Proceedings of DETC ’07
ASME 2007 International Design Engineering Technical Conferences and
19th International Conference on Design Theory and Methodology (DTM)
Las Vegas, Nevada USA, September 4-7, 2007

DETC2007-35915

Platform Investment Decisions in Product Family Design

Nicolas A. Zacharias & Ali A. Yassine[*]
Product Development Research Laboratory
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

The paper proposes an intermediate modeling ground that bridges the gap between engineering design models and marketing models for the development of platform-based product families. In this model, each variant in the product family is assumed to contribute a percentage to the overall market coverage inside a target market segment and we wish to maximize this coverage subject to an available development budget.

Following the conceptual engineering design phase of the product family, this model will optimize the initial investment in the platform, the commonality level between variants, and the number of variants to be produced in order to maximize market coverage. An application of the model using a Drill product family is included to demonstrate the usefulness of the proposed model.

(Keywords: Product Development, Product Family Planning, Platform Design, Product Architecture, Product Commonality).

Taxonomy

B = budget available for the whole project
α = commonality factor ranging between 0 and 1
i = index for the proposal number; $1 \le i \le P$
P = number of variants proposed by engineering department
N = number of variants created in the project
V_i = 1 if the i^{th} proposal for a variant is implemented, 0 if the i^{th} proposal for a variant is not implemented
C_i = cost to create the i^{th} proposal
D_i = development cost of the i^{th} proposal
V_0 = development base value
E_i = effort factor of the i^{th} proposal
m_i = percentage of modules changed in the peripheral set when creating the i^{th} proposal
T_i = integration and testing cost of the i^{th} proposal
A_0 = integration & testing base value
X_i = complexity factor of the i^{th} proposal
f_i = number of interfaces to adapt to create the i^{th} proposal
F = flexibility factor of the platform
I_0 = initial investment in the platform
β = investment ratio of I_0 to B ranging between 0 and 1
M = market coverage in the target market segment ranging between 0 and 1
a = share of variable costs in the development process of variants
b = share of fixed costs in the development process of variants
c = commonality of benchmark variant for integration and testing
d = interface complexity factor
e = benchmark constant for β
f = target market segment fitting constant
g = scaling value for market coverage

1. Introduction

Product families are defined as a group of related products that share common features, components, and subsystems, where the common parts are referred to as the product platform (Simpson *et al.*, 2006). Usually the concept of a platform-based product family is realized using a modular product architecture, which simultaneously provides the ability to supply sufficient variety to the market while maintaining the economies of scale and scope within the companies’ manufacturing capabilities (Jiao *et al.,* 2006). Several trade-offs are involved in the platform design, such as the balance between distinctiveness and commonality, the choice of a product architecture, and product performance issues (Martin and Ishii, 2002; Mark *et al.*, 2005). Investment constraints and market related objectives must also be taken into consideration, since they have high impact on the success

---

[*] Corresponding author: yassine@uiuc.edu. Tel. (217)333-8765.

of the platform-based product family and its development process.

Recent literature shows that existing models of platform-based product families are either too engineering oriented (Simpson *et al*., 2001; Messac *et al.,* 2002) or too marketing oriented (i.e. taking primarily marketing considerations into account) (Vollerthun, 2002; Mark *et al.,* 2005). This paper proposes an intermediate modeling ground that bridges this gap by simultaneously considering important engineering and marketing issues. The goal of this paper is first to understand the interdependencies between engineering design, development cost, and market coverage. Then, based on this understanding we propose a new model for designing platform-based product families taking into account these interdependencies. The proposed model provides answers to the following: (a) how much of the total funds should be invested in the development of the platform, (b) which variants should be created (the total count of variants will be a side result as well), and (c) the optimal commonality used in the product family.

The paper is organized as follows. In Section 2, we review the relevant literature and make some comparisons to the proposed model. Section 3 introduces the base model including problem formulation, related numerical functions, and the optimal analytical solution. The application of the model to realistic scenarios using a Drill product family case study is presented in Section 4. A discussion of case study results, along with various sensitivity analyses are covered in Section 5. Our conclusions, limitations of the model, and future research directions are presented in Section 6.

2. Literature Review

Several papers study the possibilities for optimizing the development process of a product family. For instance, Holtta and Otto (2003) introduces a procedure to design products with optimal complexity at the module boundaries so that changes in the modules require minimum redesign effort. Hsuan (1999) develops a mathematical model to analyze the opportunities for modularization of a product with respect to its interface constraints at different levels (e.g. component or modules level). Others focus on the trade-off between the benefit of part commonality and the inherent performance loss linked to it (Fellini *et al.,* 2000, 2002). Gonzalez-Zugasti *et al.* (2001) deal with the problem of trading-off higher variety of products with an accompanying reduction of resources available for their development.

Further literature exists on the optimization of the product family design for efficient manufacturing (Bi and Zhang, 2001). Yigit and Allahverdi (2003) address the needs caused by rapidly changing markets using a reconfigurable manufacturing system for rapid adjustment of production capacity and functionality. A production cost model for product family design is introduced by Park and Simpson (2003), which balances the commonality of the product platform with its distinctiveness of the individual products in the family. Reducing production cost can be achieved by keeping production volumes as high as possible. This leads to the main objective of product family designers, which is to select the right components or modules to share between products to maintain economies of scale with minimum sacrifice in the performance of each variant in the product family. Along similar lines, Michalek *et al*. (2006) proposed a model that trades-off the reduced cost and reduced revenue resulting from designs that less expensive to produce (e.g. variants based on a common platform) but also less desirable in the marketplace (e.g. capture a lower market share).

Vollerthun (2002) presents an approach to integrate conceptual design and marketing by using a system engineering approach. He focuses on the necessity of all products to meet essential customer requirements of the targeted market segment. The model is meant for the early engineering phases and specifies the interrelations between conceptual design, life-cycle cost, and market revenues. Mark *et al.* (2005) focus on a system-level approach for embedding flexibility in the design of a new engineering system, along with potential performance penalties. He presents an optimization framework to design flexible systems and to quantify the trade-offs inherent to flexibility. Allada and Lan (2002) deal with the optimization of module deployment at the product planning stage. They formulate the problem as a dynamic program to answer the questions of when to replace specific modules in a product family, what modules need to be replaced, and what modules will serve as replacements.

Our model differs from those described above in several ways. First, we link conceptual engineering design, development cost, marketing decisions, and their interdependencies, into a well-balanced optimization model for the design of platform-based product families. Second, the objective to maximize market coverage does not cause the model to neglect the optimization of the product architecture by providing feedback to conceptual engineering design. Third, the outcome of the model is a platform-based product family, possesses optimized product architecture, and maximizes market coverage. Finally, in this paper we assume that all variants would be built from a single common platform. In some cases, however, when the differences between variants are too great, this may not be true. In those situations, multiple platforms may be required as in the model described by de Weck et al. (2003).

3. Base Model

The development of a platform-based product family requires a variety of decisions. At the beginning of the development process, management decides on the target market segment and allocates a specific budget (*B*) to the project. The objective of this model is to maximize the market coverage of the whole product family. To achieve this goal, a product platform will be developed, which consists of the modules common to all variants. Furthermore, there are peripheral sets of modules used to produce product variants in combination with the platform. Interlinks between the modules will be defined as interfaces, which define interactions among modules.[1] The fraction of common modules represented by the platform compared to the total count of modules used for a specific product derivative/variant will be expressed by a commonality factor α. Note that the platform by itself cannot be offered as a product in the market. Corresponding peripheral sets of modules will have to be created, which will result in products when combined with the platform. Each variant represents a different product that can be offered in the target market segment.[2] All variants together outline the final product family, which can be introduced into the market either all at once (i.e. when the development of all variants is completed, the whole product family is offered in the market) or in a phased fashion (i.e. variants are introduced in the market one after another).[3]

In this development project a variety of costs are involved. Although the budget could be altered at a later stage of the project, we assume a fixed investment amount allocated for the whole development project. The funds can be assigned to cover the various development expenses. There is an initial investment to develop, design and test the product platform. There are additional costs for creating each variant, which must be designed, integrated with the platform, and tested in conjunction with the platform.

3.1. The Product Architecture

The major engineering information used in this model is derived from the function diagram of the product family (Pahl and Beitz, 1996). Depending on how closely the different variants are related, there is one function diagram required for the whole product family or one diagram for each of the possible variants. The functional structure will represent the base upon which the product is divided up into modules.[4] Using a function diagram, modules can be labeled as boxes around one or more of the functions (Otto and Wood, 2001). Modules which are common to all products (even across different function diagrams) represent the platform of the product family. *Energy, material,* and *signal flows* between functions of different modules represent interfaces (Alizon *et al.*, 2006). Matching each of the flows in the function diagram into the physical structure of the product results in one or more interfaces of different complexity (e.g. a mechanical *energy flow* could be realized using two physical couplings). Holtta and Otto (2003) model this matching process by assigning an *interface design effort complexity* to each *flow* to rate its design difficulty. Since interfaces are not the focus of our model, we assume, for simplicity, equal complexity inherent to all interfaces, and thus match each *flow* in the function diagram with exactly one interface in the physical structure of the product.[5]

When creating a new variant, m_i will represent the percentage of modules changed in the peripheral set, where $1 \le i \le P$ and P is the total number of variants proposed. The corresponding f_i counts the number of interfaces between the non-changed and the changed set of modules involved in the redesign of those modules. Figure 1 provides an example of the design of a new variant having 50% of modules changed in the peripheral set (i.e. $m_i = 0.5$) and involving 7 interfaces (i.e. $f_i = 7$).[6]

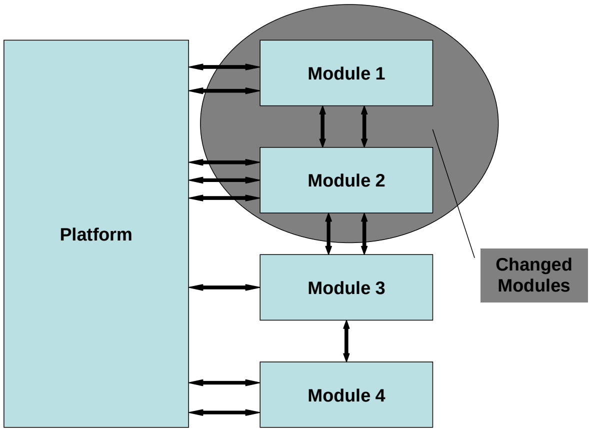


Figure 1: Modular architecture of a sample product including changed modules and interfaces involved

In addition, a conceptual design of the product is another input to the model. Although the output of the model will influence the final conceptual outline of the product (i.e. the optimal commonality factor), engineering must initially come up with a pre-design for the product family based on the function diagram(s). One way for facilitating this step is to investigate the preliminary location of each of the key product functions which are most valued by the customer, which could be placed either in the platform or in the peripheral set of modules.[7] A placement of the majority of key product

---

[1] These include all sorts of couplings including energy, signal and material flows (Pimmler and Eppinger, 1994).

[2] By target segment we mean a market niche possessing similar customer needs (e.g. the sports car market). This approach will not be suited to create a product family to serve the whole market (e.g. the car market), since a platform strategy may not be adequate for such large discrepancies in customer requirements.

[3] Both introduction strategies are further explored and explained in Zacharias (2006).

[4] This step can be optimized using the Design Structure Matrix (DSM) to reduce the number of interfaces between the modules (Yu *et al.*, 2003) or by using other heuristics approaches such as the dominant flow heuristic (Stone *et al.*, 2000; Otto and Wood, 2001).

[5] This assumption can be relaxed in future research, possibly using similar approaches as Holtta and Otto (2005).

[6] Note that interfaces between changed modules do not count.

[7] Methods of quality function deployment (QFD) can be used to ascertain the most important design features/functions for satisfying customer requirements (Otto and Wood, 2001). Once these design features are identified, they can be matched to functions in the function diagram, which we denote as key product functions (KPF). Key product functions do not

functions in the platform will lead to higher commonality (α) but also to less differentiation since customers detect differences between variants mainly based on these key product functions (e.g. product performance, form, *etc*.).

Having decided on a preliminary platform design, engineering is expected to suggest a number of possible variants P for the product family. Since this includes information about the modules subject to change as well as the associated interfaces, one or more sequence must exist, in which the proposed variants are derived from each other. Starting with the first base variant, one or more variant can be developed in a predetermined order. Figure 2 outlines a possible product family consisting of 10 proposed variants split into 3 sequences. Of course, the unconstrained case is a special case where any variant can be derived from any other existing variant.

3.2. Cost Function for Variants

The model captures the complexity of developing proposed variants in a single cost function C_i for each proposal i. To create a certain variant, the cost to develop the peripheral set of modules (D_i) is summed with the cost to integrate and test this set combined with the platform (T_i). Therefore, the total development cost of proposal i is:[8]

$$C_i = D_i + T_i \qquad \text{Eq. (1)}$$

The development cost for proposal i consists of a base value V_0 multiplied by an effort factor E_i (that is, $D_i = V_0 E_i$). The base value, V_0, is estimated from the development cost of a null-platform product in the same market (i.e. a single integral product), serving as a benchmark. As a result, D_i is always less than or equal to V_0.[9] The effort factor, E_i, depends on two other factors: The percentage of modules changed in the peripheral part of the product, m_i, and on the commonality factor, α. For the borderline case (where $\alpha = 0$ and $m_i = 1$) the whole product must be redesigned.[10] For the other borderline case (where $\alpha = 1$ and $m_i = 0$) the redesign effort will be close to zero; nonetheless, a certain amount of fixed cost must always be accounted for in any development process. For values of α and m_i between 0 and 1 the effort factor will be lower than 1. Hence, E_i will be modeled as a decreasing function of α and an increasing function of m_i; therefore, $E_i = f(\alpha, m_i) = E_i(\alpha, m_i)$.

The integration and testing cost for a proposed variant depends on the complexity of the peripheral set as well as on the flexibility of the platform to adapt to a new set of modules. It is calculated by multiplying a base value A_0 with the complexity factor X_i and the flexibility factor F; that is, $T_i = A_0 X_i F$. The integration and testing base value, A_0, is also estimated from the cost of integrating and testing a basic benchmark product. Since X_i and F are generally greater than 1, the integration and testing cost for a variant is a multiple of A_0.

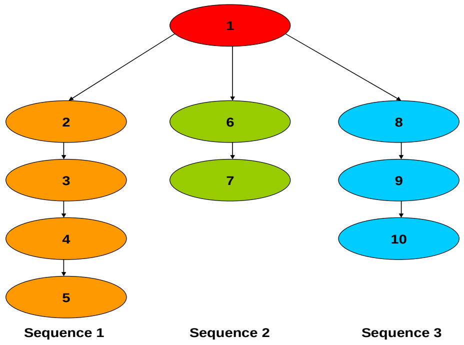


Figure 2: Possible product family divided up in development sequences

The complexity factor X_i is influenced by both the number of interfaces associated with the change of modules (f_i) and by the commonality factor α (i.e., $X_i = f(f_i, \alpha) = X_i(f_i, \alpha)$). The more interfaces between the changed modules and the unchanged modules (which are subject to adaptation), the more difficult is the integration task of the variant. Therefore X_i is an increasing function of the number of interfaces having to adapt (f_i).

The higher the commonality factor (α) is, the more key functions of the product are likely to be implemented in the platform. This is synonymic of a high fraction of well-tested functions which have been already proven reliable. Since α is going to be constant across all variants, a higher α results in a lower integration and testing complexity. Alternatively, a low α depicts a relatively large peripheral set in which changes are going to occur. Hence, the integration and testing complexity increases when a larger portion of the product is changed and thus cannot be used as a reliable base for integration and testing of future variants. Thus, the complexity factor X_i is also a decreasing function of the commonality factor α.

The flexibility factor F is a measurement for the adaptation ability of the platform. It stays constant across all variants because it is platform dependent. In determining F, the initial investments (I_0) in the platform and the number of variants N to be created play a key role ($N \leq P$). I_0 is directly related to the design effort put into the platform. As more money is invested in the platform, the less variant specific the platform architecture becomes. A high initial investment will result in a flexible platform, which can be easily (in this case cost effectively) adapted to new variants. With an increasing

---

necessarily map with modules as there can be several KPF in a module or none.

[8] Since the focus of our model is on design and development of platform-based product families, Equation (1) does not explicitly consider the production cost of variants. However, our model could be easily extended to include other essential life-cycle costs such as production and service for example.

[9] The cost of developing a variant can never exceed the cost of creating an integral product, regardless of whatever changes are made to the variant.

[10] Hence the development cost will be V_0 (the cost for designing a null-platform product) and E_i is assumed to be 1.

number of variants the constraints for the platform increase. Since it still has to fulfill the needs of all prior variants, it cannot freely adapt to the new set of peripheral modules (i.e. the interface constraints on the platform increase as the number of variants rises). Taking both of these facts into account, the flexibility factor F will be modeled as a decreasing function of the initial investment I_0 but at the same time as an increasing function of the number of variants based on it; that is, $F = f(I_0, N) = F(I_0, N)$.

In summary, the cost function to create proposal i becomes:

$$C_i = V_0 E_i(\alpha, m_i) + A_0 X_i(f_i, \alpha) F(I_0, N) \qquad \text{Eq. (2)}$$

3.3. The Market

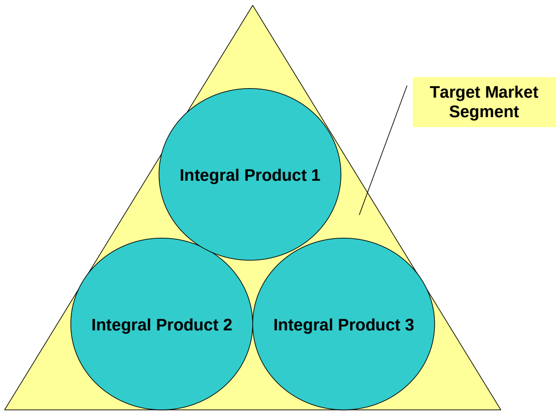
To develop an expression for the product family's market coverage, we assume that a specific number of integral products (i.e. one-of-a-kind products designed independently without the use of a common platform) are able to cover the whole target market segment. Based on this benchmark scenario, we derive the market coverage of the platform-based product family. As an example, a market which is fully covered with three integral products is shown in Figure 3(a). Accordingly, we assume that three variants of a product family with a commonality factor of 0 (null platform) will have approximately the same market coverage as those three integral products.

In our model, as the commonality increases, the variants have less chance in satisfying the customer requirements in the entire target segment. If the platform is placed as an atomic point in the middle of the target segment (the platform can not be offered as a product) the variants are placed at a certain distance away from it. The distance depends on the degree of differentiation between the variants and hence is inversely related to the satisfaction of customer requirements. Figure 3(b) illustrates a product family with three variants and a medium sized differentiation within the same market segment of Figure 3(a). The lower market coverage of the platform-based family, compared to the integral products, is mainly due to commonality and overlapped market coverage of the multiple variants. In an extreme case where the commonality factor is close to 1, the variants will be very closely related and hence will not create much differentiation. Even if the maximum number of variants is created, it will be difficult to reach high market coverage. Taking these considerations into account the market coverage contributed by each product depends highly on its uniqueness compared to other products in the same product family. Since the effort factor (E_i) gives a good estimate of how different a proposed variant is compared to another variant, it must also represent a good proxy for a variant's market coverage compared to a null-platform product (E_i=1). Therefore, we will assign a weight to each created variant in the market proportional to its effort factor.

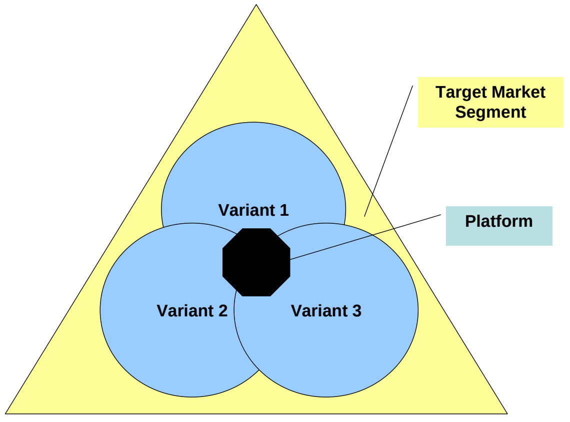
The total market coverage M, which increases as more variants are introduced, is obtained by summing up the market coverage contribution of all variants. Nonetheless, the market coverage function must be diminishing with respect to the number of weighted variants created since the chance of covering the same parts of the market twice rises as more variants are introduced (Kim and Chhajed, 2000). Consequently, the market coverage function is as follows:

$$M = f\left(\sum_{i=1}^{P}(V_i E_i)\right) = M\left(\sum_{i=1}^{P}(V_i E_i)\right), \qquad \text{Eq. (3)}$$

where V_i= 1 if the i^{th} proposal for a variant is implemented, or 0 otherwise.



(a) using integral products



(b) using a platform-based product family

Figure 3: Target market segment coverage

3.4. Problem Formulation

The model maximizes the market coverage for a given budget and initial investment in the platform. Furthermore, it will also give guidance to which of the proposed variants must be created as well as what the optimal commonality factor is.

Collecting the constraints and modeling assumptions together, we have the following integer (and binary) non-linear program:

$$\underset{\alpha, V_i}{Maximize}\, M \quad \text{(maximize market coverage of family)} \qquad \text{Eq. (4)}$$

Subject to:

$$V_i \in (0,1) \;\; ;1 \leq i \leq P \; \text{(which proposals to implement)} \qquad \text{Eq. (5)}$$

$$I_0 + \sum_{i=1}^{P} (V_i C_i) \leq B \; \text{(available budget)} \qquad \text{Eq. (6)}$$

$$0 \leq \alpha \leq 1 \; \text{(commonality factor)} \qquad \text{Eq. (7)}$$

$$\sum_{i=1}^{P} V_i = N \; \text{(total number of variants)} \qquad \text{Eq. (8)}$$

Furthermore, additional sequence constraints can be added to the formulation depending on the development environment. For example, the constraints matching the 3 sequences from Figure 2 are depicted in Table 1.

Sequence 1	Sequence 2	Sequence 3
$V_1 \geq V_2$	$V_1 \geq V_6$	$V_1 \geq V_8$
$V_2 \geq V_3$	$V_6 \geq V_7$	$V_8 \geq V_9$
$V_3 \geq V_4$		$V_9 \geq V_{10}$
$V_4 \geq V_5$		

Table 1: Sample sequence constraints as shown in Figure 2

To expand the model further, we introduce the variable *β* which can be varied iteratively in order to come up with a solution for the market coverage M depending on *β*. The global maximum market share *M** for this product family will then be linked to an optimal investment ratio *β**.

$$\beta = \frac{I_0}{B} \qquad \text{Eq. (9)}$$

The model presented in Equations (4) through (9) can be solved analytically (i.e. exact solution) if few additional assumptions are made. Finding a general solution is discussed elsewhere (Zacharias, 2006). First, the discontinuity of the solution must be addressed. Since the problem formulation involves binary variables and hence the total number of created variants *N* is an integer, the solution function will only be continuous as long as *N* stays constant. This is accounted for by solving for each possible *N* within a reasonable range and then comparing the optimal *M* values of each solution to find the global optimum *M** at *β**.

Since market coverage is a monotonically increasing function of the number of variants created it can be assumed that the whole budget is always spent on the project. To reduce the number of variables even further we assume f_i and m_i to be the average of all the proposed variants, which will also lead to a constant E_i across all variants.[11]

Using these assumptions in the budget constraint in Equation (6) and combining it with Equation (9), we can derive Equation (10) for a predetermined *N*. In this equation C_i is the same across all proposed variants *i* and subject to the variables *α* and *β*.

$$(1-\beta)B = NC_i \qquad \text{Eq. (10)}$$

This equation can now be solved for *α*, by substituting Equation (2) in Equation (10), which creates a function for the commonality (*α*) depending only on *β*. Then, this function for *α* can be plugged into the effort factor portion of the market coverage function, which eliminates *α* as a variable, and yields equation (11). Analysis of Equation (11) and its properties provides the way to determine a closed-form solution for the optimization model presented in Equations (4)-(9).

$$M = f(\beta) \qquad \text{Eq. (11)}$$

Corollary 1: *M* is a decreasing function of *α*.

Proof: From Equation (3) it can be seen that *M* is an increasing function of E_i. Furthermore, the effort factor is a decreasing function of *α*, which results in *M* being a decreasing function of *α*.

Theorem 1: For a predetermined *N*, the market coverage is maximal when commonality (*α*) is at its minimum.

Proof: Taking Corollary 1 into account, the location of the minimum (*β**) of *α* will be equivalent to the location of the maximum market coverage. Setting the first derivative of *α* with respect to *β* equal to 0 will lead to the extreme values of the commonality:

$$\frac{\partial \alpha}{\partial \beta} = 0 \qquad \text{Eq. (12)}$$

Corollary 2: The commonality (*α*) is a concave function in the range of interest.

Proof: The second derivative is greater than 0 across the whole interval (between 0 and 1). Therefore, the commonality is a concave function in the range of interest.

$$\frac{\partial^2 \alpha}{\partial^2 \beta} > 0 \qquad \text{Eq. (13)}$$

Theorem 2: *M* is a convex function in the interval between 0 and 1.

Proof: Combining corollary 1 with corollary 2 leads to the conclusion that *M* must be a convex function in the interval between 0 and 1.

Following Theorem 2, a set of closed form solutions is obtained for all reasonable values of *N*. The optimal

---

[11] Unfortunately, the assumption of constant m_i and f_i across all variants renders the sequence constraints ineffective because there is no differentiation between the variants anymore (same m_i and same f_i). Thus, the order of development does not matter.

solution is going to be the global maximum of the market coverage M^* (located at β^* of the function with the specific N value, which generates this market coverage).

3.5. Numerical Functions

To use this model for addressing real problems, it would be useful to construct approximate numeric formulations for each of the functions and factors. In this section these functions will be constructed as well as the logic they are based on.[12] An application of these numeric values will be part of the case study in Section 4.

The development cost for proposal i (consisting of V_0 and E_i) is formulated as follows:

$$D_i = V_0 E_i(\alpha, m_i) = V_0((1-\alpha)m_i a + b), \qquad \text{Eq. (14)}$$

where the constants a and b describe the portions of variable cost and fixed cost, respectively, in the development process.

An expression for the integration and testing cost is developed as follows. A variant with a distinct commonality c and possessing one interface having to adapt will make up a benchmark variant leading to a complexity factor X_i equals 1. Hence in this case, the integration and testing cost will equal A_0 (the integration and testing cost of the benchmark product). The complexity factor will rise exponentially as α drops. As a borderline case X_i will diverge towards infinity for values of α close to 0. This case would match the senseless attempt to integrate a huge peripheral set into an atomic platform. A better option might be not to use the platform approach at all. Further, the constant c will be included in the complexity function to assure that the benchmark variant on which A_0 is based has X_i equals 1. Also each interface, in addition to the first one, will increase the complexity by factor d. This leads to the following expression for X_i:

$$X_i = \frac{c}{\alpha} d^{f_i - 1} \qquad \text{Eq. (15)}$$

The flexibility factor, F, is determined by the performance of the platform. The constant e will describe the relation of β (a percentage value) to other values in absolute money terms (e.g. the cost of variants), thus influencing the behavior of F.

$$F = \frac{e}{\beta} N \qquad \text{Eq. (16)}$$

The flexibility factor is further modeled as a linear function in the number of variants N as can be seen in Equation (16). Since the factor is calculated each time a variant is created, the accumulated cost caused by the flexibility factor will behave like a quadratic function in N. The adaptation ability of the platform is also an exponentially decreasing function in β. A value of β close to 0 will give a flexibility factor diverging towards infinity.[13] Accordingly, the subsequent function for the integration and testing cost of proposal i is:

$$T_i = A_0 X_i F = A_0 \frac{c}{\alpha} d^{f_i - 1} \frac{e}{\beta} N \qquad \text{Eq. (17)}$$

The market coverage must also be expressed as a function of the explicit variants created 1 through N and their weights in the market (E_i), which is related to the performance gap when compared to integral products. As described in section about the market, the market coverage function will be a diminishing function of the summed up weighted variants. Further it will be based on the assumption that a specific number of integral products will be able to cover the whole target market segment (approximately 90%). Depending on the target segment of interest the market coverage function can be fitted to represent this case by adjusting the constant f. For example, for a market having a 90% coverage with 5 integral products ($\sum_{i=1}^{P}(V_i E_i) = 5$), f would equal 2. Afterwards the market coverage function will be used to calculate the market coverage for a modular product family. To show how the market coverage is influenced by the commonality in the product family compared to the integral products, the constant g is introduced. If not stated otherwise, g is assumed to equal 1 for all future cases, because its value does not influence our results, except the absolute value of market coverage. These properties are summarized by Equation (18).

$$M = 1 - \exp\left(-\frac{\sum_{i=1}^{P}(V_i E_i) g}{f}\right) \qquad \text{Eq. (18)}$$

Utilizing the numerical functions in Equation (10) the exact analytical solution can now be derived. This leads to a quadratic equation and thus two solutions. Only one of these two solutions is valid, because the second value is far from the allowed interval of α (between 0 and 1). The valid solution is shown in Equation (19) through Equation (22).

$$\alpha = \frac{-r - \sqrt{r^2 - 4qs}}{2q} \qquad \text{Eq. (19)}$$

$$q = -N V_0 m_i a \qquad \text{Eq. (20)}$$

$$r = N V_0 (a m_i + b) - (1-\beta) B \qquad \text{Eq. (21)}$$

$$s = N^2 A_0 c d^{f_i - 1} \frac{e}{\beta} \qquad \text{Eq. (22)}$$

If this function for α is plugged into the market coverage function, it will only be dependent on β, which provides us a closed form solution. When the first derivative of Equation (19) is set to 0, it yields Equation (23), which displays the cubic equation for β.

$$\beta^3 + \beta^2 \frac{(V_0 N(a m_i + b) - B)}{2B} - \frac{m_i a V_0 N^3 A_0 d^{f_i - 1}}{2B^2} = 0 \qquad \text{Eq. (23)}$$

---

[12] However, it is worth nothing that some variables are highly dependent on the specific product development environment.

[13] This represents the case of an extremely cheap and hence inflexible platform. That is, it would be better to invest the budget in a null platform product instead of trying to make peripheral sets of modules fit to a poorly engineered platform.

This cubic equation can be solved for β using Cardano's Formula (Merziger, 1996). This will provide us three solutions for β, of which one will be a real solution and the two other ones will be conjugated complex solutions. In this problem, the real solution only is of interest, which refers to the optimal β^*.

The solution procedure has to be repeated for each possible value of N to find the global optimum. Moreover, an average value for the number of modules m_i as well as for the number of interfaces f_i will be used because the analytical solution cannot account for the exact proposals by the engineering department (this can only be done using optimization). Nonetheless the solution will be very exact since those variables do not have high impact on the global solution, which will be shown in the case study and the sensitivity analysis.

3.6. Iterative Application

The m_i and f_i values (suggested by the engineering department) are based on a specific commonality value that the development team had in mind during the pre-design process. If the optimal commonality after running the optimization model and α of the pre-design match closely, then the choice for the size of the platform used in the pre-design process has been optimal in terms of market coverage. In this case the solution after running the model one time can be declared final. The acceptable gap between the pre-design and the optimal α is subjective. Since both α values are in terms of the percentage of modules in the product, the gap describes also the percentage of modules in the product. Arbitrarily, there is no need for iteration, if the gap is lower than half the percentage of a single module (e.g. in a product consisting of 8 modules, one module would make up for 12.5% of the product).

If the gap is larger than half the percentage of a single module, then starting a new iteration of the model will increase the accuracy of the solution and also support the conceptual redesign of the product. First the engineering department must come up with a new pre-design using a common set of modules as platform with a share of approximately α_1^* from the first run of the model. Based on this new platform the m_i and f_i values have to be reevaluated. The function diagram will serve again as the basic tool to support this step, because it displays the modular structure of the product family. When the updated values are gathered, a new set of input for the model has been obtained. The model can now be run again and a second more accurate solution can be acquired (that is, α_2^*). The iterations will be continued until the optimal commonality converges against the commonality underlying the pre-design of the previous iteration cycle. The final run of the model will arrive at the optimal solution.

4. Drill Product Family Case Study

We illustrate the model using the development of a Drill product family. The family consists of up to five cordless drills, which will be leveraged vertically in the market. To cover the whole market segment of drills, the marketing department suggests designing the following five different drills: Professional, heavy-duty, value brand, home-use, and a low-price model. The value brand drill will be used as a base variant (i.e., the other variants will be derived from it). The platform will be initially chosen using the family function diagram shown in Figure 4.

The drills on the right side of Figure 4, from top to bottom are: Professional, heavy-duty, value brand, home-use and a low-price model. The function diagram in Figure 4 reveals the modular structure of the product. The drills consist of the following 10 modules, of which one is optional: (1) Casing (including battery), (2) Contact, (3) Switch, (4) Trigger, (5) Motor, (6) Transmission, (7) Slip Clutch, (8) Speed changer (optional: only heavy-duty/professional module), (9) Drilling, and (10) Chuck.

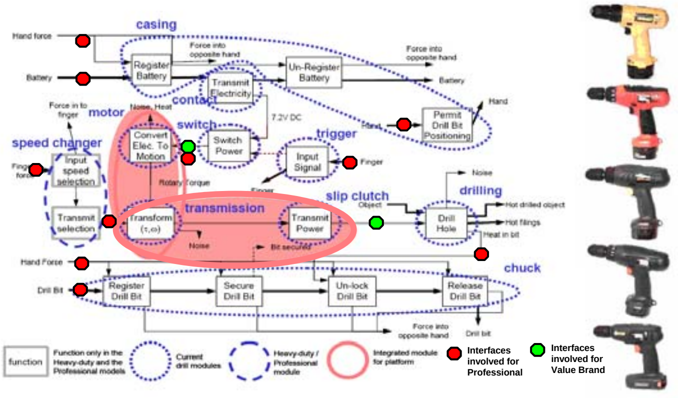


Figure 4: Drill family function structure with modules (Adapted from Otto and Holtta, 2004)

We assume that prior market research revealed that customers of drills value the external features (casing, possible functions etc.) more than the internal features (motor, transmission etc.). Based on this, the engineering department suggests developing the different drills using a platform consisting of the motor, the transmission and the slip clutch module; that is, the assumed α of the pre-design is 30% (3 out of 10 modules. These modules proposed to be on the platform are marked with bold circles in red (medium grey in black/white printout) in Figure 4. They are the most costly modules in the product and thus are greatly appropriate to make use of economies of scale later in the manufacturing process. Furthermore, this configuration leaves the external features subject to change and customization.

Management has assigned a certain amount of funds to the development process of the drills family and wants to maximize the market share in the drills market segment. There might not be enough resources to develop all 5 variants. The model will provide answers to: (a) how much of the total funds should be invested in the development of the platform, (b) which variants should be created (the total count of variants will be auxiliary result as well), and (c) the optimal commonality to be used in the product family.

4.1. Specific Problem Formulation

As mentioned earlier, the value brand represents the base variant, which will be created at first. The other variants will be derived from it. The professional and the heavy-duty drill are rather closely related as well as the home-use and the low-price model (which will probably be used mainly in households as well). Taking the changes in modules from one variant to the next into account, two development sequences are proposed:

1. Value brand → Professional → Heavy-duty
2. Value brand → Home-use → Low-price model

We assume that the variants can only be developed in the above proposed sequences. For example, creating the heavy-duty tool requires the development of the value brand as well as the professional variant; the low-price model requires the value brand and home-use variant.

Next, the inputs for the model must be specified. This includes gathering data about the company's product development (PD) department, as well as data from other reference products and further data acquired from the function structure for each of the variants. We start by figuring out the m_i (percentage of modules changed in the peripheral set) and f_i (interfaces involved in the change) values for the 5 different drills. The performance requirements of the drills determine the modules subject to change (e.g. the professional version requires an extremely advanced chuck and therefore this module must be changed compared to the value brand). If the modules subject to change are identified, then we can use the function structure to determine the interfaces having to adapt to implement these changes. Recall that each flow will be assumed to match exactly one interface in the physical structure of the product. Interfaces connecting changed modules to non-changed modules will be counted, as well as external interfaces going in and out of modules changed (e.g. the applied hand force to the case would be counted as interface if the casing is changed). There are a total of 7 modules in the peripheral set at this point. Tables 2 and 3 describe the input values for the development process when we move along both of the sequences. For example, in Table 2, the professional variant was developed based on the value brand by changing casing, contact, trigger, switch, chuck, and speed changer. Six changed modules account for 86% of the total peripheral set (consisting of 7 modules), which results in 10 interfaces involved as shown in the solid red dots in Figure 4. Similarly, the heavy-duty variation is developed based on the professional drill by changing casing, and chuck. These two modules account for 29% of the total peripheral set, which results in 6 interfaces having to adapt. Table 3 can be read the same way for sequence 2.

Sequence 1	**Modules changed**	**m_i**	**f_i**
Value brand	None	100%	2
Professional	Casing, contact, trigger, switch, chuck, speed changer	86%	10
Heavy-duty	Casing, chuck	29%	7

Table 2: Input for drill development sequence 1

Sequence 2	**Modules changed**	**m_i**	**f_i**
Value brand	None	100%	2
Home-use	Casing, contact, trigger, switch	57%	5
Low-price model	Casing, contact, chuck	43%	7

Table 3: Input for drill development sequence 2

Moreover, the funds assigned to the development process (B), the development base value (V_0), the integrating and testing base value (A_0) and the constants a through g must be determined. For this case study we assume that management allocates \$6 million to the project. The development base value depends on the development cost of a comparable integral product. In this company the development cost from the last generation of drills is known, which has not been designed using the platform approach. Since we need only one benchmark value of a single integral product, the best selling tool from the last generation is picked and its development cost will be used as V_0 after adjusting for possibly occurred internal and external changes. For example, a different workforce size employed in the PD department compared to the development of the last generation could increase (decrease) the cost due to lower (higher) efficiency. We assume this adjusted value equal to \$1,000,000.

To estimate the integrating and testing base value (A_0) a platform based benchmark product is required. In the past, this company has developed a family of small hand-held power tools (e.g. screwdrivers, sanders, etc.) for the work in narrow spaces. From documented data about the development of this product family, it is known that for a commonality of 50% (the commonality of the small hand-held power tools) the integrating and testing cost had been approximately \$40,000 if there was just one interface to be tested. These numbers are derived from the combined data of the accounting department as well as from the production units, which integrate and test the products (e.g. information about time and other resources expended to fulfill the task). Moreover, by comparing the different variants of the benchmark product family, it was ascertained that each additional interface beside the first one increases the integrating and testing effort by a factor of 1.15. Also from past empirical data of this product family, a workgroup consisting of engineers and managers derived the constant e, which relates the percentage value of β to other values in absolute money terms. When a development process of some kind is initialized in the PD department, a certain amount of fixed cost is always inherent in this decision. Over the last decade the share of fixed cost in this company turned out to be 24%, hence leading to variable development cost of 76%.

To assess the last two constants *f* and *g*, we have to take a closer look at the target market segment. Until the development of the upcoming drill product family, the company has served the target market segment quite well by offering three different integral types of drills (a standard version, an industrial version and a low-price model). Although the company was relatively successful over the past years, the integral products are out of date (market share is declining rapidly) and the customers demand new innovative products. The marketing department has information that these three products made up for coverage of around 85% of the market. A market survey has also concluded that customers do not care very much for the commonality of internal modules, which are hidden to the outer appearance of the product. This information might help to satisfy a constant *g* slightly higher than 1 in this case. Collecting the data from the above paragraph provides all needed input for the model as shown in Table 4.

Input	Value
B	\$ 6,000,000
V_0	\$ 1,000,000
A_0	\$ 40,000
a	0.76
b	0.24
c	0.50
d	1.15
e	0.30
f	1.50
g	1.05

Table 4: Input for drill family

4.2. Solution Finding and Interpretation

As mentioned earlier, we assume a constant m_i and constant f_i across all variants. In return the sequence constraints can be loosened, because there is no differentiation between the variants (same m_i and same f_i) and thus the order of development doesn't matter. The problem must be solved for each possible count of variants ranging from 1 to 5 in order to find an optimum for each *N*. Comparing these partial solutions with one another the global optimum can be obtained.

In our case the average value for the percentage changed per variant equals 0.63. The average value for the interfaces changed is 6.2. The global optimum is located at β equal 0.31. The analytical solution suggests creating all 5 variants leading to market share of 79.94%. Furthermore, the optimal α is suggested to be 54.25%. The analytical solutions for different values of *N* are plotted in Figure 5 using MATLAB. The functions are created by plugging Eq. (19) as part of E_i into Eq. (18). Then, *M* is plotted subject to β for each specific *N*.

If we take a closer look at the graph for $N = 4$ (red line or function with second highest maximum in black/white printout) and for $N = 5$, we note that the difference between both solutions is minimal. The maximum market share obtained with 4 variants is about 1% lower at about 79.05% and the commonality to obtain this optimum is approximately 33%. Since the underlying α of the pre-design is 30% (3 out of 10 modules), it would be wise to choose the maximum market share using $N = 4$ to avoid another iteration. Therefore, the company should develop 4 variants that adhere to the sequence constraints.

5. Discussion

The model is more sensitive to certain parameters than others. In this section various parameters are carefully examined. The examination of some of the output data and how they are affected by certain input variables leads to an analysis of the robustness of the proposed model.

First, we analyze the sensitivity of β^* and *N* to the commonality factor of the pre-design. The commonality underlying the pre-design indirectly influences the m_i and f_i values of the proposals, which are input to the model (the pre-design α is not a direct input to the model). Generally changing the pre-design will cause the total cost for each variant to fluctuate by a fairly low amount (e.g. if α of the pre-design is changed by a certain percentage the total cost for each variant will change by a far lower percentage)[14]. However, if an alternative pre-design leads to a different *N*, there will be a rise/drop in the commonality factor and also a change in β^* (a different *N* requires another optimal flexibility factor, which is influenced by β). In conclusion, the parameters β^* and *N* are relatively insensitive to changes in the product architecture (i.e. m_i and f_i values of proposals). The constant *e* is directly related to the flexibility factor and represents a benchmark constant for β. Nevertheless, its influence on β^* is very low. If it was not a borderline case, a change of *e* of almost 20% would be required to cause a different *N*. In summary, β^* is relatively insensitive to *e* and the total number of variants created is fairly sensitive to changes in the same constant.

One of the more sensitive parameters in this model is the commonality factor. It is calculated as a residue, because it has to balance market coverage and cost. Therefore, this value must react to any changes in the input. The commonality factor will fluctuate rather strongly, if such a small change (e.g. a change in m_i and f_i) results in a different *N*. Moreover, when the exact set of created variants is changed (*N* stays constant), the commonality factor shows a noticeable reaction as well.

---

[14] In the case study, the commonality underlying the pre-design was changed by 17% between the first and the second iteration. Not a single m_i value changed by more than 10% and none of the f_i values changed by more than 1. This change in the pre-design lead to a total cost decrease for all variants by less than 10%.

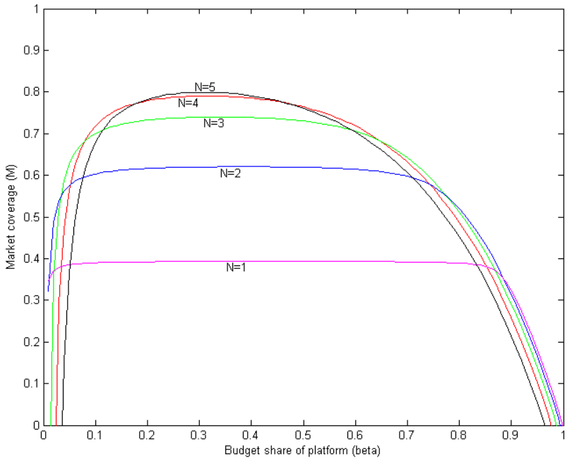


Figure 5: Analytical solution for drill family

The most sensitive parameters in this model are represented by the binary decision variables V_i. Small changes in the input, can label one variant to be more cost effective in creating market coverage than another one. Not only changes in m_i and f_i values are able to cause different V_i, but also diverse values of β. This is due to the binary nature of the problem, which is similar to a knapsack, where replacing one of the items by another one produces a slightly higher objective function value.

Finally, we address the sensitivity of the market coverage. Since this is the objective function of the problem, it reacts to any changes in any of the input parameters. Different budgets seem to have the highest impact on M. The explanation for this is reasonable. If more money is spent on the project, the resulting market coverage rises. This is independent on how this higher market coverage is gained (either by introducing more variants or by lowering the commonality). Conversely, this is also true for a lower budget.

6. Conclusion

In this paper we have introduced a model for the development of modular product families in order to maximize market coverage. The proposed model merged engineering conceptual design, development cost, marketing issues, and their interdependencies in a single model to effectively optimize the development process. The product design is based on the function diagram, which enables the illustration of module boundaries and interfaces. Commonality plays a key role in product family design, as there exist an inherent trade-off between cost saving in the development process and the distinctiveness of the variants. The more differentiated the variants are, the more market coverage they gain in the market model.

Given a set of input parameters and a conceptual engineering design, the model solves the problem by maximizing the market coverage of the product family. This is done by deciding whether to develop a certain variant out of the proposals or not, by choosing the optimal commonality factor, and by determining the share of the total budget to invest in the platform. Under certain assumptions, the model can be solved analytically (i.e. closed form). For solving more realistic scenarios, these assumptions could be relaxed, enabling the use of simulation to solve the model instead. The usefulness of the model is further expanded by an iterative application, which can serve as a guide to improve the conceptual engineering design.

The results of the model depend on assumptions which might limit its application. First, this model focuses on design and development of product families, but do not explicitly account for manufacturing costs. Second, certain assumptions were made concerning the information needed to obtain some of the input parameters regarding benchmark information. Such data might be difficult to obtain, especially for small companies. Finally, the development cost for each variant could be further refined by reducing the generalization that the size and complexity of the modules do not matter, compared to the count of modules.

References

Alizon, F., Shooter, S., and Thevenot, H., “Design Structure Matrix Flow for Improving Identification and Specification of Modules,” Proceeding of the ASME 2006 International Design Engineering Technical Conferences, Philadelphia, Pennsylvania, USA, Sept. 10-13, 2006.

Allada, V. and Lan, J., “New Modules Launch Planning for Evolving Modular Product Families”, ASME Design Engineering Technical Conference, Montreal, Canada, ASME, September 29 – October 2, 2002, Paper No. DETC/DFM-34190.

Bi, Z. M. and Zhang, W. J., “Modularity Technology in Manufacturing: Taxonomy and Issues”, International Journal for Advanced Manufacturing Technology, Vol. 18, 2001, pp. 381-390.

Browning, T. R., “Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions”, IEEE Transactions on Engineering Management, Vol. 48, No. 3, 2001.

de Weck, O. Suh, E.S., “Product Family and Platform Portfolio Optimization”, ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference, Chicago, IL, ASME, September 2-6, 2003, Paper No. DETC2003/DAC-48721.

Fellini, R., Kokkolaras, M., Papalambros, P. and Perez-Duarte, A., “Platform Selection under Performance Loss Constraints in Optimal Design of Product Families”, ASME Design Engineering Technical Conference – Design Automation Conference, Montreal, Canada, ASME, September 29 – October 2, 2002, Paper No. DETC2002/DAC-34099.

Fellini, R., Papalambros, P. and Weber, T., "Application of a Product Platform Design Process to Automotive Powertrains", 8th AIAA/USAF/NASSA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, 2000, Paper No. AIAA-2000-4849.

Gonzalez-Zugasti, J., Otto, K., "Modular Platform-Based Product Family Design," Proceedings of DETC'00, ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Baltimore, Maryland, September 10-13, 2000.

Gonzalez-Zugasti, J. P., Otto, K. N. and Baker, J. D., "Assessing Value in Platformed Product Family Design", Research in Engineering Design, Vol. 13, No. 1, 2001, pp. 30-41.

Holtta, K. and Otto K., "Incorporating Design Complexity Measures in Architectural Assessment," In Proc of ASME Design Engineering Technical Conferences. Chicago, IL. September 2-6, 2003.

Hsuan, J., "Modularization in New Product Development: A Mathematical Modeling Approach", DRUID Summer Conference on National Innovation Systems, Industrial Dynamics and Innovation Policy, Rebild, Denmark, June 9-12, 1999.

Jiao, J., Simpson, T. W. and Siddique, Z., "Product Family Design and Platform-Based Product Development: A State-of-the-Art Review", Journal of Intelligent Manufacturing, Special Issue on Product Family Design and Platform-Based Product Development, 2006.

Kim, K., Chhajed, D., "Commonality in product design: Cost saving, valuation change and cannibalization," *European Journal of Operational Research*, 125(3), 2000, pp. 602-621.

Martin, M.V., Ishii, K., 2002, Design for Variety: Developing standardized and modularized product platform architectures, Research in Engineering Design, 13(4): 213- 235.

Mark, G. T., Saleh, J. H. and Feron, E., "Flexible Platforms: Reducing the Performance Gap", ASME 2005 International Conference on Design Theory and Methodology, Long Beach, CA, ASME September 24-28, 2005, Paper No. DETC2005-85434.

Merziger, G., Formeln und Hilfen zur Höheren Mathematik, 2nd Edition, Binomi Verlag, Springe, Germany, 1996.

Messac, A., Martinez, M. P. and Simpson, T. W., "Introduction of a Product Family Penalty Function Using Physical Programming", Journal of Mechanical Design, Vol .124, 2002, pp.164-172.

Michalek, J., Ceryan, O., Papalambros, P., Koren, Y., "Balancing Marketing and Manufacturing Objectives in Product Line Design," ASME Journal of Mechanical Design, 128 (11) 2006, pp. 1196-1204.

Otto, K. and Holtta, K., "A Multi-Criteria Framework for Screening Preliminary Product Platform Concepts", In Proc of ASME Design Engineering Technical Conferences, Salt Lake City, UT, September 28- October 2, 2004.

Otto, K. and Wood, K., Product Design: Techniques in Reverse Engineering and New Product Development, Prentice Hall, Upper Saddle River, NJ, 2001.

Pahl, G. and Beitz, W., Engineering Design: A Systematic Approach, 2nd Edition, Springer, Berlin Heidelberg New York, 1996.

Park, J. and Simpson, T. W., "Production Cost Modeling to Support Product Family Design Optimization", ASME 2003 Design Engineering Technical Conference and Computers and Information in Engineering Conference, Chicago, IL, ASME, September 2-6, 2003, Paper No. DETC2003/DAC-48720.

Pimmler, T., and Eppinger, S.D., "Integration Analysis of Product Decompositions," DE-Vol. 68, Design Theory and Methodology- DTM 94, ASME 1994.

Simpson, T. W., Maier, J. R. A. and Mistree, F., "Product Platform Design: Method and Application", Research in Engineering Design, Vol. 13, No. 1, 2001, pp. 2-22.

Simpson, T., Siddique, Z., Jiao, J., "Platform-Based Product Family Development", Product Platform and Product Family Design: Methods and Applications, T. W. Simpson, Z. Siddique and J. Jiao, eds., Springer, New York, pp. 1-15. 2006.

Stone, R., Wood, K., and Crawford, R., "A Heuristic Method for Identifying Modules for Product Architectures," Design Studies, 21(1), 5-31, 2000.

Ulrich, K. T. and Eppinger, S. D., Product Design and Development, 2nd Edition, McGraw-Hill, New York, 2000.

Vollerthun, A., "Design-to-Market: Integrating Conceptual Design and Marketing", System Engineering, Vol. 5, No. 4, 2002, pp. 315-326.

Yigit, A. S. and Allahverdi, A., "Optimal Selection of Module Instances for Modular Products in Reconfigurable Manufacturing Systems", International Journal of Production Research, Vol. 41, No. 17, 2003, pp. 4063-4074.

Yu, T.-L., Yassine, A., and Goldberg, D. E., "A Genetic Algorithm for Developing Modular Product Architectures," Proceedings of the ASME 2003 International Design Engineering Technical Conferences, DTM-48657, 2003.

Zacharias, N. A., "Optimizing Product Platform Investment Decisions for Market Coverage: A Formal Model for Product Family Design", Thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2006.