Standard Article

Concurrent Engineering: Research
and Applications
1–11

DOI: 10.1177/1063293X15624133
cer.sagepub.com
SAGE

Assessing the cost saving potential of shared product architectures

Niels Henrik Mortensen[1], **Christian Lindschou Hansen**[1], **Martin Løkkegaard**[1] **and Lars Hvam**[2]

Abstract

This article presents a method for calculating cost savings of shared architectures in industrial companies called Architecture Mapping and Evaluation. The main contribution is an operational method to evaluate the cost potential and evaluate the number of product architectures in an industrial company. Experiences from the case company show it is possible to reduce the number of architectures with 60% which leads to significant reduction in direct material and labor costs. This can be achieved without compromising the market offerings of products. Experiences from the case study indicate cost reductions between 0.5% and 2% of turnover. The main implication is that the method provides a quantitative basis for the discussion on whether or not to implement shared product architectures. This means a more fact-based approach is introduced.

Keywords

product architecture, manufacturing architecture, modularization, concurrent engineering

Introduction

Many industrial companies (developing, producing, and selling physical products) have developed the product portfolio sequentially product by product over a number of years. There can be many good reasons to continue doing so, for example, ability to develop specific products for specific markets and targeting specific low-cost needs and high-end needs. The consequences are, however, often that there exist a large portfolio of products, where there is very limited sharing between the product families, leading to increased complexity cost, several "inventing the wheel" projects and thereby increased time to market and profit for new products (Andreasen, 1980; Hansen, 2015; Harlou, 2006; Levandowsky et al., 2014). A warning signal is often that costs are increasing faster than turnover.

Companies typically have challenges such as the need to reduce cost, increase quality, reduce delivery time, and launch more new innovative products faster. One of the means to address this challenge that is often discussed in both academia and industry is application of modular architectures (Ericsson and Erixon, 1999; Guðlaugsson et al., 2014; Herrmann et al., 2004). The basic idea of modular product architectures is to build up product lines based on a limited well-defined module having well-defined performance steps with clear definition of interfaces (Ericsson and Erixon, 1999). This should lead to the reduction in the number of components, cost reduction in general, and more focused effort on key modules leading to more cost-effective products. The rationalization benefits may be utilized to develop more new innovative products.

In principle, everyone, from board of directors, board of management, and down in an industrial organization, agree on this. But, in practice, there are many uncertainties and many opinions. In our research, we have often come across viewpoints such as "we have exactly the products that we need," "all products are profitable and needed for our customers and markets," "we are already modular," and "if we increase the level of modularity, we will compromise key customer

---

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

Corresponding author:
Niels Henrik Mortensen, Section of Engineering Design and Product Development, Department of Mechanical Engineering, Technical University of Denmark, Building 426, DK-2800 Kongens Lyngby, Denmark.
Email: nhmo@mek.dtu.dk

requirements and increase cost." All the above viewpoints can often not be proved right or wrong.

Organizational wise, this is a very sensitive topic. Asking the question "could we do better with modular architectures." From research and development (R&D), the reaction is often that they are accused of not having done the perfect job. From sales, the viewpoint will often be that due to the competitive situation and so on, all products are required. In manufacturing, the reaction is very positive—but often more short-term initiatives are in focus such as day-to-day process improvements.

The basic question asked in this article is as follows: how to find out what the financial potential is of shared modular architectures? The target audience is board of management. The intention has been to identify a quantitative method in such a way that discussions are based on facts and not opinions of individuals.

Concerning the link between the number of architectures and concurrent engineering (CE), it is the assumption that when the number of architectures in product and manufacturing is reduced, it will lead to increased efficiency and increased possibilities of achieving concurrent development of product and manufacturing.

The structure of the article is as follows: in section "Research approach," the research method is explained, and section "What is a product architecture and which evaluation parameters to include?" will go through the benefit dimensions and the reasons for including them. Section "State of the art" describes state-of-the-art literature. In section "AME method," the method for calculation of the benefits of shared architectures is presented and section "Application of the AME method" contains experience from application of the method in a large-scale organization. Section "Discussion" contains the conclusion.

Research approach

The suggested method for assessing the potential of shared product architectures has been developed by taking from both the existing literature and some experiences of practitioners. More specifically, the method is based on classical systems' thinking (Andreasen, 1980; Skyttner, 2005), theory of technical systems (Hubka, 1973), and Product Family Master Plan (PFMP) (Harlou, 2006). The work is carried out by three MSc projects and two PhD projects (Bruun, 2015; Hansen, 2015) at the Technical University of Denmark, Department of Mechanical Engineering, Section of Engineering Design and Product Development. The basic assumption is that the competitiveness of a product program cannot be evaluated in itself—only when mapped relative to the market and the internal functions of the company, conclusions can be made. In this study, three aspects are considered: market, product, and manufacturing. The links between these three aspects are the foundations for evaluating the potential of shared architectures. Most approaches in the literature on shared product architectures are concerned with the product aspects in terms of, for example, shared parts. Even though this can be of relevance, it is not sufficient.

A method named Architecture Mapping and Evaluation (AME) method is proposed. The method has been tested in a global company that has approximately 75,000 commercial variants in the market. The company has divided the product portfolio into six product lines. The AME has been tested on all six product lines. This means that in total, six global data sets have been collected and evaluations have been carried out. The main aim of this case study is to test the suggested operational method and receive feedback from the managers in the company.

With regard to internal validity, the research team has full access to detailed data from the company. In order to gather accurate qualitative data, un- and semi-structured interviews are performed with the "key" informants. The research group had semi-structured interviews with the managers, involved in this project, in order to assess the results and receive feedback. The received feedback is valuable for the verification of the results from the analysis and for assessing the proposed method. The studies have been carried out over a time span of 2 years from 2012 to 2014. The next section will discuss the meaning of product architecture and what evaluation parameters to include.

What is a product architecture and which evaluation parameters to include?

Like most phenomena in engineering design, there does not exist a common and agreed way of defining architecture. In this article, a distinction between product structure and product architecture is made (Hansen et al., 2012). Product structure means the way a single product is built up from systems and components. Product architecture means the way a product family or portfolio of products is built up. Traditionally, companies have good control of product structure in, for example, computer-aided design (CAD), enterprise resource planning (ERP), and product data management (PDM) systems. Product architecture is normally very weakly taken care of. Traditionally, responsibility for product structure is well defined, but responsibility of product architecture is ill defined. It is the main assumption in this article that the number of product architectures is a very important fundamental aspect, and that top management and other key persons should consider very carefully. Having too many architectures will lead to high complexity cost and long time to market for product development. The implication of too few architectures can be too high cost for product in, for example, the lower performance areas of the portfolio or simply that the company cannot serve the variety of needs among customers.

In this work, product architecture has the following characteristics:

- Shared core interfaces.
- Core modules/systems exist in balanced performance steps.
- The architecture is explicitly prepared for derivative products and related properties in terms of cost and performances are known.

The above phenomena will briefly be explained.

Shared core interfaces

Only a small fraction of interfaces play an important role, but a few are extremely important for, for example, quality and time to market. An example of a core interface of a truck might be the interface between the cab and the rest of the chassis. If this interface is stable, the cab can be developed without changing the rest of the chassis. The whole product family can be upgraded in one step with one development project.

Core modules exist in well-balanced performance steps

An example of a core module could be the wash-group of a washing machine; some of the performance steps could be 6, 8, 10, and 12 kg. Balanced means that there the number of modules is consciously determined according to market needs and internal complexity within the company, for example, production, service, stock level and development capability. One "ideal" way of balanced performance thinking is "one need—one solution."

Architecture is prepared for future launches

An example could be boggies of a truck. There might exist 21 and 30 ton, but modules are prepared for a 26-ton variant with adding only a few new parts. Another consequence of this is that interfaces have to be stable over time. This is one of the weak parts of architecture work in most companies that we have studied (Bruun et al., 2014; Hvam et al., 2008).

In this article, a product architecture is considered shared when more than 90% of the core interfaces are shared. Then one can ask what a core interface is. This is pragmatic defined among senior market, product, and manufacturing persons. For a car, an example of a core interface is between engine and transmission. For a drilling tool, a core interface will exist between battery and chassis. The basic assumption is that the number of architectures is driving complexity cost; it is driving CAPital Expenditures (CAPEX) in manufacturing and it is often constraining the ability to launch new products and product variants. Then why put the requirement on 90% sharing of core interfaces? This is a pragmatic decision, but due to the size of the test company, the criteria had to be explicitly defined in such a way that each division of the case company could not have individual perceptions.

Having clarified the meaning of architecture, the next question is what evaluation parameters to include. There is in principle an infinite number of evaluation parameters that could be studied. In the literature, roughly two types of evaluations parameters are reported in the literature (Fixson, 2005; Hultink et al., 1997; Krause et al., 2013; Ulrich, 1995) from application of shared architectures. They can be divided into growth parameters and rationalization parameters. Examples of growth parameters could be time to market and ability to make new innovative products. Examples of rationalization parameters could be direct material cost and labor cost. In this article, it has been decided not to include growth parameters, not because it is irrelevant, but because it is difficult to obtain quantifiable data. There are often many opinions but very few facts. On the rationalization side, it has been decided to include data that are available in modern companies with modern information and technology (IT) systems, mainly ERP systems. Again, there are many possibilities, but included are four parameters: direct material cost, direct labor cost in manufacturing and CAPEX on tooling, and number of architectures. The basic assumption is that if benefits can be justified in these dimensions, the rest such as the growth parameters will be additional benefits.

State of the art

The review of the state-of-the-art includes a review of five different groups of supporting methods for the identification of shared architecture benefits for a product program including product lines. The five groups identified are function-based models, matrix-based models, CE, design for manufacture (DFM), and mathematical models.

Function-based models

Methods describing the development of modular product architectures often choose to start with the conscious mapping of functional structures into physical modules (Levandowsky et al., 2014). Functions can be represented in function-based models, for example, functions-and-means trees (Andreasen, 1980), or by schematics of the product including physical elements to a meaningful extent (Stone et al., 2000).

The understanding product functions can be used in different ways to identify possible modules. To improve the identification of modules and make sure that the modular architecture will serve its objectives (Fixson, 2005), define a set of module drivers. The module drivers can support the reasoning behind the module identification by elaborating the justification of the modules' existence, for example, "planned product changes" module, "process" module, "different specification" module, and "technology evolution" module. The module drivers are a part of a comprehensive framework called modular function deployment (MFD), which in analog to the quality function deployment (QFD) method provides support for the linking of relationship between the module drivers and technical solutions.

Matrix-based models

Another approach to identify modules is the application of design structure matrices (DSMs). This approach takes its point of departure in the decomposition of a product into parts and/or subsystems while identifying the relations (and possible future interfaces) among these (Gonzalez-Zugasti et al., 2000; Otto and Wood, 1998). By applying different algorithms and clustering techniques, it is possible to encapsulate functional "chunks" that have the potential of becoming physical modules, due to their functional interrelations. DSM techniques are the subject of many research initiatives and serve as the basis for an array of derived methodologies. An example of this is the multi-domain-matrix (Ulrich, 1995). Alternatively, other design tools focus more on the specific task of examining different functional flows with the aim of identifying modules (Otto and Wood, 1998; Pimmler and Eppinger, 1994). These methods are heuristically based.

Other more general methods focus on the identification of common features in the existing product program in order to point out the basis of the product architecture. By formulating the design task as a quantitative problem, which can be subject to optimization, this method is balancing inputs from requirements and product variants design with data models of performance and costs. By iteration, the optimal product variants are designed and evaluated through quantitative performance metrics.

CE

From the associated area of CE, one can also find research into the concurrent development of product and production architectures, with phrasings such as "methods supporting the development of product platforms." Nevertheless, interesting contributions are submitted within this area. Otto and Wood (1998) introduced a three-dimensional (3D) methodology superimposing the traditional domains of CE, by suggesting the linking of technology, architecture, and focus relations in the process, product, and supply chain domains. Olesen (1992) proposed an important step of operationalization of this 3D-CE approach by developing a multi-dimensional framework that enables comprehensive assessment of alternative product architectures.

The concept of architecture for product family (APF) is introduced as a conceptual structure, proposing logics for the generation of product families (Hultink et al., 1997). The generic product structure (GPS) is then proposed as the platform for tailoring products to individual customer needs. In Andreasen and Olesen (1990), another systematic method for concurrent development of product families is presented, by combining QFD-based methods with quantified DSM techniques and morphology analysis to visualize concepts.

DFM

Original contributions from Olesen (1992) proposed a framework for the concurrent development of manufacturing supported by the theory of dispositions (Andreasen and Olesen, 1990). This is done by proposing a set of models aligning the product design and the product life system phase of manufacturing to create a fit. However, for the case with Design for Assembly (DFA) and DFM methodologies, the main focus is single product development. Herrmann et al. (2004) comment that an extension of the DFM tools to comprise multi-product development will hold the key to achievement of competitiveness.

Mathematical models

Some researchers have undergone the task of developing methods based on mathematical models. Some methods are based on measures of modularity, which act as subjects of optimization using different techniques (Hultink et al., 1997). Others seek to integrate product platform, manufacturing process, and supply chain decisions through the application of mathematical models, thus extending the concept of the generic bills of materials (GBOM) by quantifying relations between decisions from the different domains.

Conclusion

It is evident that the contributions mentioned above can play a role in the identification of program architectures. Situated in this cross-functional research field, it is clear how research centered within either the product or production domain tends to leave out important aspects of the adjacent fields, and considering the identification of program architecture this is a deficiency considering the contributions listed above. Extensive research is also found within the reengineering of business processes and different means of optimization of operations, but these areas exclude necessary details within the field of architectures. They are simply not concrete enough, or deal with sub-optimization of operations and processes leaving out the product domain. The methods do not explain how the modeling and evaluation is carried out for very large product programs with, for example, 70,000 commercial products and 300,000 parts. There is very little support for supporting the very fundamental question: "how many product architectures are right for our company?"

AME method

This section presents a seven-step method to evaluate the benefits of shared modular architectures. These seven steps are as follows:

Step 1: map the market globally and main required properties;
Step 2: map cost/performance for core module areas;
Step 3: map each as-is product architectures;
Step 4: map each as-is manufacturing architectures;
Step 5: identify to-be product architectures and manufacturing architectures;
Step 6: map cycle plan;
Step 7: calculate financial impact.

In the following, each step will be explained.

Step 1: map the market globally and main required properties

In this step, the market and required properties are mapped according to Fixson (2005), Levandowsky et al. (2014), and Meyer and Lehnerd (1997). It means that the market for a product line is grouped into approximately 4–12 categories. There are normally two axes in the mapping (segment and performance levels such as high end and basic), see Figure 1.

For a pump manufacturer, it might segment wise be geographical area (e.g. North America, Europe, Asia pacific) and performance wise, media pressure (up to 2 bar and above). For each group, key properties, for example, energy efficiency and lifting height, are identified. The product line properties are then mapped and competitor product (best in class) is mapped. The result is a number of "spider charts" as shown in Figure 1. Finally trend indicators are identified. It means in

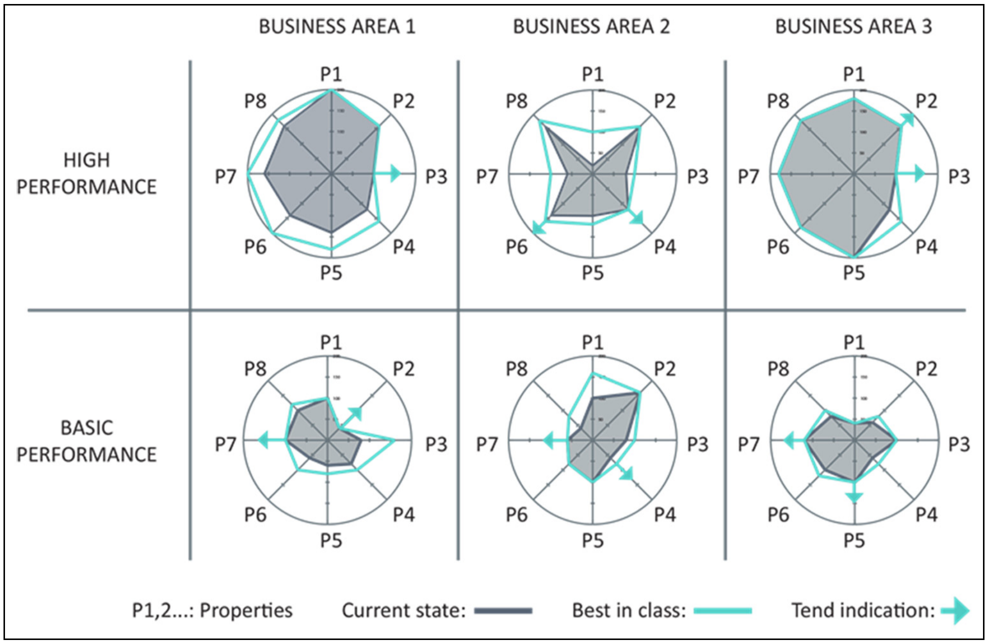


Figure 1. Mapping of market segments and required properties.

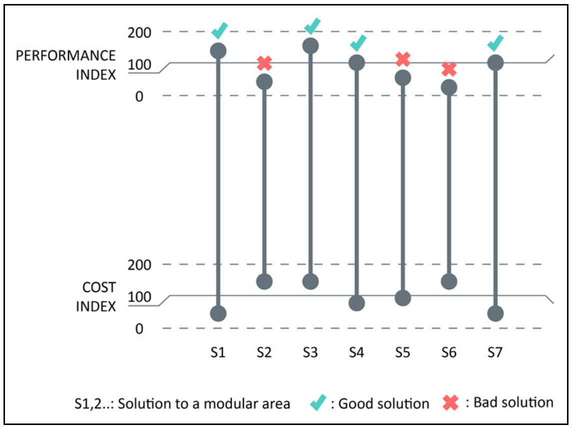


Figure 2. Description of cost/performance for a key module is mapped. Each vertical line represents a module area. The lower dot is the cost and the upper dot is a certain performance of a property.

which directions do the company expect that requirements will change. Concerning the energy efficiency, it is very likely to be reduced in next generation of pumps. Trend indicators are utilized in Step 5. It is very important that the architectures are prepared to deliver the right properties. One CEO explained, "It is important to be prepared for the next war and not the previous one."

Step 2: map cost/performance for core module areas

In this step, the products in a product line are pragmatically divided into a number of module areas that are the carrier of key properties for a product. For a pump, it might be motor, hydraulics, controls, and so on. Then a few key module areas are identified, which are the carrier of major cost and major properties (Huang et al., 2005; Otto and Wood, 1998). The purpose of this step is to map key module areas in a direct material cost and a relevant performance dimension. For a pump manufacturer, it could be the motor and the controls. Often, a few module areas cover the majority of the cost and performance in a product. Then each module area variant is mapped in a cost performance diagram as shown in Figure 2.

This overview is quite important in the method (Guo and Gershenson, 2007). Often, there will be different module areas with very different cost levels but is delivering same performance. In other cases, there will be module areas that have low performance and high cost. In other words, the module area is expensive and can do very little. What should be the immediate reaction to such a module area "is there any good reason for having this module area in the product line." What has been observed in the case project is that there is up to a factor 3 in direct cost differences between module areas that have similar performance. So there are significant direct material cost reduction possibilities by consequently utilizing the most cost-effective modules.

Step 3: map each as-is product architecture

In this step, the number of architectures within a certain product line is identified (Olesen, 1992). The process is that key interfaces are identified. This number has in this research project been 10 or below. Examples on key interfaces in a pump might be between housing and impeller. The interfaces play a crucial role in development for an industrial company. If and only if interfaces are shared, the modules can be shared. Figure 3 shows an example of how the number of architectures is identified. There is a very important link between Step 1 and Step 3. Reasoning from Step 1 to Step 3 should bring forward the question: how many architectures are right for our company in order to deliver good products in the different segments? In the case company, there has been a clear tendency that the companies have more product architectures than can be justified from a market point of view.

Step 4: map each as-is manufacturing architecture

The main purpose of this step is to identify differences in manufacturing properties, that is, labor cost (Andreasen and Olesen, 1990; Stone et al., 2000). Figure 4 shows an example where a product with different architectures is manufactured in different factories in Europe, United States, and China. What is compared are the differences in labor assembly time on subassembly lines and main assembly lines. In the case projects there has been a factor 2 deviation in labor time between the best and worst performing product architecture. This means that the product architecture plays a major role for efficiency in production. In principle, the productivity in the studied factories can be improved with a factor 2 by conscious selection of the best product architectures.

Step 5: identify to-be product architectures and manufacturing architectures

In this step, experienced persons from sales, product development, and manufacturing are taking a top–down look from a market point of view and identify how many architectures and module variants are needed in order to serve the market (Lindemann et al., 2009; Meyer and Lehnerd, 1997). This is really an expert judgment, where the most senior people in the organization have to be involved. In the case project, the reduction possibilities in terms of product

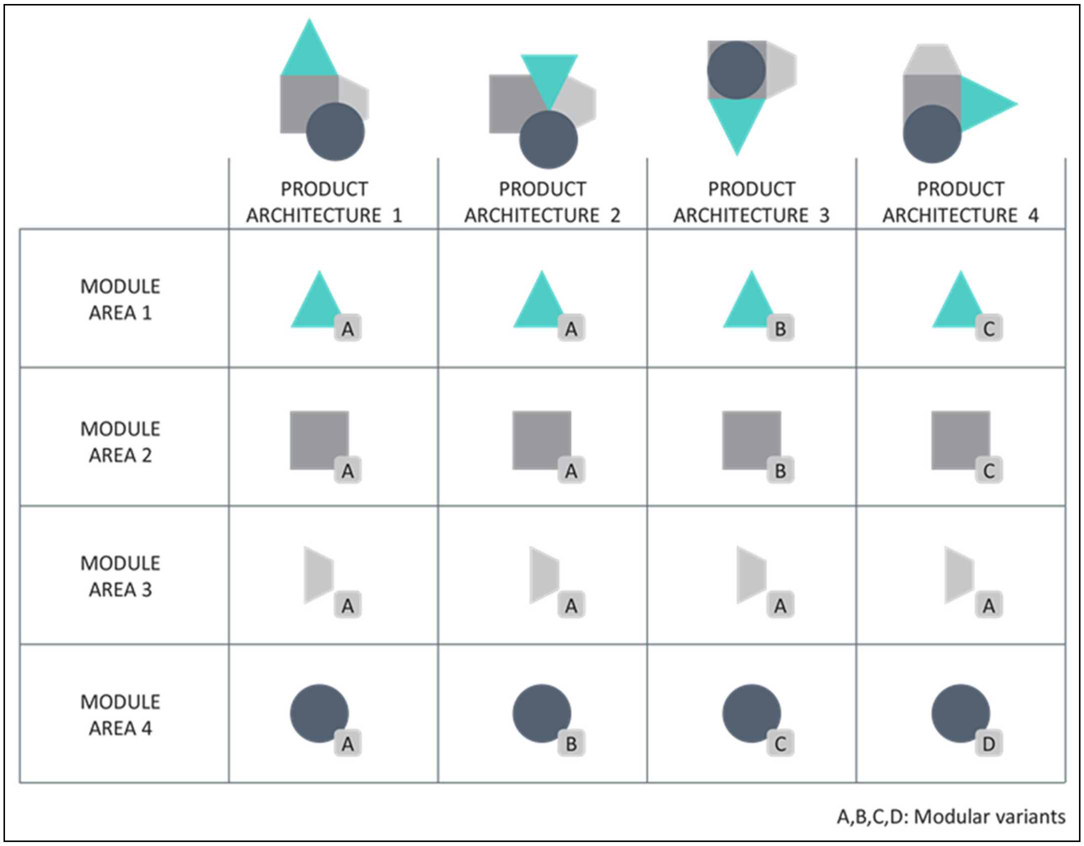


Figure 3. Mapping of current architectures.

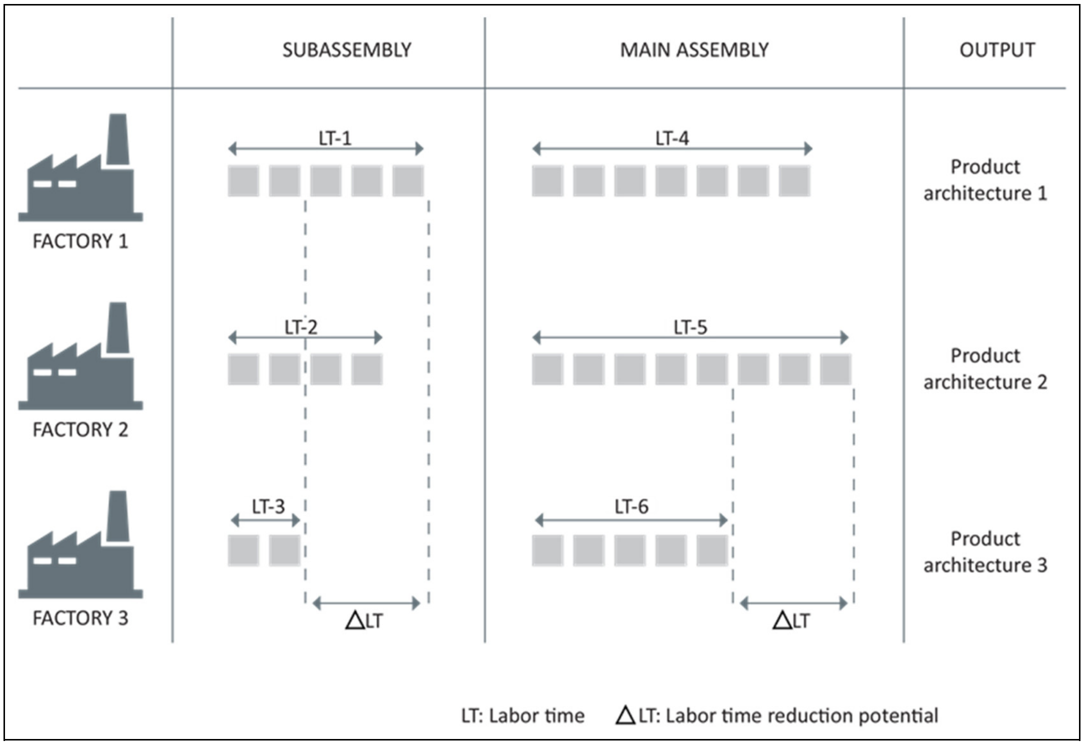


Figure 4. Manufacturing architectures.

architectures, manufacturing architectures, and module variants have been between 5% and 50%. In other words, the company is much more complex than needed.

Step 6: map cycle plan

The starting point in this step is a 5- or 10-year cycle plan, showing when products and product line are expected to be upgraded or relaunched (Krause et al., 2013; Pimmler and Eppinger, 1994). Next, phase out and phase in of architectures are added. It is further assumed that the best modules are consequently utilized across the product lines. Based on the reduction in product and manufacturing architectures, utilizing the most cost-effective module areas, it is possible to estimate direct material savings and direct labor savings.

Step 7: calculate financial impact

In this step, the benefits in terms of direct material cost, direct labor cost, and CAPEX avoidance are summed up (Du et al., 2001; Kester et al., 2013; MacDuffie, 2013). The results are three numbers explaining the financial potential of shared architectures. In the case company, this has been a very important step in order to put shared architectures on the top management agenda. One of the main advantages is that now such an initiative can be compared to other big initiatives such as automation, low-cost country sourcing, and manufacturing footprint location.

Application of the AME method

The method has been tested in a global business to consumer company. The case company has sales companies in 100 + countries and 35 factories in Europe, Asia, North America, and South America. There are six R&D centers that develop six product lines. The company has approximately 75,000 commercial product variants in the market and around 400,000 part numbers. The company has over a longer period been part of several mergers and acquisitions.

For several years, there had been a discussion in the board of management concerning the complexity of the product lines. It has among certain members been the assumption that it should be possible to serve the markets with fewer architectures and parts, but no definitive conclusions could be made. Therefore, the company wanted to test the AME method.

A team of three full-time persons (called the core team) for each product line was appointed. The core team consisted of a researcher, a senior R&D person, and a financial controller. This core team has ad hoc access to senior experts in sales/marketing, R&D, manufacturing, purchase, and financial control. In total, approximately 30 persons for each product line have been active in the work.

The AME work has been carried out during 20 weeks for each product line. The assessment work has been divided into three phases.

Phase 1

This included Steps 1, 2, 3, and 4. The main way of working has been interviews with key persons, site visits to key factories, and data extract from the ERP systems.

Phase 2

In Steps 5 and 6, three workshops with senior market, product, and manufacturing experts were carried out. The work was fundamentally anchored around the number of architectures. The main question asked was as follows: how many architectures do the company need in future? It is an illusion that there will be consensus concerning this. What happened in the workshops is that there were structured discussions and viewpoints were delivered from the experts. After the workshop, the core team made a conclusion concerning the needed number of product architectures, manufacturing architectures, and module area performance steps. This is a very crucial step—and much further detailed work has to be carried out later on in implementation. Table 1 shows a possible reduction in product architectures from 60 to 25. Perhaps, detailed studies will later show that, for example, 30 or 15 product architectures are better. This will, however, not change the main conclusion—that significant cost reductions are possible.

Phase 3

This is calculating (Step 7) the benefits in terms of direct material, direct labor, and CAPEX avoidance

Table 1. List of as-is architectures, to-be architectures, and financial impact.

Product line	As-is product architectures	To-be product architectures	Financial impact (% of turnover)
1	8	4	2.0
2	9	4	0.5
3	12	6	1.2
4	5	3	0.9
5	10	4	2.1
6	16	4	1.0
Total	60	25	

concerning tooling. The main inputs are the cycle plan, number of new architectures, and number of key module areas with "best of breed" cost/performance levels obtained in Step 3. This means that impact calculations are very conservative, that is, it is based on solutions and principles that are already available in the company today. The main results are summarized in Table 1. The work has lead to significant conclusions and discussions in the board of management.

Reduction in the number of product architectures. It is possible to reduce the number of product architectures significantly from 60 to 25 without comprising the number of commercial variants in the market. No one can for sure know whether this is completely true, but it seems that a significant reduction is possible. It has become clear to the board of management that the number of product architectures is strategic decision in the company that has to be anchored on senior vice president level. One vice president explained, "One architecture is very wrong—there will be bad cost/performance compromises. On the other hand 20 architectures is also wrong—this will lead to high complexity on and unfocused R&D effort."

Additional benefits of fewer architectures. It is the assumption that the benefits in Table 1 are only the top of the iceberg. There are additional savings in terms of reduced ware house cost, due to fewer module areas and part number. The efficiency in factories should increase due to fewer change-overs on the assembly lines. It should also be possible to introduce later customer order decoupling points, which should reduce delivery time. Furthermore, it should be possible to increase utilization level in factories, due to fewer parts, modules, and architectures. Furthermore, additional savings can be expected in purchase due to higher purchasing volume.

From an R&D perspective, fewer architectures means that the R&D effort on each architecture could be increased. This should again lead to increased quality, higher level of innovation, and reduction in time to market for new variants.

Product line design principles. During the work, it has been clear that some product lines are fundamentally wrongly designed. The engineering design approach has been wrong. It means that high-end products have been designed first and then the approach has been to "strip" them to reach mid- and low-end markets. The results have been that the costs for mid- and low-end products are too high. One R&D manager explained, "Stripping a Rolls Royce will not lead to a cost effective Polo car." The conclusion is that every module area design should in the future be based on scale up thinking rather than scale down thinking and part of one or more well-defined architectures.

Implementation. Two fundamental implementation alternatives are being considered. The traditional organization could drive implementation according to the approved cycle plans. The implementation time would then be approximately 7 years for all product lines. Another alternative is to establish a separate product and manufacturing architecture organization that has the full responsibility for all product lines and manufacturing. This would reduce implementation time but increase CAPEX. So far, no conclusions have been made. Another concern is the coordination between shared product architectures and increased automation in assembly. These two initiatives naturally have to be coordinated. It would be waste of resources to automate product architectures that will be phased out. Implementing shared architectures and then afterward increased atomization might take too long time.

Discussion

In the state-of-art literature on platforms and engineering design in general, it is often the assumption that concepts for the future product program have to be developed in order to evaluate cost reduction potentials. For practical reasons, this will not be possible in large global companies, so another approach is necessary to evaluate financial impact of shared architectures. The main contribution in this article is a top–down reasoning approach. This means reasoning from what is required in the market and relating this to the number of as-is product architectures. Hereby, the mismatch between market requirement and the current number of product architectures should be recognized and the ideal future number of architectures is identified. Compared to a real conceptualization project, the AME method will not provide financial benefits with the same level of confidence, but still good enough to evaluate whether it is relevant to continue working toward shared and fewer product architectures.

The AME method is very dependent on senior people in an organization, the top–down reasoning from the current state to future state is often difficult and there might be conflicting opinions. The viewpoint of the authors is that even though the "ideal" number of architectures is slightly higher or lower, it will not change the main conclusions. This means that a significant reduction in the number of product architectures is possible without compromising the market coverage.

Concerning application to the AME, the ideal company is mass producing with a history of mergers and acquisitions, distributed R&D, and manufacturing. Due to mergers and acquisitions, there will often be product lines with overlapping products. Due to distributed R&D and manufacturing, there will often be misalignment, that is, reinventing the "wheel" examples.

Conclusion

The article has presented a relatively simple method for calculating the benefits of shared architectures, the so-called AME method. There are three major contributions. The first one is an operational way to describe and count the number of product architectures. Second, the cost performance mapping shows in simple way how the performance steps of modules are realized. Number 3 contribution is top–down reasoning concerning the number of product and production architectures. From a practical point of view, the main contribution is the increased ability to have strategic discussion on the right number of architectures in a company based on facts.

There are many improvement areas in the AME method. One of them is finding out how to reason from requirements in the market, to the number of product architectures, and to the number of manufactured architectures. There must be sound principles for obtaining the right balance between the product and manufacturing architectures. Second, it should also be possible to include other quantifiable benefit dimensions such as time to market, R&D efficiency, and complexity reduction in manufacturing.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

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

References

Andreasen MM (1980) *Machine design methods based on a systematic approach—contribution to a design theory*. PhD Thesis, Department of Machine Design, Lund University, Lund (in Danish).

Andreasen MM and Olesen J (1990) The concept of dispositions. *Journal of Engineering Design* 1: 17–36.

Bruun HPL (2015) *PLM support to architecture based development: contribution to computer-supported architecture modelling*. PhD Thesis, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby, 188 pp. (DCAMM Special report no. S174).

Bruun HPL, Mortensen NH and Harlou U (2014) Interface diagram: design tool for supporting the development of modularity in complex product. *Concurrent Engineering: Research and Applications* 22: 62–76.

Du X, Jiao J and Tseng MM (2001) Architecture of product family: fundamentals and methodology. *Concurrent Engineering: Research and Applications* 9: 309–325.

Ericsson A and Erixon G (eds) (1999) *Controlling Design Variants—Modular Product Platforms*. Dearborn, MI: Society of Manufacturing Engineers.

Fixson SK (2005) Product architecture assessment: a tool to link product, process, and supply chain design decisions. *Journal of Operations Management* 23: 345–369.

Gonzalez-Zugasti J, Otto KN and Baker JD (2000) A method for architecting product platforms. *Research in Engineering Design* 12: 61–72.

Guo F and Gershenson JK (2007) Discovering relationships between modularity and cost. *Journal of Intelligent Manufacturing* 18: 143–157.

Guðlaugsson TV, Ravn PM, Mortensen NH, et al. (2014) Front-end conceptual platform modeling. *Concurrent Engineering: Research and Applications* 22(4): 267–276.

Hansen CL (2015) *On the identification of architectures for product programs In a complexity cost perspective*. PhD Thesis, Department of Mechanical Engineering, Technical University of Denmark, Kongens Lyngby.

Hansen CL, Mortensen NH and Hvam L (2012) On the market aspect of product program design: towards a definition of an architecture of the market. In: *12th international design conference—design*, Dubrovnik, 21–24 May.

Harlou U (2006) *Developing Product Families Based on Architectures—Contribution to a Theory of Product Families*. Kongens Lyngby: Department of Mechanical Engineering, Technical University of Denmark.

Herrmann JW, Cooper J, Gupta SK, et al. (2004) New directions in design for manufacturing. IDETC/CIE, pp. 1–9. Available at: http://proceedings.asmedigitalcollection.asme.org/proceeding.aspx?articleid = 1651563

Huang GQ, Zhang XY and Liang L (2005) Towards integrated optimal configuration of platform products, manufacturing processes, and supply chains. *Journal of Operations Management* 23: 267–290.

Hubka V (1973) *Theorie Technischer Systeme—Grundlagen einer wissenschaftlichen Konstruktionslehre*. Berlin, Heidelberg: Springer-Verlag (in German).

Hultink EJ, Griffin A, Hart S, et al. (1997) Industrial new product launch strategies and product development performance. *Journal of Product Innovation Management* 14: 243–257.

Hvam L, Mortensen NH and Riis J (2008) *Product Customization*. Berlin, Heidelberg: Springer-Verlag.

Kester L, Hultink EJ and Griffin A (2013) An empirical exploration of the antecedents and outcomes of NPD portfolio success. Available at: http://repository.tudelft.nl/view/ir/uuid%3A276695e8-1a3e-4914-a6d9-db8b8937cc3e/

Krause D, Eilmus S and Jonas H (2013) Developing modular product families with perspectives for the product program. In: Abramovici M and Stark R (eds) *Smart Product Engineering*. Berlin, Heidelberg: Springer-Verlag, pp. 543–552.

Levandowsky C, Michaelis MT and Johannesson H (2014) Set-based development using an integrated product and manufacturing system platform. *Journal of Concurrent Engineering: Research and Applications* 22(3): 40–58.

Lindemann U, Maurer M and Braun T (eds) (2009) *Structural Complexity Management: An Approach for the Field of Product Design*. Berlin: Springer.

MacDuffie JP (2013) Modularity-as-property, modularization-as-process, and "Modularity"-as-frame: lessons from product architecture initiatives in the global automotive industry. *Global Strategy Journal* 3: 8–40.

Meyer MH and Lehnerd AP (eds) (1997) *The Power of Product Platforms—Building Value and Cost Leadership*. New York: The Free Press.

Olesen J (1992) *Concurrent development in manufacturing—based on dispositional mechanisms*. PhD Thesis, Department of Engineering Design, Technical University of Denmark, Kongens Lyngby.

Otto KN and Wood KL (1998) Product evolution: a reverse engineering and redesign methodology. *Research in Engineering Design* 10: 226–243.

Pimmler TU and Eppinger SD (1994) Integration analysis of product decompositions. Design Engineering Division, American Society of Mechanical Engineers. Available at: http://web.mit.edu/eppinger/www/pdf/Pimmler_DTM1994.pdf

Skyttner L (2005) *General Systems Theory*. Singapore: World Scientific Publishing Co.

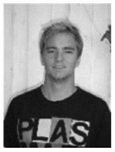
Stone RB, Wood KL and Crawford RH (2000) Using quantitative functional models to develop product architectures. *Design Studies* 21: 239–260.

Ulrich K (1995) The role of product architecture in the manufacturing firm. *Research Policy* 24: 419–440.

Author biographies

Niels Henrik Mortensen is professor of product development and do research on methods for developing product families.

Christian Lindschou Hansen is a PhD student focusing on complexity cost calcation.



Martin Løkkegaard is a PhD student focusing on development of system architectures.

Lars Hvam is professor of configuration management and do research on configuration system development and complexity cost calculation.