DTU Library

Deciding on the total number of product architectures

Askhøj, Christoffer; Mortensen, Niels Henrik

Published in:
Concurrent Engineering: Research and Applications

Link to article, DOI:
10.1177/1063293X19888968

Publication date:
2020

Document Version
Peer reviewed version

Deciding on the total number of product architectures

Christoffer Askhøj[1] & Niels Henrik Mortensen[1]

[1]Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby, Denmark

Corresponding author:

Niels Henrik Mortensen, Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark, Building 404, DK-2800 Kongens Lyngby, Denmark.

Email: nhmo@mek.dtu.dk

Abstract

Many industries have to deliver mass customised products rather than mass produced products in today's markets, and companies often aim to accommodate this by introducing modular product architectures. However, today's methods do not explicitly support the decision on how many architectures a company's entire product programme should be based on. Therefore, this article presents a method to help companies decide on the total number of product architectures they require. The method goes through four stages: market segmentation, mapping new generation with an existing architecture strategy, architecture changes and the new architecture strategy. Essentially, the method is a way of model- ling and challenging the total number of architectures in a company. The method was tested on a case company and helped it to reduce its number of architectures from 19 to 7 while at the same time introducing new product variants. Cost reductions due to a higher level of commonality were achieved and the new product architectures were prepared for future innovations.

Keywords

Product architecture, modularization, concurrent engineering, product platform, architecture strategy

Introduction

In past decades, many industries have experienced a shift in demand from mass produced products to mass customised products (Pine, 1993). In practice, this means that companies must develop a much larger portfolio of products. Often, product portfolios have been developed sequentially, product-by-product, resulting in overlap in design solutions, the coupling of product functionalities and generally increased internal complexity. As a result, many companies have found that costs increase faster than turnover and that the time to market for new products is extended (Meyer and Lehnerd, 1997; Wilson and Perumal, 2009).

One way to cope with the challenges of offering sufficient product variance to the market, while at the same time maintaining economies of scale and low time to market, is facilitating product-related processes through modular product architectures (Meyer and Utterback, 1993; Robertson and Ulrich, 1998). Modular product architectures not only offer more external variance while keeping internal complexity down, but also offer a technical product division that helps enable concurrent engineering (Prasad, 1996, 1997) in manufacturing companies. If modules are decoupled and interfaces are locked, development can happen in parallel across functions in a company (Prasad, 1999). Planning across the total product portfolio when developing modular product architectures only enhances the possibility of concurrent work.

Much research already exists in the field of modular product architectures, product family design and platform-based product development. However, most of this research focuses on the development of single product platforms, commonality across product families, product portfolio rationalisation etc. (Dahmus et al., 2001; Hansen, 2014; Harlou, 2006). Little research has been conducted on deciding the total number of architectures for a full portfolio of products with multiple product families (Mortensen et al., 2016). Hence, in this paper we aim to address this gap in the research.

The structure of this paper is as follows: First, we establish a common nomenclature and describe the research approach. The section entitled "State of the art" describes the state of the literature. Following this is the section "DNA method (Deciding the Number of Architectures)", which describes the method developed for determining the total number of product architectures for a full product portfolio. Next, the application of the method in a case company is described in the section entitled "Application of the DNA method". The paper also includes a discussion of the method, the results of the study and a final conclusion.

Defining product architectures

Since this paper presents a method for deciding the total number of product architectures in a full product portfolio, it is essential to define product architectures. Also, since the scope of the method lies within modular product architectures, a definition of module is also provided.

Product architecture

Many authors have defined product architectures (Erens and Verhulst, 1997; Henderson and Clark, 1990; Sanchez, 2010; Ulrich and Eppinger, 2012). In this paper, we refer to Sanchez's (2013) definition of modular product architecture. More precisely, we refer to Sanchez's term strategic modularity when we use the term product architecture. The former is defined by:

- The strategic partitioning of a product into functional modules
- Clearly defined interfaces between modules and scalability principles
- Strategic flexibility to introduce a range of component variations that enable the configuration of a range of product variations

Modularity

Modularity is defined here as a one-to-one mapping of a function that is perceived by the customer to be an important source of differentiation onto a single component or subsystem of a component (Sanchez, 2010). Interfaces are designed so that a functional module can be swapped without having to compensate in other areas of the product.

Research approach

The DNA method (Deciding Number of Architectures) was developed through a combination of examining the existing literature, drawing on knowledge from experienced practitioners and implementing the method in a case company.

The state of the art in relation to the developed method was obtained by reading 14 literature reviews on the topic of modularity, platform-based product development, product family design, modular product architectures and variety management (Bonvoisin et al., 2016; Campagnolo and Camuffo, 2010;

ElMaraghy et al., 2013; Gershenson et al., 2003, 2004; Greve and Krause, 2018; Hölttä-Otto et al., 2012; Jiao et al., 2007; Jose and Tollenaere, 2005; Otto et al., 2016; Parslov and Mortensen, 2015; Pirmoradi et al., 2014; Salvador, 2007; Simpson, 2004). These reviews give a comprehensive overview of past research within this field, including more than 1.200 unique references, and show that little research has focused specifically on how to help designers and management decide on the exact number of product architectures for a full product portfolio.

The development and findings of the DNA method were achieved through an action research approach (Karlsson, 2009: 7), in which the researcher takes an active role in applying a theoretical method in a company. The outcome is not only a contribution to research-based knowledge, but also action knowledge aimed at practitioners. Truly understanding the processes in a company in which the DNA method must operate is crucial to making the method operational. Therefore, the action research approach is suitable for the present study.

This action research study is centred around workshops (biweekly on average) with four senior R&D (Research & Development) engineers (two mechanical, one electrical and one software engineer), one senior manufacturing engineer, a product manager and a managing director who has past experience in the R&D department. In addition, engineers from an engineering consulting company with skills in mechanical and electrical engineering were also involved. The workshops focused on the product architectures and how to divide the products into a feasible number of modules with robust interfaces between them. Individual work was done between the workshops by each engineer to conceptualise some of the architectural ideas from one workshop to the next. The knowledge gained from the workshops was documented by keeping a logbook over six months, during which time the project was ongoing, and by formalising all ideas on posters.

With regard to the company's data, we had access to all CAD (Computer-Aided Design) drawings. Some of the product documentation consisted of 2D drawings, and some were 3D. All sales data, which showed the distribution of sold product variants, were available going back three years. Production data was also available, but the registered times for operations had not been updated for years, so the quality of this data was questionable.

State of the art

We have divided the state of the art into five categories: function-based models, graphical methods, matrix-based models, mathematical models and platform-based methods. The referenced work is not a full list of all the scholarly work done in this field. However, based on the abovementioned literature reviews, the authors assessed that the referenced work does represent the state of the art regarding the design of modular product architectures.

Function-based models

Function-based models seek to map functional structures onto physical components and modules to describe product architectures. Function-based models include e.g. function diagrams (Ulrich and Eppinger, 2012) and functions-and-means trees (Andreasen, 1980).

Function-based models are also used to locate module candidates, as was done with the family function structures described by Zamirowski and Otto (1999). In addition, Erixon et al. (1996) presented 12 module drivers and used them in a Module Indication Matrix (MIM) to score product functions related to the drivers and thereby locate module candidates. The module drivers and the MIM are part of an extensive framework called Modular Function Deployment (MFD). The MFD method involves five steps, including clarifying the product design specifications through a QFD (Quality Function Deployment) matrix. The MFD method has been used with success for many single-family products. An example of this can be seen in Börjesson (2014).

Graphical models

Bruun and Mortensen (2012) developed the Interface Diagram, which builds on the Generic Organ Diagram (Harlou, 2006). The Interface Diagram is a tool used to capture the structural characteristics of a system by mapping the interfaces. It also defines the interface type between components. By using the Interface Diagram, one can identify potential modules in a product family by locating areas that are highly connected. In general, the tool can be used to support the design of highly complex systems.

Matrix-based models

The matrix-based model most commonly used to support modular product design and the design of product architectures is the Design Structure Matrix (DSM), which was first introduced by Steward (1981). Using the DSM, one can get an overview of all the components of a complex product system and locate potential module candidates. Algorithms have been developed to cluster the matrix so that highly connected components are grouped together to form modules. IGTA (Idicula, Gutierrez And Thebeau) algorithms (Borjesson and Hölttä-Otto, 2014) have been used and developed by several scholars along with other types of clustering algorithms. However, genetic algorithms are slowly becoming the predominant approach for product family design (Simpson et al., 2012).

Eppinger et al. (1994) applied the DSM tool to industrial project planning and improvement. They used the DSM to map the sequence of project design tasks. Then, they generated alternative sequences that could eventually speed up the development process by streamlining task coordination between and within teams.

Mathematical models

With regard to determining the right number of architectures, scholars have previously created mathematical models that can help define what they call “platform extent” – that is, how many products are covered by a certain platform and how many platforms are needed to cover the market. De Weck (2006) attributed the first paper on this issue to Seepersad et al. (2000), who produced an extensive mathematical model using data from the domains of product architecture, engineering performance, manufacturing cost and capital investment. De Weck (2006) developed an enhanced mathematical model for the same purpose but also took into account the effects of customer valuation, market demand and competition.

Schuh et al. (2017) developed a method for the contextual design of a modular product platform that adapts existing mathematical matching concepts. The method takes into account exogenous factors related to the product platform that cannot be controlled by the manufacturer while at the same time focusing on a company’s strategic targets.

Platform-based methods

Krause et al. (2014) introduced the Module Interface Graph (MIG), which is a graphical visualisation of the modular spatial arrangement, module boundaries, optional components and module interaction. This model is part of a larger framework, the PKT-approach, for the development of modular product families.

Mortensen et al. (2016) developed a comprehensive framework called AME – Architecture Mapping and Evaluation – that maps product and manufacturing architectures, links them to the market and calculates the financial impact of changing the architectures. The framework is largely based on a company's existing product portfolio, and its main benefit is grooming existing architectures rather than changing them.

Løkkegaard et al. (2018) created a modelling principle for Business Critical Design Rules (BDCR) that can be used to define a modularisation strategy. The BDCR focuses on three levels of a company's products: the portfolio level, architecture level and module level.

Otto et al. (2016) defined a set of 13 steps for developing a platform concept. The steps were identified by examining the product platform development processes used in several companies and by reviewing the literature.

Conclusion

We did not find any methods in the literature that can explicitly help companies decide on the total number of product architectures. Most of the methods described above are only concerned with single architectures and do not look across product lines. In addition, many of them have a high level of abstraction. The mathematical methods regarding architecture extend require much data input. Often in smaller companies, this data is not available, or it can be hard to get the full truth from a mathematical model. Moreover, matrix-based methods are not concerned with the transformation of a product architecture but are better used to describe existing product structures.

The method developed and presented in this paper was tested alongside a new product generation project. Hence, the new product architectures had to incorporate new solutions that could help convince top management of the value of the project. Introducing new product solutions to the existing portfolio while developing product architectures is not part of the scope of the existing methods presented above.

DNA method (Deciding the Number of Architectures)

The research presented in this paper deals only with mature industries and companies with existing portfolios of products, as the method takes its starting point from existing product structures. Therefore, situations in which there is no existing product structures are out of the scope of this research. In addition, the product structures are expected to be based on some level of modularity (see definition of module and architecture in the introduction section). The DNA method really seeks to improve a company's modularity rather than creating it from scratch, as can be done with e.g. Erixon et al.'s (1996) method using module drives.

The DNA method fundamentally builds on the work done by Harlou (2006) and Mortesen et al. (2016) in combination with morphology methods (Ko and Kuo, 2010; Zwicky, 1967). It is a way of modelling and challenging the total numbers of architectures in a company. Figure 1 shows an overview of the DNA method. The method essentially encompasses four stages: market segmentation, mapping new generation with an existing architecture strategy, architecture changes and the new architecture strategy. It is recognised that the process is iterative and that one may jump back and forward through the four stages. We will now describe each stage in more detail.

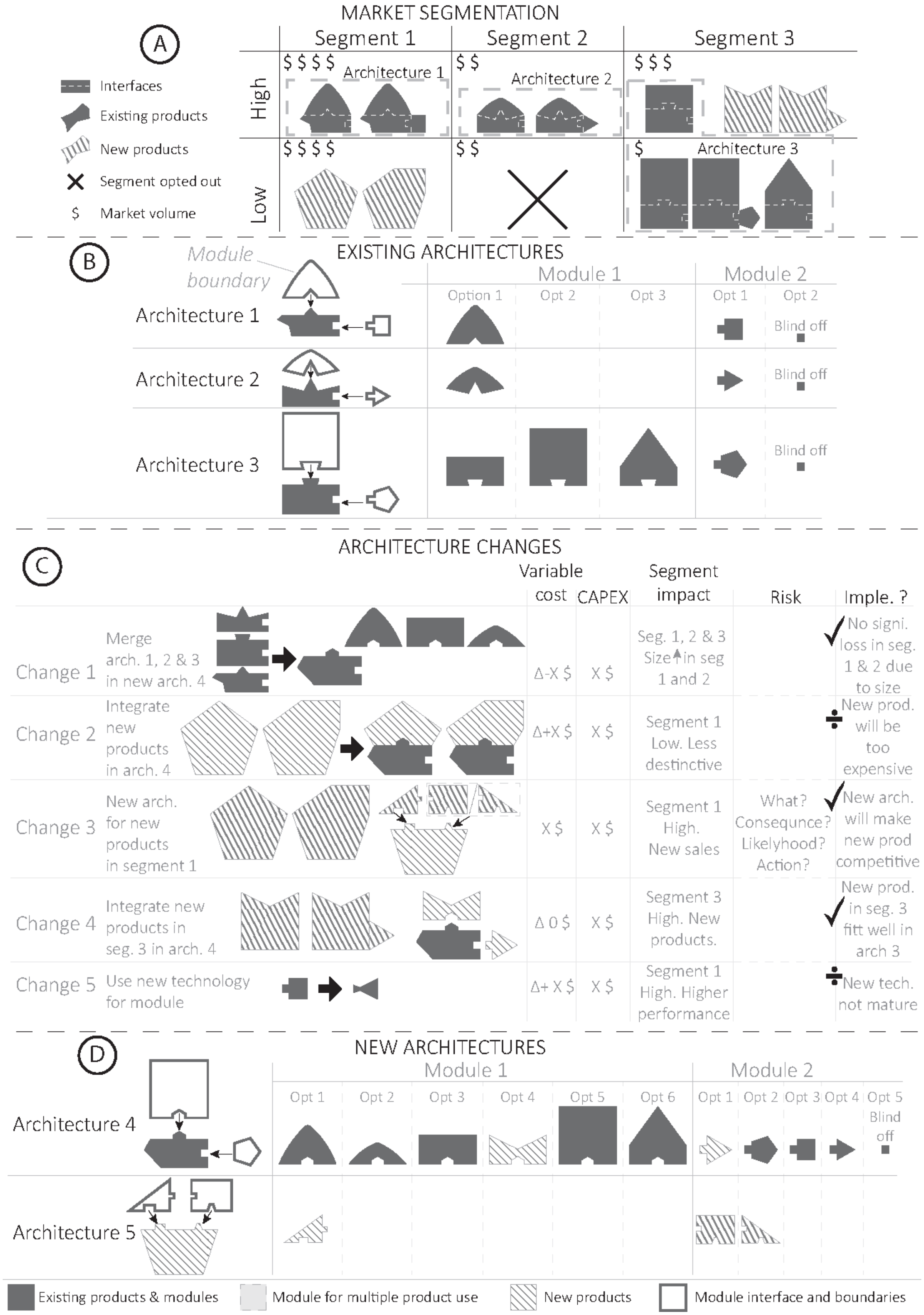

Figure 1: The four stages of the DNA method

Stage 1: Market segmentation

Many scholars, point out that segmentation analysis and planning is a key activity when developing product architectures, especially in the beginning of the process (de Weck, 2006; Meyer and Lehnerd, 1997; Mortensen et al., 2016; Otto et al., 2016). Therefore the first stage of the DNA method is concerned with analysing the market segments and competition (Figure 1-A). We used a segmentation grid (Meyer and Lehnerd, 1997) to map the size of each segment together with both new and existing products that are sold within the segment. This gives a clear overview of which products should be offered and which markets have opted out. The architecture to which each product belongs is also mapped.

This stage is the foundation for the new architectures, and it helps set the portfolio of products into context for the development team. In turn, this helps ensure that product designs are made to accommodate certain market demands at the right performance degree and cost.

Stage 2: Existing architectures

In this stage, one should map the existing architectures that represent the company's product offerings (Figure 1-B). Creating a clear mapping of the as-is situation is crucial to understanding what areas for improvement exist (Harlou, 2006; Mortensen et al., 2016). The architecture description should clearly show the functional modules and the interfaces between modules. We used a component-based approach to map the product architectures (Otto et al., 2016) in the case company. With less product knowledge, a function-based approach may be a better starting point and could be slowly developed to be more component-based. New product offerings would not belong to any architecture in this stage.

Stage 3: Architecture changes

When a baseline is made by mapping all existing architectures, the next stage can look into architectural changes (Figure 1-C) while thoroughly evaluating the trade-offs (Pirmoradi et al., 2014), from reducing complexity/costs to increasing product performance or competitiveness. This stage requires engagement from the most experienced R&D engineers, persons that have detailed knowledge of the market and experienced manufacturing engineers, as well as sign-off from top management (Otto et al., 2016; Sanchez, 2013).

Inspired by Mortensen (2016) and Løkkegaard (2018), each change should be evaluated at least with respect to the variable costs, CAPEX (capital expenditure) and segments impact, and each change should also undergo risk-assessment. Once this is done, it is a top management decision as to which of the changes should be executed, or at least the decision must be fully backed by top management (Sanchez, 2013). In the example in Figure 1-C, the decision is made to implement three of the changes and dismiss the last two. Implementing the three changes will: reduce the total number of architectures to two, integrate two new products into the merged architecture and create a new architecture for the last two new products. This brings us to the final stage in the DNA method of mapping the new architectures.

Stage 4: New architectures

This stage focus on mapping the new architectures (see Figure 1-D) based on the approved changes from Stage 3. a morphology-like approach (Ko and Kuo, 2010) to create alternative architectures by combining the change suggestions from Stage 3 in different ways may help in understanding how changes match and impact each other.

When the new architectures have been determined, the interfaces are locked so that the concurrent development of modules can take place when developing the new product generation (Prasad, 1996, 1997; Sanchez, 2013). The new architectures should be mapped clearly, stating which products are leveraged from which architectures. The new architectures should also be consolidated with the market overview in Stage 1 to ensure that no unintended gaps are left in the market. When the new architectures are in place, a roadmap of the implementation should be made (Albright, 2002). Reasons not to launch everything from the start could include a low technology maturity level (illustrated in Figure 1-C), the time needed to set up new production facilities to facilitate the changes, old products that must be phased out, etc.

Application of the DNA method

This method has been tested on a Danish SME (Small and Medium-sized Enterprise) that both develops and manufactures products. The company produces ovens for professional applications and needed a method for describing new architectures in designing their next product generation. The company was struggling with variable costs that were too high, as well as slow and inefficient product development. Therefore, both the internal complexity and variable costs of the products needed to be reduced. At the same time, the company aimed to broaden its portfolio to compete against the market leader, which had sales figures that were about 20 times higher.

The case company had recently changed its structures so that all strategic products decisions was driven by a product management function. This included decisions such as market offerings, interface freeze, product performance steps and target cost prices. Along with a new product architecture role and a voice-of-the customer role, the managing director took an active role in product management, which gave this team the necessary organisational power. This meant that almost all product development tasks were controlled by this function. By doing this, the company gained what Sanchez (2013) defines as a higher level of modularity maturity. A level higher than four is an important prerequisite for benefiting from the full potential of the DNA method. Given the reorganisation and its communication of the modularisation goals as a clear part of the company strategy, the case company was estimated to have achieved between levels 5-7 out of seven.

We will now briefly cover the findings from each stage of the DNA method conducted in the case company.

Stage 1: Market segmentation

Figure 2 shows a market segmentation grid for the case company. Both existing and new products, launched to better meet market demands, are mapped. The new products were expected to build on new architectures, as they would be fundamentally different from the existing products in terms of both function and cost-point. Markets that were opted out of are marked with a cross.

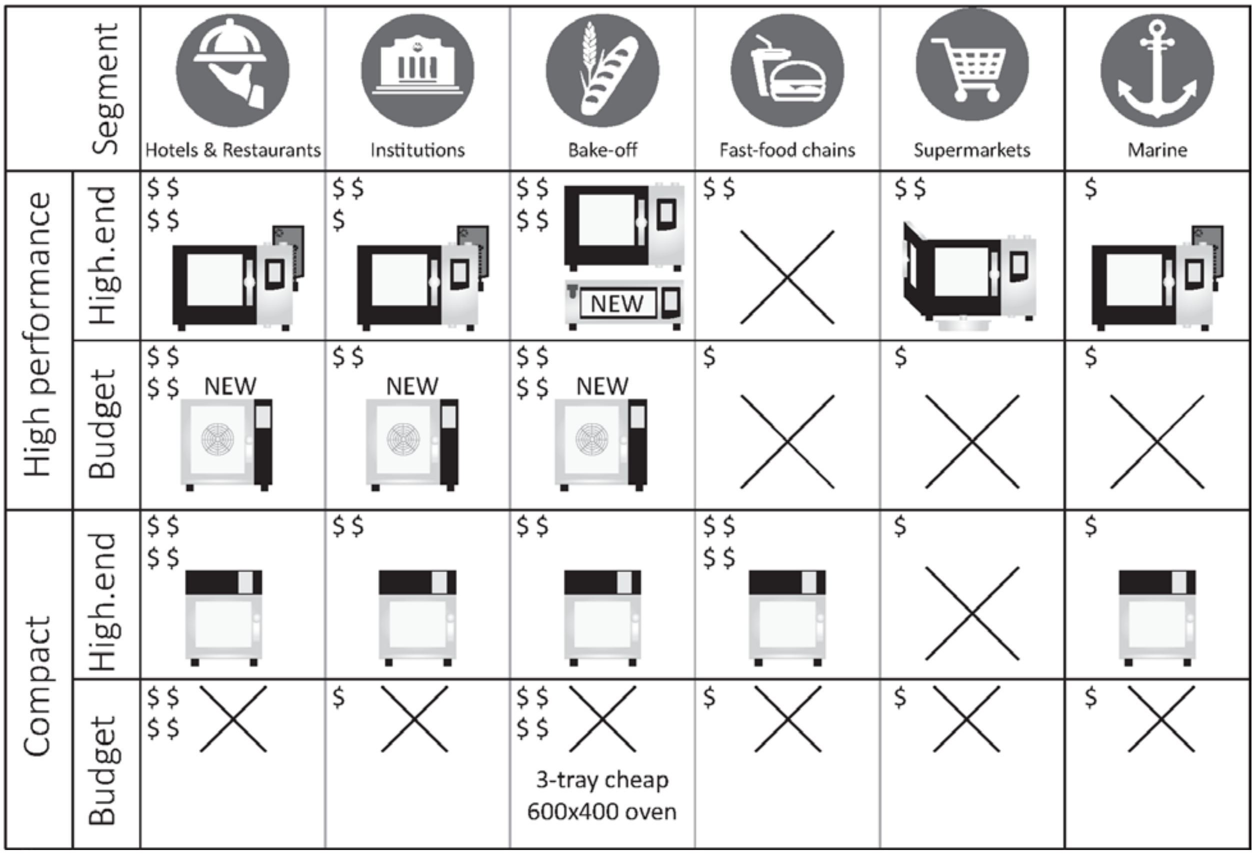

Figure 2: Market segmentation grid for case company

The market analysis indicated that it would be worth considering the introduction of two new architectures to launch new solutions on top of the existing portfolio. In addition, it was found that some of the existing products were too similar and were targeting the same customer requirements (small steps in load capacity), meaning that both a grooming and broadening of the portfolio was required.

Stage 2: Existing architectures

A section of the case company's next product generation using the existing product architectures is shown in Figure 3. Two of the new segments that the company wanted to enter could not be leveraged with the existing architectures. This naturally introduced the need to radically change the existing architectures or introduce new architectures (in this case, two new product lines as well).

If the existing architectures were not changed, the company would end up with so many different architectures that fixed costs would probably continue to eat up all the EBIT (Earnings Before Interest and Taxes), and lead time for new products would still be too long. This issue was clear when mapping the next generation of products only through the existing architectures.

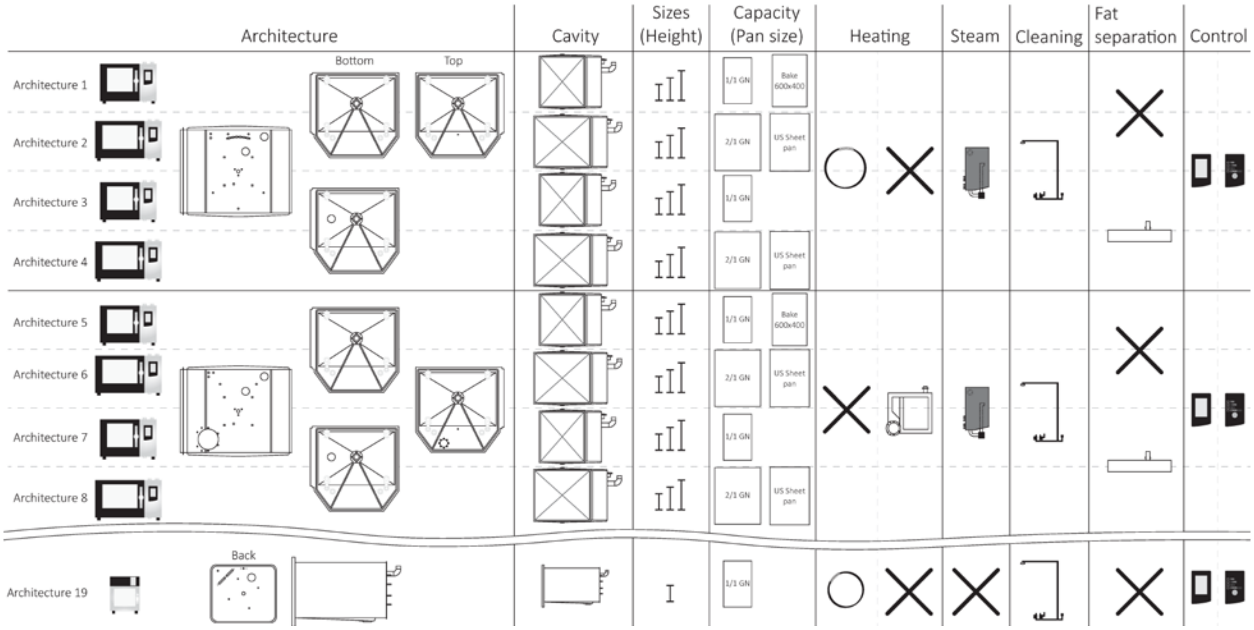

Figure 3: Section of the case company existing architecture

Stage 3: Architecture changes

Figure 4 shows a section of the architectural change suggestions that were made in the case company. The work done to produce this overview stretched over approximately six months and was facilitated by the knowledge of experienced internal product developers and external consultants. In the case company, the architectural changes fell into four categories:

- Architectural changes related to new production methods with a significant impact on the architecture design
- Changes reducing internal complexity by creating smarter interfaces that could facilitate more variance on the same architecture
- New architectures aimed at broadening the product portfolio into the new markets
- Technology improvements that required interface changes

Modules impact	Architectural change	Segments	Impact on architectures	CAPEX	Variable cost	Risk	Implement?
Cavity Fat sep. Drain	Make drain larger and direct fat draining through drain whole. May require 2 variants of the drain box		Removes need for special Fat cavity Potentially consolidates 4*4 architectures into 4*2	Deep draw tool: $$ *Smarter interface to reduce complexity*	0	Implies costs on ovens without fat separation Consequence: approx $ less annual margin with $ or. oven without ✓	Does not inflict significant cost on products without fat-sep.
Heating Cavity	Design new gas heat exchanger with inlet and outlet on one flange	All High performance	Removes cavity top plate dependency *Technology improvement requiering interface change*	0	+ $	New desing too high direct costs Consequence: less cempetitive on price or less margin Likelihood: Medium ✓	Will prob. be a little more exp. but opens possibility for doing gas and electric on same cavity
Cavity Cavity Front	Design Cavity to be robot welded *New production method with significant impact on the architecture*	All High performance	Cavity design needs to be changed radically to acommodate robot welding	Robot welding: $$$ Tooling: $$$	- $$	Set-up difficulties Consequence: Higher implemtation costs Likelihood: Medium (happened in another case in the company) ✓	If business case is approved by owners
MadeBy	New line of ovens Scaling of compact principle. Cheaper than high-en Only 1/1 GN	New product line *New architecture aimed at new markets*	- Cost efficient - Extra complexity of new architecture	Welding tool and production line: $$$		Unable to reach cost target Consequence: No added comp. or EBIT Likelihood: Medium ÷ ✓	Implement in near future

Figure 4: Section of case company architectural change suggestions

Stage 4: New architecture

Finally, a mapping of the new architectures is shown in Figure 5. The new architectures resulted from implementing the architecture changes from above that were judged feasible given the existing architectures.

Two of the architectural change suggestions accounted for the biggest reduction in the total number of architectures – namely, the suggestion to mount the two different heating module variants (electric and gas) on the same interface and the suggestion to integrate the interface to the fat separation system in the drain. In total, these two changes consolidated what would have been 16 architectures into four architectures. Furthermore, two new architectures were introduced to move into new markets. One existing architecture was left almost unchanged due to strict agreements with a large customer.

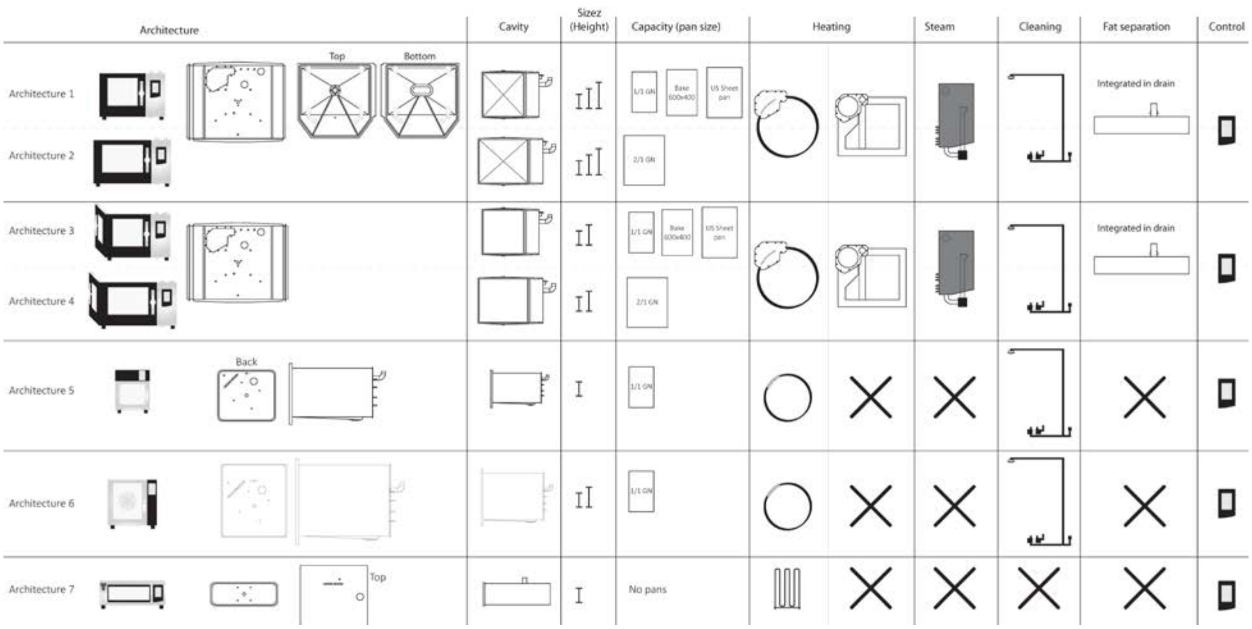

Figure 5: Case company new architectures

Not all architectural changes were to be made at product generation launch. A new product line was to be developed after launch because resources were insufficient. In addition, a new technology for producing steam was found to be too immature to implement at this stage and was planned for further consideration in 2-3 years' time. However, the architectures were prepared for the new technology.

Results of application of the DNA method

The DNA method helped the case company consolidate its architectures and reduce the total number of architectures to seven from what would have been 19 with the existing architecture design. At the same time, the architectures were prepared for future innovations.

In addition, by using the DNA method, a shortage was identified in the portfolio: a range of lower-cost ovens. It was found that by scaling a compact oven design while making some smaller design changes, a new architecture could be made to serve this market segment.

The new product portfolio was more efficient from a the perspectives of the customer-order decoupling point, product development and costs due to a higher commonality level (Pirmoradi et al., 2014). This was all due to a better architecture layout. At the same time, the new portfolio covered the market better than it had previously.

In this case, the answer to the optimal total number of architectures was fewer than existed. This was due to the fact that the company had for many years been developing single products without directly considering the full range of products. Thus, the company ended up with solutions built on different architectures that could in fact have been built on the same architecture. However, if the company chose to reduce the total number of architectures even more, it is likely that excessively high costs would be imposed on the cheaper products, and performance could be lowered in the high-end products. The answer might not always be that there should be fewer architectures. However, mature companies in mature industries have a tendency to introduce excessive complexity over time, which reduces efficiency (Dekkers, 2006; Meyer and Lehnerd, 1997; Patel and Jayaram, 2014) or have overlapping performance steps of products that end up competing for the same customers (Mortensen et al., 2016).

Discussion

As mentioned previously, no methods that focus on developing multiple product architectures with multiple product lines currently exist. One way to deal with this could be to expand the existing methods. However, methods like the MFD (Erixon et al., 1996), DSM (Steward, 1981) or even graphical-based methods like the Interface Diagram (Bruun and Henrik, 2012) tend to lose the overview when they get too complex. One way to cope with this is to create visual representations of the architectures instead and to focus on the key interfaces.

The mathematical models presented in the state of the art section dealt to some extent with multiple product lines in the way that they looked into platform extend (de Weck, 2006; Seepersad et al., 2000). However, these methods analyse one architecture/platform at a time, meaning that one would have to make many iterations between each platform to make sure they do not overlap, making it a relatively resource-intensive task.

All in all, the DNA method presented in this paper, in a way, is an extension of the platform-based methods presented in the state of the art section, being a qualitative visual modelling approach to deciding the total number of architectures in a new generation project with multiple product lines. The existing methods focus on single product lines (Krause et al., 2014; Otto et al., 2016) or mainly address rationalization potentials (Løkkegaard et al., 2018; Mortensen et al., 2016). Our contribution does not cover the processes that need to be in place when developing modular architectures for new product generations, such as lean product development tools/frameworks, as presented in e.g. Prasad (2016; 2016).

The DNA method was tested on a SME manufacturing company in Denmark that was in the process of developing its next generation of products. According to the Annual Report of European SME's 2017/2018 (Muller et al., 2018), SMEs made up 56.8% of the value added in 2017 in Europe, with large enterprises making up the last 43.2%. With so many SMEs, manufacturing companies with similar organisational structures and challenges than the case company should be common, making the case relevant to others. In general, companies manufacturing mechanical or mechatronic products, as in this case, should be able to use the method. What is most important to achieve the full potential of the method is that the power to make product decisions throughout all functions in the company is present.

Regarding the test company, the researchers had unlimited access to all data, and without this access, it would simply not have been possible to use the method. Moreover, without the strong support of top management, it would not have been possible to carry out the changes that were suggested by the DNA method. Full control of a company's product portfolio is required when using the DNA method. Therefore, the method will be more difficult to use in very large global companies. Such companies often have distributed product development activities at different sites, and it would be difficult to bring all of these departments together to work as one. However, the method might be applicable to single-unit R&D centres, as described in e.g. (Mortensen et al., 2016).

One aspect of the DNA method that makes it less attractive is that when comparing it with the practice of not designing architectures before product development, it could require quite extensive work prior to the actual development process of a new product generation. All functional modules and interfaces should be frozen and described precisely, so that concurrent engineering can take place during the detailed product development process that follows (Prasad, 1999) and during the years that the developed product generation will exist. However, if the “ideal” number of architectures is found, the extra development resources required could be paid back over the following years (Cameron, 2011; Meyer and Lehnerd, 1997; Schuh et al., 2010).

Conclusion

The main contribution of this paper is the DNA method, which is used to decide on the total number of product architectures in a company developing a new product generation. The method includes four stages – 1) market segmentation, 2) synthesizing a new portfolio with existing architectures, 3) suggesting architectural changes, 4) creating new architectures – and is a way of modelling and challenging the total number of product architectures in a company. However, the progress of the method is not linear but iterative, as it jumps back and forth from one stage to another. Considerable knowledge of a company’s existing products, manufacturing capabilities and market requirements is required to use the method, as it suggests that a new architecture design of a full product program can be made by transforming and adding to existing architectures.

The DNA method has been used in a Danish manufacturing company to decide the total number of product architectures in a new product generation project. Using the method in the case company helped in gaining important insights into the different aspects, both internal and external, that influence how cost- and resource-efficient the company will need to be to deliver new products in the future and with a competitive edge in different market segments. The company found DNA method efficient in the way it clearly visualized challenges with the as-is architectures quantity and formulated ways to move to a more efficient number of architectures. The answer to an efficient modularity level was in this case fewer architectures, but at the same time, new products were introduced. As discussed earlier, the answer to the total number of architectures might not always be fewer. In well-established companies in mature industries though, there is a tendency for companies to have introduced too much variance with regard to the total number of architectures.

In the case company, the risks of each architecture change were assessed using a simple consequence and likelihood of happening assessment. We have not developed a proper method for risk assessment of decisions involving the total number of product architectures and have not been able to find any examples of this in the literature. Thus, further research is warranted to identify a risk assessment method with regard to developing a certain number of product architectures that represent a full product program. Moreover, research on the total number of architectures and actual product development performance could provide a better understanding of how the total number of architectures affect a company.

Funding

The author(s) received no financial support for the research, authorship and/or publication of this article.

Declaration of conflicting interests

The Authors declared that there is no conflict of interest.

References

Albright R (2002) How to use roadmapping for global platform products. *PDMA VISIONS*.

Andreasen MM (1980) *Machine design methods based on a systematic approach-Contribution to a design theory*. Lund University.

Bonvoisin J, Halstenberg F, Buchert T, et al. (2016) A systematic literature review on modular product design. *Journal of Engineering Design* 27(7): 488–514. DOI: 10.1080/09544828.2016.1166482.

Borjesson F and 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. DOI: 10.1007/s00163-013-0164-2.

Börjesson F (2014) Modular Function Deployment Applied to a Cordless Handheld Vacuum. In: *Advances in Product Family and Product Platform Design*. New York, NY: Springer New York, pp. 605–623. DOI: 10.1007/978-1-4614-7937-6_24.

Bruun HPL and Henrik N (2012) Visual product architecture modelling for structuring data in a PLM system. *IFIP AICT - Advances in Information and Communication technology* 388: 598–611. DOI: 10.1007/978-3-642-35758-9.

Cameron BG (2011) *Costing commonality : evaluating the impact of platform divergence on internal investment returns*. Massachusetts Institute of Technology.

Campagnolo D and Camuffo A (2010) The concept of modularity in management studies: A literature review. *International Journal of Management Reviews* 12(3): 259–283. DOI: 10.1111/j.1468-2370.2009.00260.x.

Dahmus JB, Gonzalez-Zugasti JP and Otto KN (2001) Modular product architecture. *Design Studies* 22(5): 409–424. DOI: 10.1016/S0142-694X(01)00004-7.

de Weck OL (2006) Determining Product Platform Extent. In: *Product Platform and Product Family Design*. New York, NY: Springer US, pp. 241–301. DOI: 10.1007/0-387-29197-0_12.

Dekkers R (2006) Engineering management and the Order Entry Point. *International Journal of Production Research* 44(18–19): 4011–4025. DOI: 10.1080/00207540600696328.

ElMaraghy H, Schuh G, Elmaraghy W, et al. (2013) Product variety management. *CIRP Annals - Manufacturing Technology* 62(2): 629–652. DOI: 10.1016/j.cirp.2013.05.007.

Eppinger SD, Whitney DE, Smith RP, et al. (1994) A model-based method for organizing tasks in product development. *Research in Engineering Design* 6(1): 1–13. DOI: 10.1007/BF01588087.

Erens F and Verhulst K (1997) Architectures for product families. *Computers in Industry* 33(2–3): 165–178. DOI: 10.1016/S0166-3615(97)00022-5.

Erixon G, von Yxkull A and Arnström A (1996) Modularity – the Basis for Product and Factory Reengineering. *CIRP Annals* 45(1): 1–6. DOI: 10.1016/S0007-8506(07)63005-4.

Gershenson JK, Prasad GJ and Zhang Y (2003) Product modularity: Definitions and benefits. *Journal of Engineering Design* 14(3): 295–313. DOI: 10.1080/0954482031000091068.

Gershenson JK, Prasad GJ and Zhang Y (2004) Product modularity: measures and design methods. *Journal of Engineering Design* 15(1): 33–51. DOI:

10.1080/0954482032000101731.
Greve E and Krause D (2018) An Assessment of Methods to Support the Design of Future Robust Modular Architectures. *International Design Conference - Design 2018*: 335–346. DOI: 10.21278/idc.2018.0249.
Hansen CL (2014) *On the identification of architectures for product programs*. Technical University of Denmark.
Harlou U (2006) *Developing product families based on architectures : contribution to a theory of product families*. Department of Mechanical Engineering, Technical University of Denmark.
Henderson RM and Clark KB (1990) Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Administrative Science Quarterly* 35(1): 9–30. DOI: 10.2307/2393549.
Hölttä-Otto K, Chiriac NA, Lysy D, et al. (2012) Comparative analysis of coupling modularity metrics. *Journal of Engineering Design* 23(10–11): 790–806. DOI: 10.1080/09544828.2012.701728.
Jiao J, Simpson TW and 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. DOI: 10.1007/s10845-007-0003-2.
Jose A and Tollenaere M (2005) Modular and platform methods for product family design: Literature analysis. *Journal of Intelligent Manufacturing* 16(3): 371–390. DOI: 10.1007/s10845-005-7030-7.
Karlsson C (2009) *Researching Operations Management*. Routledge.
Ko Y-T and Kuo P-H (2010) Modeling Concurrent Design Method for Product Variety. *Concurrent Engineering* 18(3): 207–217. DOI: 10.1177/1063293X10379762.
Krause D, Beckmann G, Eilmus S, et al. (2014) Integrated Development of Modular Product Families: A Methods Toolkit. In: *Advances in Product Family and Product Platform Design*. New York, NY: Springer New York, pp. 245–269. DOI: 10.1007/978-1-4614-7937-6_10.
Løkkegaard M, Mortensen NH and Hvam L (2018) Using business critical design rules to frame new architecture introduction in multi-architecture portfolios. *International Journal of Production Research* 56(24): 7313–7329. DOI: 10.1080/00207543.2018.1450531.
Meyer MH and Lehnerd AP (1997) *The Power of Product Platforms : Building Value and Cost Leadership*. Free Press.
Meyer MH and Utterback J (1993) The product family and the dynamics of core capability. *Sloan Management Review* 34(3): 29–47.
Mortensen NH, Hansen CL, Løkkegaard M, et al. (2016) Assessing the cost saving potential of shared product architectures. *Concurrent Engineering Research and Applications* 24(2): 153–163. DOI: 10.1177/1063293X15624133.
Muller P, Mattes A, Klitou D, et al. (2018) *Annual Report of European SMEs 2017-2018*. Luxembourg. DOI: 10.2873/248745.
Otto KN, Hölttä-Otto K, Simpson TW, et al. (2016) Global Views on Modular Design Research: Linking Alternative Methods to Support Modular Product Family Concept Development. *Journal of Mechanical Design* 138(7): 071101 1–16. DOI: 10.1115/1.4033654.
Parslov JF and Mortensen NH (2015) Interface definitions in literature: A reality check. *Concurrent Engineering Research and Applications* 23(3): 183–198. DOI: 10.1177/1063293X15580136.

Patel PC and Jayaram J (2014) The antecedents and consequences of product variety in new ventures: An empirical study. *Journal of Operations Management* 32(1–2): 34–50. DOI: 10.1016/j.jom.2013.07.002.

Pine B (1993) *Mass Customization - The New Frontier in Business Competition*. Harvard Business School Press.

Pirmoradi Z, Wang GG and Simpson TW (2014) A Review of Recent Literature in Product Family Design and Platform-Based Product Development. In: *Advances in Product Family and Product Platform Design*. New York, NY: Springer New York, pp. 1–46. DOI: 10.1007/978-1-4614-7937-6_1.

Prasad B (1996) *Concurrent Engineering Fundamentals, Volume I: Integrated Product and Process Organization*. Upper Saddle River: Prentice Hall PTR.

Prasad B (1997) *Concurrent Engineering Fundamentals, Volume II: Integrated Product Development.* Goodwin B (ed.). Upper Saddle River: Prentice Hall PTR. DOI: 10.13140/RG.2.1.4710.1527.

Prasad B (1999) Enabling principles of concurrency and simultaneity in concurrent engineering. *AI EDAM* 13(03): 185–204. DOI: 10.1017/S0890060499133055.

Prasad B (2016) Lean, Integrated & Connected Framework for developing Smart Products. In: Batalla JM, Mastorakis G, Mavromoustakis CX, et al. (eds) *A Handbook on 'Beyond the Internet of Things: Everything Interconnected'*. Draft-Fir. Springer-Verlag, pp. 1–25.

Prasad B (Brian) (2016) On mapping tasks during product development. *Concurrent Engineering* 24(2): 105–112. DOI: 10.1177/1063293X15625098.

Robertson D and Ulrich KT (1998) Planning for product platforms. *Sloan Management Review* 39(4): 19–31.

Salvador F (2007) Toward a Product System Modularity Construct: Literature Review and Reconceptualization. *IEEE Transactions on Engineering Management* 54(2): 219–240. DOI: 10.1109/TEM.2007.893996.

Sanchez R (2010) Creating Modular Platforms for Strategic Flexibility. *Design Management Review* 15(1): 58–67. DOI: 10.1111/j.1948-7169.2004.tb00151.x.

Sanchez R (2013) Building real modularity competence in automotive design, development, production, and after-service. *International Journal of Automotive Technology and Management* 13(3): 204–236. DOI: 10.1504/IJATM.2013.054918.

Schuh G, Arnoscht J and Rudolf S (2010) Integrated development of modular product platforms. In: *Picmet 2010: Technology Management for Global Economic Growth*, 2010, pp. 1–13. IEEE Computer Society.

Schuh G, Riesener M, Barg S, et al. (2017) Methodology for the contextual design of a modular product platform concept. In: *21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17*, 2017, pp. 1–10. DESIGN SOC.

Seepersad CC, Hernandez G and Allen J (2000) A quantitative approach to determining product platform extent. In: *ASME Advances in Design*, 2000, pp. 1–14. ASME.

Simpson TW (2004) Product platform design and customization: Status and promise. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM* 18(1): 3–20. DOI: 10.1017/S0890060404040028.

Simpson TW, Bobuk A, Slingerland LA, et al. (2012) From user requirements to commonality specifications: an integrated approach to product family design. *Research in Engineering Design* 23(2): 141–153. DOI: 10.1007/s00163-011-0119-4.

Steward D V. (1981) *Systems Analysis and Management: Structure, Strategy, and Design*.

Petrocelli Books, New York.
Ulrich KT and Eppinger SD (2012) *Product Design and Development*. McGraw-Hill/Irwin.
Wilson S and Perumal A (2009) *Waging War on Complexity Costs*. McGraw-Hill Education. DOI: 10.1038/nsmb.3375.
Zamirowski E and Otto KN (1999) Identifying product portfolio architecture modularity using function and variety heuristics. In: *11th International Conference on Design Theory and Methodology*, 1999, pp. 1–12. ASME.
Zwicky F (1967) The Morphological Approach to Discovery, Invention, Research and Construction. In: *New Methods of Thought and Procedure* (eds Fritz Zwicky and AG Wilson), Berlin, Heidelberg, 1967, pp. 273–297. Springer Berlin Heidelberg.