Scenario Based Development of Robust Product Architectures

Günther Schuh, Michael Schiffer, Jens Arnoscht
RWTH Aachen University, Laboratory for Machine Tools and Production Engineering WZL, Aachen, Germany

Abstract—Today many companies develop modular product architectures to face the challenge of maintaining a high degree of differentiation whilst reducing costs through economies of scale. To achieve the latter, the number of variants and product generations based on one architecture have increased significantly. This brings new challenges to the development process, especially when handling emerging technologies in dynamic markets. Expected changes in future product generations have to be anticipated when designing the architecture to cope with increased architecture lifetimes and minimize modification efforts.

A recent survey states that a high robustness of planning is a success factor when designing modular product architectures. Nevertheless most firms do not use the required tools in the early design stages to increase this robustness. Current research approaches for designing modular product architectures are mainly based on static requirements and thereby also neglect the dynamics of the market.

This paper aims at presenting a process model utilizing scenario planning in the design process of modular product architectures to minimize the modification efforts during the lifetime of a product architecture. The impact of the anticipated changes on the architecture is designed using a network analysis approach to identify critical architecture elements.

I. INTRODUCTION

In recent decades many industries show an increase in the variety of products due to different reasons. Often product ranges have grown over a long time without a proper controlling and harmonization. Today increasing dynamics in technology and innovation as well as a higher desire for more individualized products lead to a further acceleration of this trend. The situation becomes even more challenging because of emerging markets in Asia and Eastern Europe, which require specific products and hence increase the external complexity even more [1–3]. To simultaneously provide a fit of the produced products to the customer’s individual needs while participating in the price competition of globalized markets, therefore is one of the biggest challenges to be mastered by manufacturing firms [4].

To face this challenge and control this diversity, many companies are using or intend to use new product architecture approaches (e.g. platforms) to generate a high level of external diversity in the form of customized variants with a reduced internal diversity by combining the different architectural elements such as modules. This way, almost individually configurable end products are produced without having to renounce series cross-scale effects. Thus an increasing number of product variants and product generations is based on one product architecture that, in order also to be able to really take advantage of the economies of scale, has longer life cycles. This, however, has one major disadvantage, since increasing the lifetime of product architectures, while simultaneously extending the product range based on it, leads to an increased uncertainty in the customer's requirements to be fulfilled and rising costs of changes to the product architecture.

Given these problems, the increased dynamics of the market and customer requirements has moved even further into focus for manufacturing companies. While the time actually required to respond to environmental changes raises with the growing complexity of systems, the actual available reaction time decreases with an increasing volatility. Thus companies are facing the problem of an opening time gap in recent years. To face this situation, they have to anticipate these dynamics in the early stages of the product development and variant definition and take them into account for their products and product architectures [5, 6].

Hence future approaches for structuring products have to be adapted to these challenges. Currently this process is largely static and only inadequately involves future requirements, which means that the product architecture, which has been developed at great expense, has to be adjusted within only a few years or months or even has to be discarded again.

II. SURVEY ON THE EFFECTS OF DYNAMICS IN MODULAR PRODUCT PLATFORM DEVELOPMENT

In 2009 the Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University conducted a survey to analyse the status quo when designing modular product platforms, a special and very flexible form of product architecture for product lines with a high variance [7, 8].

One focus of the study was the success factor robustness of planning, which is essential for the sustainable design of product architectures. The questions on this success factor covered aspects such as the number of unplanned changes within the modular product platform due to dynamic disturbances. Characteristics in this area were [8]

- the number of unplanned module variants within a modular product platform generation,
- the number of unplanned changes to the modular product platform and
- the number of unplanned module variations for each new series.

Companies, in which this success factor is particularly well developed, are characterized by fewer changes made to both the modules and the modular product platform during the introduction and use of the platform. They use predictive planning tools in early development phases, such as
roadmaps, to prevent unplanned changes in the life cycle. Thus, unnecessary costs are avoided [8].

A significant deficit in today’s product architecture design shows that an average between 20% and 38% of module variants in a modular platform generation occur unplanned. These unplanned changes mean that new modules cannot be implemented in all products and thus the economies of scale in the form of new standards can only be reestablished in the medium term [8].

Task of release management is to plan and control the release and introduction of new modules in the modular platform and thus in each series. It requires such a specific control to help prevent an unplanned expansion of the modular platform with the associated changes to the platform and to continuously align the products with the customer needs. However, many companies today do not fully exploit these potentials of a systematic release management. Thus, the use of a module roadmap is rarely established and used by only approximately 20% of the surveyed companies [8]. This is even more concerning as the study’s results show that firms that use a module roadmap, are among the outperformers in profitability. Financial benefits are the outcome of the fact that less change and adaptation efforts are necessary and therefore the associated costs can be saved. According to the survey, the number of unplanned changes to the modular platform is lower for companies using module roadmaps (see fig 1).

The shown deficits illustrate that today’s product architectures are designed statically based on the current requirements without an adequate regard to the future dynamics of these requirements. Furthermore, alternative product architecture designs cannot be evaluated as to whether they have a sufficient robustness and thus minimize future change and adjustment costs. The result is a structure which does not fully exhaust the achievable commonality potential.

III. RELATED WORK

As section II shows, there certainly exists a deficit when designing product architectures for dynamic environments. This section aims at presenting existing research approaches when developing product architectures and evaluating variants, to analyse whether and, if so, how they cope with dynamic future requirements.

Analogous to the failure mode and effects analysis (FMEA), the Variant Mode and Effect Analysis (VMEA) aims on detecting variants early and avoiding exotic ones, so as to control the number of variants. The variant control is carried out at both the technical and economic level and is already deployed in the concept phase. Based on a static Market oriented identification and design of product functions, design alternatives are derived and assessed to find the optimal product design [9].

Design for Variety (DFV) is a method that supports the development of product platforms, with a focus on both making it insensitive to influences from outside as well as to reduce interactions and dependencies between the system elements. Therefore MARTIN defines the Generational Variety Index (GVI) and the Coupling Index (CI) to analyze the effects of changes in individual components. Furthermore, he defines a set of rules, giving the order to optimize the components for a robust product platform, as a function of their indices [10, 10]. However he does so without describing the essential influencing parameters on the product architectures and possible prediction methods.
The modular product development by Göffert deals with the technical and organizational design of modular product architectures. For this purpose he defines the METUS (Management Engineering Tool for Unified Systems) Method, which is both an approach to the technical and organizational design of modular product architectures as well as a specific form of organization for a modular product development. The approach also covers the early concept and design phases [11]. Dynamic requirements and their impact on the product architecture are not considered during the design or the assessment of different design alternatives.

In his work, Kohlhase discusses the structuring and evaluation of modular platforms. In addition to the development of a general process model that is based on the work of Pahl & Beitz [12], the focus of the work is primarily on the cost analysis. For the latter, he describes a method for the target costing of modular platforms [13]. Kohlhase also developed useful suggestions for structuring modular systems based on the development of structural alternatives and their evaluation. The assessment is based on the technical and economic value and the flexibility offered by the modular platform [13]. Effects of dynamics are rarely covered by the assessment.

For the development of scalable product platforms and the corresponding product families Simpson et al. [14] developed the Product Platform Concept Exploration Method (PPCEM). In this method the determination of the relevant design parameters is based on a mathematical model, the Decision Support Problem (DSP). Using the DSP, the extent of the similarities and variations can be determined. A dynamic of the influencing parameters and appropriate design strategies to meet them, however are not integrated in the model.

SEKOLEC developed a methodology to support the development of modular product families in the early phases.[15] The focus is on modularization, structuring and evaluation of product families. To assess the defined product family, SEKOLEC uses multiple measures that, displayed in a portfolio model, permit conclusions about the quality of the structure and allow its systematic optimization. The approach mainly focuses the optimization of the external variety. The influence of the dynamics on the design of the internal diversity is only inadequately included.

In their approach Commonality Decisions in Product Family Design Fellini et al. [16] also focus the tradeoff between commonality and differentiation in the development of product families. The goal is to develop a platform so that resulting product variants are optimally designed regarding commonality and differentiation. In a first step, the optimal platform (zero-platform) for the individual products is determined, which is then analyzed for the components that could be used across all products. Based on the maximum acceptable performance loss for the individual products through the use of communal assembly groups (determined by so-called performance loss factors), the platform parts are determined. In this approach, the design parameters to be optimized are only shortly defined and it is also does not discuss the influence of dynamics on the design parameters and the platform design.

Considering the relevant literature it becomes clear that existing approaches to the design of product architectures in general and modular platforms in particular mainly base on static requirements and thus not sufficiently take into account the dynamics of the environment.

This conclusion is underlined by the work of Fixson, who analyzed 160 papers from 36 journals in the areas of modularity and commonality with respect to their subject, the covered performance effects and the applied research methodologies. The results show that papers in the subject area of commonality and modularity focus the effects on the quality, variety, cost and time, whilst the adaptability and flexibility advantages are rarely investigated [17].

IV. DEVELOPING ROBUST PRODUCT ARCHITECTURES

The first and one of the most important steps to develop a product architecture that is robust to different future scenarios, is a better predictability and therefore safety of the product range while defining the product architecture in the early stages of product development. Thus, on the one hand efforts for changes to the product architecture can be avoided, while on the other hand a sustainable competitive advantage can be achieved through a targeted addressing of customer needs. To realize such predictability and also take into account dynamic customer needs, the classical methods for product architecture design (see chapter III) must be extended with dynamic components.

Based on these objectives, the methodology for the scenario based development of robust product architectures has four major phases: In the first stage future projections are created as the basic input to the methodology, using the scenario technique. The created scenarios are translated into appropriate product ranges for each scenario, which are described through features and their possible specifications using the feature-tree-technique. This allows the identification of key factors with great influence on the dynamics in the product. Based on the effect relations and intensities of the influences of these key factors, the uncertainty of the realization of individual characteristics can be evaluated across all possible scenarios. In phase two design strategies for individual areas of the product are derived based on the uncertainty analysis from phase one. The design strategies affect the product architecture, e.g. in terms of interface flexibility and module boundaries. These design recommendations are used in phase three to create different product architecture alternatives that are evaluated for their scenario-robustness in phase four (see fig 2).
A. Identification of factors influencing the dynamics of the product range

In the area of strategic planning exists a variety of methods with the purpose of an accurate anticipation of future developments. For example the scenario technique is an established method for the systematic development of future projections and widely-used for corporate planning as well as technology roadmapping [18–21]. It demands thinking in alternative future projections, so that the dynamics and uncertainties of the future are considered [6].

The first step in order to design scenario-robust product architectures based on the scenario technique, is the constitution of future projections. The future projections describe the product environment at time points along the product life cycle. Based on the scenarios product requirements can be derived. This may be the same basic requirements across all scenarios, or scenario-specific requirements that are representative for dynamics and uncertainties in the market environment.

The creation of scenarios for the scenario robust design of product architectures does not differ fundamentally from the general procedure for creating scenarios from Gausemeier [22, 23]. Here, first the objective and the design field is set for the scenarios. Then the key influencing factors, with a particularly large influence on the development of the focused field, are selected from the possible influencing factors on the scenarios. Thereafter, possible developments for the key influencing factors are determined. At last coherent projections of key factors are clustered to scenarios [22].

To adapt this methodology for the purposes of the product architecture design, the influence factors that form the starting point of the scenario technique to forecast the future, are to be examined to determine how they affect the product range and therewith the product architecture.

In the first instance this means the development of descriptions of the product range for the various images of the future, based on product features and their possible specifications. The starting point is the anticipation of product features based on the requirements for the product. For this purpose, the established scenarios provide an ideal starting point. Textual or pictorial descriptions of the scenarios support the creative process to anticipate future developments and to understand the implied requirements for new products. These must be extracted from the descriptions of different scenarios and convert into product characteristics and their respective specifications. Once the description of the essential features and their possible specifications of the product have been made, the product range can be determined by the combinatorics of the specifications and the formulation of restrictions and mandatory combinations. The resulting feature tree is the solution-neutral description of the planned new product range for each of the underlying scenarios.

To meet the idea of different visions of the future as a planning basis, the construction of product ranges based on features and specifications is done scenario-specific. This reflects the fact that the various developments in the key influencing factors may entail different requirements for the new product. This is reflected in different, optimal product ranges for each scenario (see fig 3).
To enable the targeted derivation of design strategies for the optimal product architecture for all possible scenarios, an assessment of the interactions between the key influencing factors and the individual product specifications is required. This interaction can be represented by mapping the interactions between key influencing factors and the specifications of the product features. In addition to mapping a relationship and the effective direction of the relationship, it is classified whether it is strong, medium or weak. Thus, key factors are identified which have a particularly large influence on the planning reliability. These must be monitored as part of the product architecture design process and have to be reviewed for possible future occurrences. That way the anticipation of possibly changing requirements for a product becomes possible and probability profiles for characteristic features of the product range can be created (see fig 4).

B. Derivation of design strategies for the product architecture

To implement the obtained knowledge about the future product range in a product architecture that minimizes future change costs and times under the given probabilities, the effects of environment-induced dynamics to the product architecture must be known. The network analysis is an approach to model product architectures and the impact of changing features and their specifications.

The external variance should be realized through product features with a low effect on other product features. Such features are called uncritical. A high variance of critical product features, however, should be avoided. A critical product feature is characterized by affecting process steps that are very inflexible and thus are very sensitive to variance in the affecting features. Furthermore, critical product features have a major impact on the overall system and are almost not influenced by other product features. Fig 5 illustrates the described issue by the example of an industrial valve.
Once the customer oriented features of the product range have been mapped to the product features, the impact of changing feature specifications in the product range can be evaluated. That way the critical elements of the product architecture can be identified. These elements for example dampen the dynamic effects or have a particularly high sensitivity to changes in requirements. If these are designed according to certain rules, a maximum robustness can be achieved. These design rules or strategies are gathered in a solution catalog. For this purpose, different types of product architectures and their elements such as modules or platform components are classified regarding the way they deal with the uncertainty of the realization of certain specifications. Using this classification, design strategies for different classes of probability profiles are defined, which take into account the conditions that result from the specific profile. Depending on the assessment of the realization probabilities of individual specifications of the product features as well as the impact analysis, results the selection of the different design strategies from the catalog of solutions.

However, selecting the right design strategy is not only based on the considerations of the realization probabilities of the specifications, but also on accompanying factors such as cost considerations. This allows different design strategies for a probably less popular product feature. Two possible design strategies would be the local modularization of the architecture for an easier retrofit or interchangeability as well as an over-sizing in terms of realization of a maximum possible specification. Since the strategies design for individual product features require each other or are mutually exclusive, e.g. in the realization of several product features in one component, an evaluation and selection of appropriate design strategies is only possible in a general context by comparing different product architectures. Thus for an assessment of the scenario robustness, various alternatives of possible product architectures have to be designed.

C. Development of alternative product architectures

To obtain a truly optimal solution for different future scenarios, different product architecture designs are developed based on the design strategies. The development of the product architectures is conducted according to the product architecture development process (PADP) [24, 25]. This process is designed to meet the new challenges for the product structuring logic and the process and organizational design of the product development itself, which arise due to the enhanced degrees of freedom in the design of new product architecture types such as modular product platforms. These are characterized by an almost free combinability of individual modules and their multiple use in different products. It structures the design of basic product architectures into 10 phases (see fig 6) and the cyclical design matches the often iterative approach of product development projects for technical products [25].

Based on the possible specifications of the product features and the recommended design strategies, the final design of the alternative product architectures takes place in the steps 4 to 8 of the PADP (For a detailed description of the PADP process steps see [24, 25]). In addition to the recommendations based on the scenario analysis strategic considerations play a role in these steps. Besides planning the use of identical parts in order to achieve economies of scale, particularly the company's strategy on its core competencies for the various technologies integrated in the product architecture has to be considered. Thus the encapsulation of functions, whose technological realization was defined as a core competence in the company's strategy, may be appropriate when making make-or-buy decisions, to keep their production within the company.

![Figure 6: The Product Architecture Development Process (PADP) [25]](image-url)
In the last step, the developed product architecture alternatives have to be evaluated regarding their scenario-robustness. Here the documented cross-linking of the key influencing factors on product features and finally the components, allows the evaluation of each architecture alternative regarding the uncertain realization of individual feature specifications expressed by the scenario specific feature trees. To carry out a full assessment of the product architecture, the design-dependent piece and implementation costs as well as the scenario dependent quantities have to be estimated. In addition to the initial implementation costs, the modification costs induced by changing requirements with effects on the product line and the product architecture have to be anticipated.

The assessment therefore includes both the revenue and the expense side. Overall result three evaluation criteria:

1) Initial costs
2) Expected changing costs
3) Expected revenues

The initial costs consist of the initial development costs for the realization of a product architecture alternative and the associated investment costs, e.g. for the production resources, etc. Here it is important to consider all costs associated with the initiation in the company. A global allocation of development and other process costs should be avoided here.

For the evaluation of the expected change costs of the product architecture alternatives, the distinction between potential and secure existing customer needs is especially important. Potential needs are characterized by the fact that they only arise on the basis of a subset of the scenario projections. Secure existing requirements on the other hand represent such requirements arising from the description of all scenario projections. For the assessment of potential requirements, the divergence in the projections of the key influencing factors for each of the relevant product architecture elements is most important. This ensures that the changing costs are always viewed in the context of the actual needs for a change, resulting from differences in the projections of the key influencing factors. The estimation is done here on the level of the different feature specifications. Is specification e.g. not provided for the product architecture alternative, but is required by the customer in a scenario, the costs for the subsequent inclusion in the product architecture can be estimated and weighted by the probability of the scenario.

Finally, it is important to calculate the expected revenues. Therefore the different prices, number of pieces and piece costs for the variants have to be estimated depending on the scenario and the selected alternative product architecture. These in turn are then weighted by the probability of the scenario and summed. The various partial evaluations are finally brought together in the overall scenario robustness of a product architecture (see fig 7).

So ultimately the one product architecture alternative, allowing the best possible coverage of the customer requirements on the one hand and minimizing the change costs for needs not covered on the other hand, can be selected.

V. CONCLUSION

The presented methodology shows the potential of a combination of the scenario technique and the design of product architectures. To unlock this potential a methodological approach was developed. The design of alternative projections of the future allows the detection and assessment of changes in the customer requirements in relation to future development. The consideration of these dynamics in the customer requirements for the design of the product architecture, the scenario based development of robust product architectures, allows a significant reduction in change costs in the later stages of the product life cycle. While existing approaches to the design of product architectures mainly base on static requirements and thus not sufficiently take into account the dynamics of the environment, scenarios are an explicit component in this approach to designing product architectures.

ACKNOWLEDGMENTS

The presented results have been developed within the research project “Scenario based development of robust product architectures” funded by the Deutsche Forschungsgemeinschaft (DFG).
REFERENCES


