in the functional domain	only a few functional requirements necessary to realize the product mix	many functional requirements necessary to realize the product mix	50
	N_{DD}	Number of Elements in the physical domain	only a few components necessary to realize the product mix	many components necessary to realize the product mix	50
	N_{PrD}	Number of Elements in the process domain	only a few processes necessary to realize the product mix	many processes necessary to realize the product mix	50
	N_{RD}	Number of Elements in the resource domain	only a few resources necessary to realize the product mix	many resources necessary requirements to realize the product mix	50
Costs	TC	Total costs			1,000,000
Simulation	RUNS	Number of runs	only a few runs per parameter combination	many runs per parameter combination	5000
Simulation	seed	Random seed to reproduce the results			1234

Note. [1)] U[lb;ub] indicates a uniform distribution within the lower (lb) and upper bound (ub).