Publication VI

Otto, K. & 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.

Proceedings of DETC'04
ASME 2004 Design Engineering Technical Conferences and Computers and Information in Engineering Conference
September 28-October 2, 2004, Salt Lake City, Utah USA

DETC2004-57256

A MULTI-CRITERIA FRAMEWORK FOR SCREENING PRELIMINARY PRODUCT PLATFORM CONCEPTS

Kevin Otto
Product Genesis Inc.
245 Bent St
Cambridge, Massachusetts 02141
Tel +1 617 234 0070
Kevin_Otto@ProductGenesis.com

Katja Hölttä
Center for Innovation in Product Development
Massachusetts Institute of Technology
30 Memorial Dr., RM E60-246
Cambridge, Massachusetts 02139
and
Helsinki University of Technology, Finland

ABSTRACT

A platform must effectively support multiple product variants over a prolonged period of time. This makes platform concept evaluation a more challenging task than a single product concept evaluation. Existing platform methods develop specific criteria in depth, yet an evaluation of alternative platforms should be based on a broad set of criteria. Based on expert interviews, personal experience, and a literature search we propose here a framework of 19 metrics for multi-criteria platform evaluation. The metrics are group into six categories: customer, variety, flexibility, complexity, organization, and after-sale. The method is focused on the early platform architecture phase, before proof-of-concept prototyping. However, it can equally be used subsequently for platform refinement when more data becomes available. We demonstrate our framework using a cordless drill family as an exemplary platform.

KEYWORDS: platform, product architecture, concept evaluation

INTRODUCTION

In today's markets it is important to be able to launch new products quickly and reliably while keeping costs low to be competitive. Platforms are one solution to both demands. When a company has a good product platform, it is able to launch new product variants more quickly than developing completely new products, and with a greater chance of success. For example, the Association of National Advertisers found that 27% of product line extensions fail; whereas 31% of new products introduced into existing categories fail; and a very high 46% of new products introduced in new categories fail [1]. Clearly, creating product variants for a refined market is a simpler road to profits than completely new products. Sharing modules among product variants can also bring cost savings both in terms of material scale and scope [2], as well as engineering costs, since the same design is used in more than one product [3][28].

Yet, we find an apparent lack of comprehensive *platform concept* evaluation tools. There has been excellent work in developing product concept evaluation methods, such as Pugh's selection process, concept screening and scoring, or trade-studies [23][33]. However, these methods are for evaluating a single product concept. A platform concept has different requirements due to its longer lifetime and that it must enable several derivative products. These added requirements make the single product concept evaluation methods not directly applicable for a platform concept. We will improve on these existing methods to include the platform specific requirements in the analysis.

Some methods exist to manage numerical performance trade-offs in architectural development of platforms. Otto and Wood [23] develop platform based Pugh charts, to permit a Pugh screening process on multiple concepts. Gonzalez-Zugasti et al introduced an optimization formulation for shared platform, given supported product variants to be optimized [14]. Fujita et al [12] present a similar numerical optimization approach. Fellini [10] as well as Nelson et al [21] use pareto fronts to decide on the degree of commonality between products in a product family. Also de Weck and Chang use pareto fronts to decide between product performance and its life cycle cost when choosing an optimal architecture [8].

Table 1 Platform evaluation metrics for current drill platform and for alternatives A and B.

Metric		W	Score		
			Current	A	B
Complexity	Function and form	3.3%	**9.7**	9.6	9.5
	Redesign complexity	3.3%	**10**	10	10
	Anti-synergies	3.3%	**0**	0	0
	1 DOF adjustment	3.3%	**10**	10	10
	Challenge extremes	3.3%	**6.6**	6.6	6.6
Customer	Value add	8.3%	**8.0**	7.6	7.2
	Requirements	8.3%	**7.3**	7.3	8.8
Flexibility	Isolate unknowns	8.3%	**10**	10	10
	Flexibility to expected change	8.3%	**9.1**	9.0	6.4

Metric		W	Score		
			Current	A	B
Organization	Assembly ease	4.2%	**4.6**	5.0	4.5
	Drive the organization	4.2%	**5.4**	4.6	1.5
	Make-buy	4.2%	**9.6**	8.8	8.6
	Testing	4.2%	**5.4**	5.5	5.7
Variety	Carryover	5.5%	**9.7**	9.7	8.3
	Common unit	5.5%	**5.9**	5.9	5.4
	Different specification	5.5%	**10**	10	10
After Sale	Reliability	5.5%	**8.6**	8.5	7.9
	Service	5.5%	**10**	10	10
	Environmental friendliness	5.5%	**8.1**	8.1	8.1

Complexity Score	**7.3**	7.2	7.2	Organization Score	**6.3**	6.0	5.1
Customer Score	**7.7**	7.5	8.0	Variety Score	**8.5**	8.5	7.9
Flexibility Score	**9.6**	9.5	8.2	After Sale Score	**8.9**	8.9	8.7
Platform Score					**8.0**	**7.9**	**7.5**

Nayak, on the other hand, defines platforms based on minimizing the variations of corresponding design variables in different products of a product family [20]. Siddique and Rosen describe a method to define a platform based on the existing set of components and assembly sequences of the current set of products [27].

Generally, these works restrict to developing a couple focused criteria when evaluating platforms. The main focus seems to be on maximizing the commonality while trying to maintain the product performance requirements. Yet, when using only a few criteria to develop or evaluate platforms, one must not ignore the others. For example, one may have optimized the performance and cost of the platform but is it more reliable than another? Does it have lowest service cost? Is it easy to outsource major subsets? Is it more flexible than another platform? etc. This issue often arises e.g. when comparing two alternative platform concepts, or deciding whether to update or replace a platform.

In this paper, we introduce a platform concept evaluation scoring framework and approach that is multi-criteria in nature, and scalable to include various alternative criteria as appropriate. Several criteria, primarily as developed in the literature but also elsewhere, will be applied here to demonstrate a platform concept evaluation, considering aspects of the product from its entire life cycle. Some of the metrics drive toward more a modular architecture and some more toward an integral architecture. This multi-criteria analysis results in a concept phase analysis that helps manage risk by making all aware of the criteria that a development project may need backup plans developed, extra effort applied, and management attention.

Our approach

We developed a scoring framework that makes use of the work of many others in the field of modularity, platforming, and general product development. We generated a list of platform metrics from three sources. First, we generated a list based upon personal experiences of platform development over the last 10 years with over two-dozen platforms and the personal mistakes learned from inadequate preparation (e.g., inadequate preliminary assessment). Second, we consulted six other executive-level system engineers with an average of 17 years experience, and solicited platform metrics based upon examination of their past platforms. Lastly, we examined the literature for platform metrics used by others, such as by Ericsson and Erixon [9], Blackenfelt [5].

Interestingly, personal experiences generated several metrics not found in the literature. However, where metrics were available, they were generally well developed and researched, and therefore provided the best results and were easily adaptable to our framework. We rescale their results into a standard scale where they can then be compared. The approach is general to any metric as desired to be included. We here take into account more factors than methods developed so far. This scope is more realistic to industrial practice, where platform decisions must consider a multitude of decision impacts. Here, we attempt to include these factors into a broad decision framework.

Based upon previous literature and our experience, we have grouped and assembled a total of 19 metrics, further refined into six groups, as shown in Table 1. One should notice that this list may not be appropriate for all industries

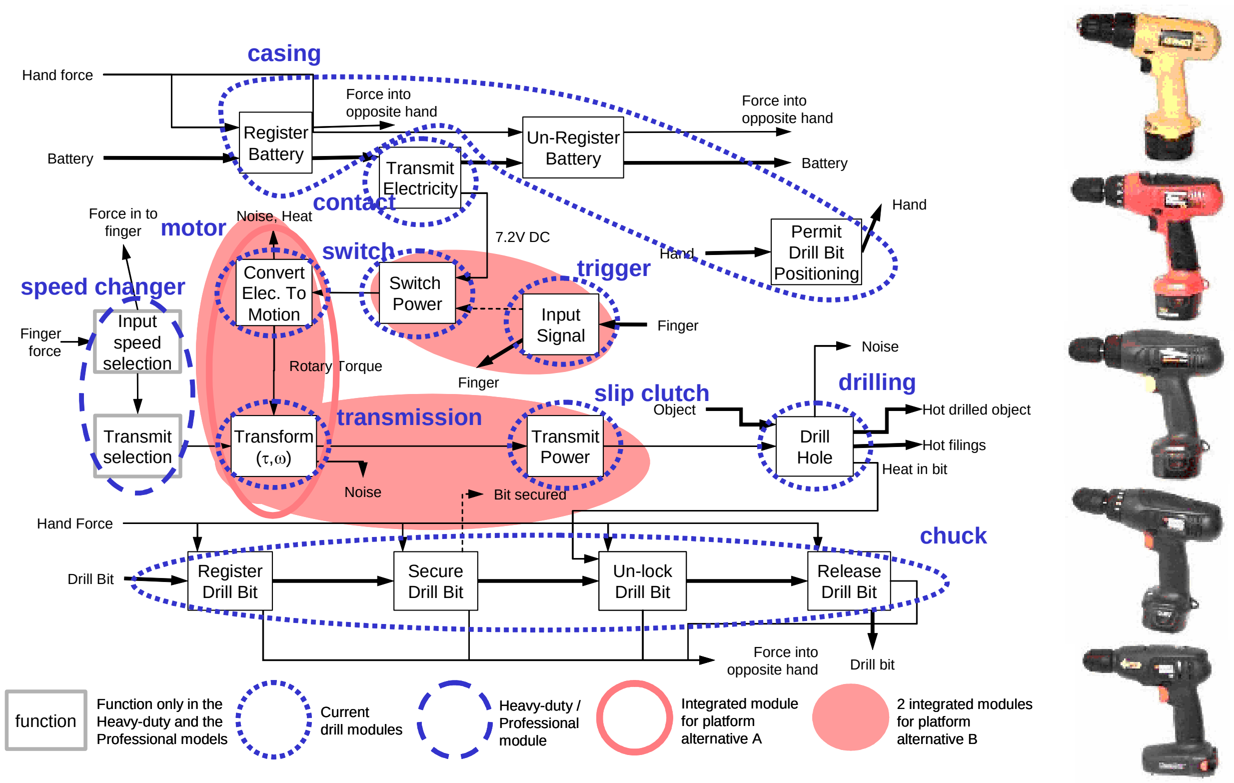


Figure 1 Case study family of drills and their family function structure, with modules shown.

and should be adjusted when needed. We do not claim comprehensiveness, we do claim these 19 serve well many electro-mechanical platforms.

To calculate an over all platform score from all the 19 metrics, we project each metric onto a calibrated interval scale. We define a merit scale using {0, 3, 5, 7, 10}, where 0 is the worst and 10 the best. This is analogous to an A-F grading scale. We assign each metric such a score from our scale. Within each metric we use more rigorous numeric scales when possible. Also, a metric may sometimes consist of multiple components. Generally, we combine these into a single score using a simple weighted sum. We also use a weighted sum to derive the overall platform score from the individual metric scores. We derive the weights based upon individual marginal contribution to long-term corporate profit. One could also extend this to more refined decision-making methods such as Analytic Hierarchy Process [26] or Utility Theory [24]. We chose a simpler approach since our aim is not to develop an absolute preference scale, but more simply to determine relative sensitivities – to point out where issues exist in individual alternative platform concepts.

We focus our method at the platform architecture concept phase, where detailed information needed for all metrics is generally not completely available. We suggest using what data is available and estimate the rest. Then, one can explore the sensitivity of the analysis to changes in the uncertain data entries. Further, as more detailed data becomes available, this evaluation can be recalculated with the refined data as needed.

After our assessment of each metric, the values are compiled into Table 1. The individual scores are combined to determine category assessments, which are then combined to indicate an overall platform assessment score. While the overall score is a rough estimate, the value of the work is in the sub-scores. They can be used for to relatively assess platform concepts that seem attractive, to understand what areas need improvement or can cause future problems. These areas of weakness should then be focus areas in subsequent platform development.

It is important to perform this platform assessment on past and competitive platforms to understand the industry's state of the art. For example, it might be that a 10 is not necessary with the current state of technology or market. Benchmarking provides the reference points needed to provide a well-defined maximum to aim for.

The platform example

We demonstrate our method and the use of various research groups' metrics through an example of a family of five different cordless drills: professional, heavy-duty, value brand, home-use, and a low price model. We use the term *platform* here to mean the common set of functions from which multiple products can be derived. In this case the platform architecture is the same as the family function structure since all functions are common between at least two members of the product family. Shown in Fig 1 is the family function structure for the cordless drills with its current modules [30]. We label each module according to its main component. Note that module "drilling" is a function performed by the user and not by the product. It is included here to keep the modularization

Function and form	w_1	ideal	casing	contact	switch	motor	trans-mission	slip clutch	drilling	trigger	chuck	speed changer
#parts in more than one module	1	0	0	0	0	0	0	0	-	0	0	0
# separate locations with the module's parts	1	1	2	1	1	1	1	1	-	1	1	1
Total (w_1#parts+w_2#locations)		1	2	1	1	1	1	1	-	1	1	1
Normalized		10	7	10	10	10	10	10	-	10	10	10

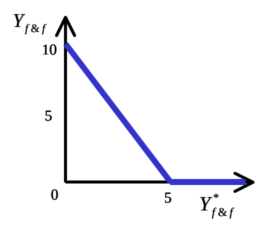


Figure 2 Evaluation of Function-Form alignment for the drill platform.

same with the previous literature and also to show that not all functions in the architecture have to be realized by the product. We also exclude the battery pack from the analysis, which is a separate product. The speed changer module is found only in the professional and heavy-duty models. The entire family function structure consists of only two distinct product function structures (merged in Fig 1), and the added variety is obtained based in how the functions are fulfilled. In addition we show two alternative modularization choices (A and B) in Fig 1. Alternative A is otherwise same as the current platform except that the motor and transmission have been combined into a module. In alternative B, the clutch is further added to the motor-transmission module and the switch and trigger modules are combined into a single module.

We will evaluate here one platform – the current cordless drill family – as an example. In addition we will show results for platform alternatives A and B. We intend that this framework will be used to evaluate multiple alternative platforms. We show here an integrated approach, but in practice many of the metrics could be done separately for the mechanical, electrical, and software parts of the platform. Our framework will help determine and clarify what criteria a platform is strong and what criteria a platform is weak. This helps manage risk by making all aware what factors a development project may need backup plans, extra effort, and management attention.

PLATFORM EVALUATION METRICS

In this section we describe all individual metrics that will then be compiled in the platform evaluation table (Table 1).

Complexity

The metrics in the first group, complexity, aim at reducing the apparent complexity of the architecture and the development. We include 4 metrics: *function and form alignment*, *redesign complexity*, *anti-synergy management*, and *alignment attachment distinction.*

Function and form

A module should have a clear function and each function should be a clear module. This is supported, for example, by Ulrich in his definition of modular architecture as a one-to-one mapping from functional elements to the physical components of the product [33], and Suh's philosophies of axiomatic design [31]. In a good architecture, a function is not distributed across several modules, since it can cause excess design communication and may result in assembly problems. If a function must be shared between modules, it should be driven by large gains on constraints: cost, volume, or mass requirements. Also Whitney supports this thinking [34].

Yet, to our knowledge, these thoughts have not been quantified into a metric that specifically states the degree of function-form independence. The metric we use is two-part. We penalize an architecture that has parts that are in more than one module – such as a photocopier with a belt, where the belt is part of both the ink transfer module and the image transfer module; and a module whose parts are in more than one separate location - such as an elevator, where the control module has a user interface on the cab and a motor controller in the machine room. Both of these architectures cause the design teams difficulty of required coordination. The metric for each module is calculated as follows:

$$Y^*_{f\&f} = w_1 \cdot (\#\text{parts in more than one module}) + w_2 \cdot (\#\text{separate locations with the module's parts})$$

where

w_1 is the difficulty weight of having parts in more than one module

w_2 is the difficultly weight of having a module's parts in separate locations

As an example, consider the cordless drill family. In general this metric is calculated for each product and the scores are then averaged, using the profit contributions of each product, to get the platform score. We show the result for the family in Fig 2. The scores are normalized according to graph. The score for the cordless drill family architecture is the average of the modules' scores: 9.7, indicating the drill platform is well functionally partitioned. This is intuitive with a teardown of the drill, each function is contained in a module.

Redesign complexity

Modules attached to the platform and the platform itself will have to change to adapt to new, often unexpected, requirements. In order for the design changes to be as easy as possible, every module should be as isolated as possible so that if it changes, the change has only a minimal effect to the other modules. This is equally valid for software [3]. One way of ensuring this is to keep the module interfaces a simple as possible from the design point of view. Hölttä and Otto [16] developed a metric for this purpose. They evaluated the redesign effort of different interaction types. The following table (Table 2) lists the weights of each interaction type.

Table 2 Redesign complexity weights.

Interaction type	Redesign complexity factor
Material flow	1.1
Acoustic energy	3.8
Electrical energy	1.2
Mechanical energy (rot)	1.7
Pneumatic energy	3.2
Thermal energy	2.2
Signal	1.3

The platform redesign complexity score is calculated from the family function structure as follows:

$$Y_{re-d} = 10 \min\left(1, \frac{2(\#\ \text{intermodule interfaces})}{\sum_{\text{intermodule interfaces}} \text{redesign complexity weights}} \right)$$

We develop this metric with the idea that any interface is sufficiently simplified when it has a complexity factor of 2 or less from Table 2. In our example, we calculate a platform score of 10 for the drill family, indicating that the interfaces are all indeed sufficiently simple for design purposes.

Anti-Synergies

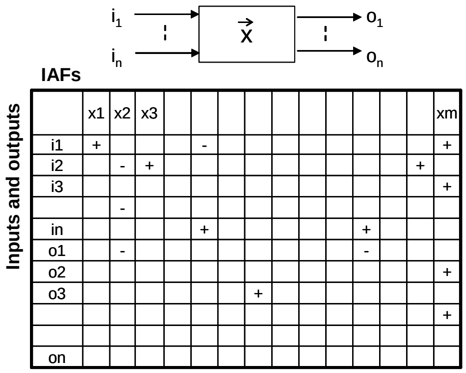
As system size grows, stable linear systems often become highly non-linear. This is because the anti-synergies of the multiple flows through each interface combine into an often-unexpected response. A way to control these unexpected interactions is to have a design controllable factor at each interface to adjust each flow response going through it. We call each of these design controllable factors an *interface adjustment factor* (IAF). Generally, the problem is that multiple interface flows exists. Clearly, one would like one IAF for each flow, and each IAF ideally influences only one flow and will not impact the remaining interface flows.

Independence is a tall order for an interface design, generally resulting in bulky impractical interfaces. More realistically, when adjusting an IAF to match modules to the various product variants or applications, the flows that are affected by the IAF should *as a set* improve or worsen monotonically. The module is then much simpler to adjust to applications. We find this equally applies to software products. Bass mentions that in order for software to be modifiable the functions should be designed to avoid a ripple effect i.e. a situation where a change indirectly affects another module [3].

We define our metric based on the interactions between the input and output flows of a module and their IAFs. We use a matrix representation, where all the flows in and out of a module are represented with rows and IAFs are represented as columns, and matrix entries are (+)'s and (-)'s to indicate the sign of the change of the flow with an increase in the IAF. As an IAF is changed, the change direction of each impacted flow should be same. That is, for any IAF generally, an ideal design is one where the sign of the sensitivity of every flow is the same.

$$Y^*_{IAF} = \frac{\#\ \text{flows improving}}{\#\ \text{flows}}\Bigg|_{\text{as IAF increases}}$$

Therefore, ideally every column has either all (+) or all (-) entries (in addition to blank entries for the flows an IAF does not impact), as shown in Fig 3. We transform this metric Y^*_{IAF} into a 1-10 scale for a module (Fig 3), where we first penalize having no IAFs for an interface flow, and only then penalize for non-synergistic IAFs. The score for the summary platform scorecard is the average of the module scores.



	x1	x2	x3											xm
i1	+				-									+
i2		-	+										+	
i3														+
		-												
in					+						+			
o1		-									-			
o2														+
o3							+							
														+
on														

$$Y_{IAF} = \begin{cases} 0 & \text{no IAF} \\ 3 & \text{IAF, but } Y_{IAF}\text{'s are } 50-50 \text{ anti-symetric with others in module} \\ 5 & \text{IAF, but } Y_{IAF}\text{'s are } 75-25 \text{ anti-symetric with others in module} \\ 10 & \text{IAF with no anti-symetries with other flows in module} \end{cases}$$

Figure 3 Adjustment matrix.

In the cordless drill platform example, the platform does not score high on this metric, as the platform was more focused on cost targets. *None* of the products in the cordless drill family have IAFs available to permit simplified changes or modifications. All power carrying modules are outsourced and if there is a change, *every* adjacent components must be changed. Therefore, the anti-synergy score for the cordless drill platform is 0.

1 DOF adjustments

This metric is relevant only to architectures of products that require service or swap-outs of modules. This is common in long service life mechanical systems, where some modules are serviced. As a part of the service procedures, sometimes the serviced or replaced module must be adjusted to fit well – it must be inserted, measured, and fine-adjusted for a flow or performance metric to be placed on a desired target. Examples include kinematic positioning, current flows with resistor adjustment, acoustic noise or frequency adjustments.

For any such serviced module, the ideal number of adjustments is zero, the module can simple be inserted and the insertion automatically aligns the module, followed by an attaching operation such as a snap fit or non-aligning attachment screw. For modules that do require precise adjustment, a good interface should have only one adjustable degree of freedom (DOF). More degrees of freedom make the adjustment a difficult and iterative process. In addition, a good interface should always have an indicator directly visible while making the adjustment. For example a current flow gauge should be visible next to a variable resistor, or a bubble indicator should be visible next to a screw positioner.

The score is calculated for each module in the platform that requires periodic service replacements. For any serviced replacement module, the metric for each module is calculated as follows:

$$Y_{DOF} = \begin{cases} 10 & \text{No adjustments} \\ 7 & \text{1 DOF to align with 1 adjustment with 1 indicator} \\ 1 & \text{2 DOF to align with 2 sequenced adjustments with indicators} \\ 0 & \text{Anything else} \end{cases}$$

All of serviced replacement modules in the platform are averaged to get the overall platform score. The cordless drill family has only one replacement module, the battery, which has no adjustments required, and so the platform scores a 10.

Challenge extremes

Requirements that are difficult to meet cause problems. An architecture driven by a few extreme requirements forces poor performance on the other requirements. Thus one should challenge extreme requirements throughout the development process, and validate them early and continuously.

For this metric we compare the new requirements of the architecture under development to the requirements of the current model in the market. The metric for a product variant in the portfolio is calculated as follows:

$$Y_{ex} = \min\left(10 - \max\left(0, \frac{\text{new req}_i - \text{standard req}_i}{\text{standard req}_i} \cdot \text{difficulty}_i\right)\right)$$

where

new req_i is the requirement level *i* for the product under development

standard req_i is the requirement level for requirement *i* that is known to be achievable

difficulty_i is the difficulty of achieving the new requirement *i* (scale 1-10)

Below are the values for some of the requirements and estimated design difficulties for the home-use drill (Table 3). The value for the platform is the minimum of the supported-product-variant scores: here 6.6.

Table 3 Home-use drill requirements and analysis of extremes.

Customer needs	Standard Reqs	New Reqs	Change %	Difficulty	Score
Powerful	105 in-lbs	200 in-lbs	90	2	8,2
Fast	800 rpm	800 rpm	0	3	10,0
Variable velocity	0-800 rpm	0-800 rpm	0	3	10,0
Portable (cordless)	9.6V bat	9.6V bat	0	5	10,0
Quiet	70dB	65dB	68	5	6,6
Affordable	$70	$75	-7	7	10

Customer

Meeting the customer needs is the primary goal of any product. This includes checking that each product *adds value* and meets the *customer requirements*.

Value Add

All modules should add value to the product or they should not be included in a platform, as argued by Baldwin and Clark [2]. Translating into an actionable concept, each module should have an identifiable function whose customer value is known.

To capture this, we apply the cost-worth approach described in [32]. We calculate the cost and worth of each module, and evaluate based upon the cost-worth difference. Ideally the cost to worth ratio is one. If larger, the module's costs should be reduced; if lower, there is room for improvement in a module since a customer values it more that what it is worth.

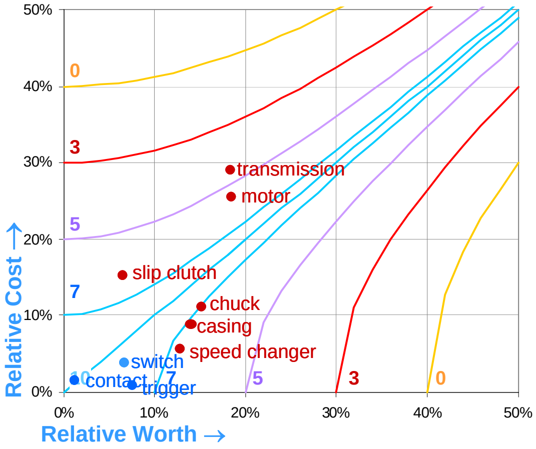


Figure 4 Relative cost worth diagram for a heavy-duty model cordless drill.

To form an interval scored metric from the standard cost worth graph, we augment it with gradations and then score based on how far a module is from the ideal, as shown in Fig 4. For example, the motor module has a score of 7 and the switch 10. The total score for the family architecture is then the average of the products' modules' scores, generating an average measure of the deviation from ideal. For the heavy-duty model brand drill we calculate a value analysis result of 7.8. This reflects what can be seen in the figure as well, that most of the modules are just outside the ideal zone and a little improvement could raise the score closer to 10. For the entire drill family we calculate 8.0.

Requirements

This metric measures how well the customer requirements are met by the platform. In fact, this is one of the most important metrics; failure here implies the platform's product variants will not sell to the known market segments the company is currently targeting. This metric includes product performance among other customer requirements. We organize our metric as a hierarchy of the product variants' requirements. We compare the product variant's ideal target on each requirement to what the platform can actually provide. This should be done in reference to competitive benchmarks, such as when using the standard methods of Quality Function Deployment.

A metric scaled to our scale is given by

$$Y_{CR} = 10 \sum_{\text{products}} w_i \left(\frac{1}{M} \sum_{\text{Requirement}} \left(y - \tau \right)^2 \right)^{1/2}$$

where

w_i is the revenue weighted importance of product i

M is the number of requirements

y is the requirement level provided by the platform

τ is the desired target for the requirement for the product variant

For the drill platform, the customer need assessment was made drill by drill against the intended market requirements, as assessed by third party commercial consumer groups. The drill platform assessed reasonably at 7.3, with the weakest scores falling under the battery run-time and purchase price, and the strongest scores falling under the weight and charge time.

Flexibility to Change

A platform will have to adapt to an ever-changing environment. Flexibility is one of the key features of a good platform and the following 2 metrics help determine whether a platform is flexible or not: *isolate uncertain requirements*, and *flexibility to expected change.*

Isolate uncertain requirements

Uncertain functionality in a system causes unexpected architectural rigidity, difficulties, and failure. To prepare for these unexpected effects one should isolate all unknowns into modules. To evaluate how well the architecture can accommodate requirements changes, the following steps should be performed: (1) Identify the requirements which are subject to change; (2) Identify which flows adjust in range to meet the range of requirement changes; (3) For each interface, identify the adjustment factor(s) that are free to vary to create the flow range, yet still fit the interface; (4) Determine the domain of the IAF operability; (5) Ensure the IAF domain provides the product flexibility to cover the range of requirements. This should be done for each module interface.

A metric for this concern is

$$Y_{IU} = \frac{10}{\#\,\text{modules}} \sum_{\substack{\text{interfaces}\\ \text{to uncertain}\\ \text{modules}}} \frac{\text{adjustment range}}{\text{required range}},$$

which is evaluated only on the modules that have uncertain flows through them. This result can then be scaled by the fraction of modules that are so classified. That is, the drill family has no unknown functions, therefore the drill platform score is 10.

Flexibility to expected change

A platform must support several product variants at any point in time. Technology evolves, there might be planned upgrades, etc. and the platform must accommodate for all these [5][9]. Rather than requirement driven upgrades as in the previous metric, this metric considers upgrades to component technology.

Martin and Ishii [19] have developed a generational variety index (GVI) to estimate the relative redesign difficulty of components by predicting possible change scenarios. Rajan et al [25] developed also a metric that is based on possible change scenarios. They identify potential change modes (scenarios), estimate the readiness of the company to deal with the change as well as flexibility of the product. In addition the estimate how often or how likely the change is to occur. They combine these four factors into a metric that we will make use of here.

$$CPN = \frac{10}{N} \sum_{i=1}^{N} \frac{(R_i + F_i) - O_i + 8}{27},$$

where

CPN = Change potential number

N =Max of (number of change modes, number of potential effects of change, number of causes of change)

R =Readiness (Readiness 1-10, 10 being completely prepared)

F =Flexibility (level of redesign effort 1-10, 10 is no redesign, 1 is new product)

O =Occurrence (Probability of occurrence, #times in every 10 yrs)

The CPN is calculated for each product, which are then averaged to get the platform score.

The drills have potential change modes such as the motor size changing, which causes redesign of the casing. We estimated all change modes, their effects, causes, and the readiness and flexibility of the company, and the occurrence of the changes. We calculate 9.1 as the upgrade flexibility score for the cordless drill family. The high score is mainly due to the low occurrence of changes and low redesign since most parts are available in a range of standard outsourced sizes.

Organization

The metrics in the next group assess how well the platform helps organize the development of the products in question and improves the development and production. These metrics include *assembly ease, drive the organization, make-buy,* and *testing*.

Assembly Ease

For assembly, we have a two-part metric. Both parts aim to minimize the assembly time. The first metric can be applied very early in the concept design phase with very little information. The second metric can be applied later, when more geometry and assembly sequence information becomes available. Only one metric should be used, depending on the data available.

The first metric was developed by Ericsson and Erixon [9] who showed that the assembly time of a product is minimized if the product consists of $K\sqrt{n}$ modules, where n is the number of parts and K is between 1 and 1.5, as fit to a particular company's assembly plant data. Fewer modules prevent parallel assembly and more modules take too long in the final assembly. This can be recursively applied for sub-assemblies and sub-sub-assemblies. In our scaled metric form, this becomes:

$$Y_{\text{assy 1}} = 10\min\left(\frac{K\sqrt{n}}{\#\,\text{modules}}, \frac{\#\,\text{modules}}{K\sqrt{n}}\right)$$

For the cordless drill platform, we calculate a perfect score 10. This indicates that the number of modules is well determined.

The second assembly metric is to simply use Boothroyd and Dewhurst [6] design-for-assembly (DFA) design efficiency metric. Boothroyd claims the assembly time for an ideal module is 3 seconds, when the module's parts are minimally complex to assemble and handle. An assembly metric for any module can then be calculated as:

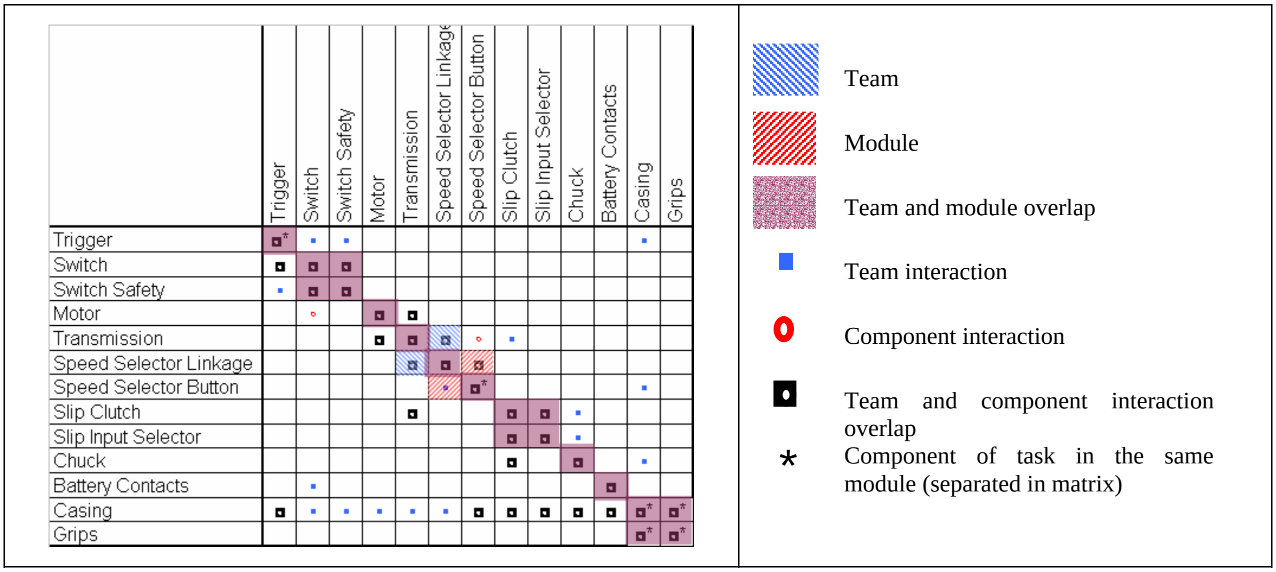
$$Y_{\text{assy 2}} = \frac{3\,n}{\text{assembly time}}$$

where n is the number of modules in a product. This design assembly efficiency can then be calculated for each module as used in each product in the platform, and each transformed to a 1 to 10 score.

While Boothroyd's design efficiency metric is theoretically well developed, the score itself is very aggressive and difficult to achieve. Therefore, the metric should be scaled based upon past assessments of design efficiencies at the targeted company. Very often designs with very low assembly cost can still have Boothroyd design efficiencies in the 20-30% range. After scaling against a reasonable standard, the platform score is determined as the average score over the modules, as weighted by the sales of product variants that make use of modules not common to all the variants. As an example, we calculated the Boothroyd assembly design efficiency score for the cordless drill family and got a low 4.6. The score shows that the drill is not effectively designed from a purely assembly theoretic point of view. The casing and motor modules drove low scores, since both include multiple screwing operations. Further, improved alignment features could aid in the assembly.

Drive the organization

For the short-term platform development to go smoothly, the organization structure should be one to one with the product modules. This reduces internal errors during development. More often an organization exists and thereby the organizational and supplier boundaries drive the product partitioning, not vice versa. This view is supported by Sosa et al [29].



	Trigger	Switch	Switch Safety	Motor	Transmission	Speed Selector Linkage	Speed Selector Button	Slip Clutch	Slip Input Selector	Chuck	Battery Contacts	Casing	Grips
Trigger	◘*	▪	▪									▪	
Switch	◘	◘	◘										
Switch Safety	▪	◘	◘										
Motor		○		◘	◘								
Transmission				◘	◘	◘	○	▪					
Speed Selector Linkage					◘	◘	◘						
Speed Selector Button						▪	◘*					▪	
Slip Clutch					◘			◘	◘	▪			
Slip Input Selector								◘	◘	▪			
Chuck								◘		◘		▪	
Battery Contacts		▪									◘		
Casing	◘	▪	▪	▪	▪	▪	◘	◘	◘	◘	◘	◘*	◘*
Grips												◘*	◘*

▨ Team
▨ Module
▩ Team and module overlap
▪ Team interaction
○ Component interaction
◘ Team and component interaction overlap
* Component of task in the same module (separated in matrix)

Figure 5 Combined component and team interaction DSM.

Make-buy	ideal	casing	contact	switch	motor	trans-mission	slip clutch	drilling	trigger	chuck	speed changer
#out-sourced components / #components*100	0 or 100	0	100	100	33	96	100	-	0	100	0
normalized	10	10	10	10	3.4	9.2	10	-	10	10	10
Inputs		casing	contact	switch	motor	trans-mission	slip clutch	drilling	trigger	chuck	speed changer
#out-sourced components		0	1	1	1	23	6	-	0	2	0
#components made in house		2	0	0	2	1	0	-	1	0	2

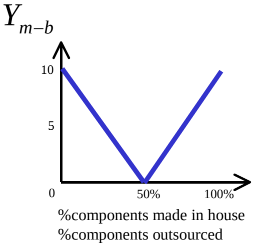


Figure 6 Make-buy metric calculations.

We extend the Design Structure Matrix (DSM) approach of Sosa et al [29] to define an organization alignment metric. However, we seek a metric for the entire platform and not simply for a single product. As with Sosa et al, we combine a component interaction matrix with a team interaction matrix and the degree of overlap of the two then demonstrates the degree of alignment of the architecture with the development organization. With this matrix, one can calculate a metric, similar to one in Newcomb [22], as follows:

$$Y_{org} = \text{components in teams} \cap \text{components in modules} / \text{components in teams} \cup \text{components in modules}$$

Notice that, in an ideal situation, common modules in the products, i.e. the platform modules, are designed only once and then used in multiple products. This also means that there will be so-called *platform teams and tasks*. The product specific modules are developed by *variant teams*. We developed an organizational alignment score for the drill family as 5.4, using the matrix of Fig 5. A re-organization of the development teams would improve the score, especially for the development of the transmission and speed changer modules.

Make-Buy

In a good architecture all out-sourced components come as separate modules, distinct from parts made in-house [5][9]. This allows clear and controlled authority and responsibility for requirements, development, test and quality. We define a scaled metric for any module as follows:

$$Y^*_{m-b} = \left(\frac{\#\text{outsourced components}}{\#\text{components}}\right) \cdot 100$$

We calculate this for all modules in the platform, and we take the average of the product scores. Since all components in a module should be either out-sourced or made in-house, the ideal values for Y^*_{m-b} are 0 or 100. We normalize the values of Y^*_{m-b} to our scale as shown in the right side of Fig 6. The calculation for the drill platform modules is shown in Fig 6. The overall score is the average of the modules' scores: 9.6. This good score results from most of the drill's components being outsourced and practically no power components being made in-house.

Testing

If a function needs separate testing, it should be isolated as a module [5][9]. A good module is one that is both easy to test and where the measured test value corresponds precisely to field requirements. This can be difficult. For example the noise vibration of a car engine can be measured on a test stand but that does not correspond to the actual noise it produces using the vehicle mounting structure and actual driving conditions. Therefore, it is important to consider testing early, at the architectural concept assessment phase.

We define a testing metric for any module by considering its flows. Each flow requiring measurement is scored according to the list below, and all single module's flow scores are averaged to get a platform score.

10 – direct measure of the flow in field conditions
7 – measure of something not the flow in field conditions
5 – indirect on a test stand statistically related to real world
3 – test stand not statistically related to field conditions
0 – no measurement done

The drill family testing score is 5.4. The low score is a result of many of the components being tested separately.

Variety

A purpose of a platform is to easily enable product variants. The following metrics measures how well a platform achieves this goal. They include: *carryover*, *common unit,* and *different specification*.

Carryover

If a specific function can be incorporated into different products without change and no technology upgrades are expected, then the function should be isolated into a module [5][9]. A corollary is that, when inserted into different products, ideally a module has either 100% carry over of its functions or none at all. To capture this thought, we define a platform metric as.

$$Y_{carry} = 10\left(\frac{\#\text{functions to carryover}}{\#\text{functions}}\right)$$

The score for the platform is the average of the modules' normalized scores. The score for the drill platform is a high 9.7, reflecting the fact that for most drill modules, the entire module will either carry over to the next generation with no change (internals), or the entire module will be changed (replaced modules or the shell).

Common unit

If a function is shared by more than one product in a product family or used more than once in a single product it is called a *common unit*. A common unit should be isolated as module from the non-common functions [5][9]. While similar to the previous thought, this goal is sufficient in importance to be called out distinctly. Zamirowski and Otto support this in their rule to define product family modules by identifying common functions in the product family function structures [35]. Kota et al [18] present a method to evaluate a platform based on how well the non-value adding components are shared in a platform. We will use a similar approach.

We start by identifying the functions that distinguish each product variant. For the remainder (the candidate common modules), we ask how common they are. For this subset of modules in the platform that are to be shared, we ask what differences there are from supported variant to supported variant. For the common core, we examine any other module that is attached to the common core. On these attached modules, we use the following scale

$$Y_{CU} = \begin{cases} 10 & \text{Can be swapped into any variant with no changes} \\ 7 & \text{Can be swapped into at least one other variant with no changes} \\ 5 & \text{Requires different mounting hardware to interchange} \\ 3 & \text{Requires interface design changes} \\ 0 & \text{Requires unique interfaces for each variant} \end{cases}$$

Figure 7 summarizes the common unit metric for the cordless drill family. The overall score for the platform is the average of the modules' scores. It is 5.9 for the cordless drill family. The drill platform modules are generally shared only with one other product in the family or are only similar, but not exactly, which leads to scores being 7 or under.

If the features of a module are same but differ in value a more quantitative approach is needed. Hölttä et al introduce a method to help choose common modules for platforms based on the Euclidian distance between the modules inputs and outputs [17]. This could possible be used as a more quantitative metric to evaluate module commonality.

Different specification

Different specification [5][9] means that there are more than one variation of a function or a module. For example, power cords for the U.S. and German markets, or a casing with or without a display. In a good architecture a function with different specifications is isolated into a module. We define a metric to illustrate this.

$$Y_{diff} = 10\left(\frac{\#\text{functions with different specification}}{\#\text{functions}}\right)$$

The score for the product is the average of the modules' normalized scores. The score for the cordless drill family is a perfect 10, reflecting that different specification functions have been well isolated into modules in the drill family.

After Sale

The responsibility of the developer does not end when a product is launched. The following three of metrics evaluate how well the products will do after it has been sold to the customer and after the customer is done using it: *reliability*, *service ease*, and *environmental friendliness*.

Reliability

Platform decisions can have a big impact upon reliability, and it is often critical. The important decisions for our platform concept evaluation purposes are the module selection and partitioning. A basic heuristic we assert is that one should first apportion functions to modules so that each has an equal reliability. Others incorrectly assert putting all unreliable functions into one module to increase overall reliability. Doing this has no impact on overall reliability, as a simple rolled reliability calculation will easily show.

As support for our equal partitioning heuristic, within the software domain under reasonable assumptions, Ferdinand has mathematically proven that to prevent errors the optimal number and size of any module is $\sqrt{n}$, for n failure modes [11]. Larger modules have excess internal opportunities for errors and smaller modules make for excess inter-module opportunities for errors. We extrapolate this to more general non-software design and define an architectural metric based upon how far the number of modules is from this ideal, similar to the assembly ease metric, as

Common unit	Ideal	casing	contact	switch	motor	trans-mission	slip clutch	drilling	trigger	chuck	speed changer
Professional	10		5	5	3	5	5		7	5	5
Heavy duty	10		7	7	7	5	5		7	7	5
Value brand	10		7	7	7	7	7		5	7	
Home use	10		5	7	7	7	7		7	7	
Low price	10		3	7	3	5	3		7	7	
TOTAL			5	7	5	6	5		7	7	5

Figure 7 Common unit calculation for the cordless drill family.

Life cycle stage	Environmental concern of module *i*					
	Materials choice	Energy use	Solid residues	Liquid residues	Gaseous residues	Total
Resource extraction	2	2	2	3	3	12
Platform manufacture	3	2	3	4	4	16
Platform packing & transport	4	4	4	4	4	20
Platform use	3	2	3	4	4	16
Refurbisment, recycling, disposal	2	3	4	4	4	17
Total	14	13	16	19	19	81 / 100
Normalized						8.1

Figure 8 Environmental friendliness of the drill family.

$$Y_{\text{rel }1} = 10\min\left(\frac{\sqrt{n}}{\#\text{ modules}}, \frac{\#\text{ modules}}{\sqrt{n}}\right),$$

While the previous result is very simple and can be applied early, it assumes equal number of failure modes for each module. On the other hand, at some point failure modes and effects analysis (FMEA) will be done, providing an opportunity to relax the uniform failure mode assumption. One can calculate a scaled metric using risk as determined by the risk priority numbers (RPN) of the FMEA. That is,

$$Y_{\text{rel }2} = 10 - \left(RPN_{\max}\right)^{\frac{1}{3}}$$

where $\text{RPN}_{\max}$ is the highest risk priority number of the modules. For the drill platform, we calculate 8.6, indicating the failure-mode-assessed reliability is reasonable, driven mostly by the chucks and contacts. Notice these two metrics naturally evolve into a rolled reliability calculation as reliability data becomes available.

Service ease

Service and maintenance is simpler when the serviceable functions are isolated into modules [5][9]. Dahmus and Otto [7] derive rules for partitioning a product based on service costs, using the service cost model of Gershenson and Ishii [13]. Using this work, it can be shown that to minimize service costs one should isolate two modules as two services module when

$$\frac{C_1}{C_2} \leq \frac{R_2(1-R_1)}{R_1(R_2-1)}$$

where:

C_i is the total cost of subset *i* (isolated and not), including material and any labor

R_i is the reliability of subset *i*

To convert this concept into a platform metric, each module should be analyzed for both whether it should be combined with its neighboring module, and whether it should be split into separate modules. A 0-10 scale can be calculated as the fraction of modules that should be changed. For the drill platform we calculate 10, given that the two modules needing service (chuck and switch) are distinct modules. A refinement of this metric is possible using the actual service cost.

Environmental friendliness

Recycling has become increasingly important driver of modularity. Ericsson and Erixon [9] and Blackenfelt [5] list recycling as a modularity driver. Smith [28] discusses reuse as a reason to modularize. The environmental friendliness of a product in general has been extensively studied. We choose to use the AT&T model developed by Graedel and Allenby [15] to assess environmental friendliness, since it comprehensive in lifecycle scope and has limited data requirements as appropriate for early platform concept evaluation.

The analysis implies giving scores from 0 to 4 (0 being worst) to different matter in different phases of a product's life cycle. The scores for each cell are summed to get an overall 0-100 score that we scale down to 10 for our purposes.

Using available information [4], we calculate the overall environmental friendliness score for the drill platform as 8.1, indicating that there is some room for improvement, but in general the product family does not cause excessive environmental harm.

Platform Summary Analysis

Now that all metrics have been filled in Table 1, we can calculate the total platform scores as well as the category sub-scores. The current drill platform gets a score of 8.0 indicating that the drill platform is well designed. We can see that flexibility is the strongest feature of the platform, but this high score is more due to the fact that there will not be many changes to the drill. It is a mature product in a very stable market and the flexibility is not needed. Organization, on the other hand, is the weak point of the drill platform. This is mainly due to the assembly score that is typically low, but the testing score should be improved.

The other two platforms (A & B) both received lower overall scores than the current platform (see Table 1). Alternative A differs from the current platform by only one module but the score was different (7.9) indicating that our framework does have the resolution to differentiate between platforms. The main differences between the platform A and the original are in the customer and organization metrics. The alternative B represented a more integral platform than the current drill family platform but it was not deemed superior by the scores (8.0 vs 7.5). As expected, the more integral platform received a higher score in meeting customer requirements since the performance, weight, and similar requirements could be better tailored with integral designs. However, the integral design is weaker than the original in all other categories due to e.g. poor flexibility toward change.

CONCLUSIONS

This work developed a comprehensive set of metrics to technically evaluate a product platform. While reasonably comprehensive of what is required for a technical platform evaluation, we find the available metrics non-uniform in information requirements and quality of use. Some metrics are very well researched, validated, and understood. Others have no research available at all.

Carrying the argument further, we recognize a need for not just one metric for these concerns. Ideally, a simplified analysis would be available at the very early concept phase to evaluate platform alternatives. This screening analysis could then be refined with more detailed analyses in the early development phase. For example, the reliability and assemblability analyses have simple metrics to evaluate a platform, based simply on the number of functions and modules. Both also have detailed rollup assessments when more information becomes available on geometry and failure modes. The latter analysis is inappropriate at the early concept phase, the former analysis is inappropriate for the preliminary design phase. Yet, we have a comprehensive platform evaluation tool that can be used to evaluate different product architecture alternatives during development.

REFERENCES

1. Association of National Advertisers, *Prescription for New Product Success*, New York, 1984.
2. Baldwin C. and Clark, K., 2000, *Design Rules: The Power of Modularity Design*. MIT Press, Cambridge.
3. Bass, L., Clements, P., and Kazman, R., 2003, *Software architecture in practice*, 2^{nd} ed, Addison-Wesley.
4. Black and Decker corporate website. http://www.bdk.com/environment/environmental_affairs.htm (viewed January 16, 2004).
5. Blackenfelt, M., 2001, *Managing complexity by product modularization*, Doctoral Thesis, Dept. of Machine Design, Royal Institute of Technology, Stockholm.
6. Boothroyd, G., Dewhurst, P., and Knight, W., 2002, *Product Design for Manufacture and Assembly*, 2nd ed, Marcel Dekker, Inc., New York.
7. Dahmus, J. and Otto, K., 2001, "Incorporating Lifecycle costs into Product Architecture Decisions", *ASME Design Automation Conference*, Pittsburgh.
8. De Weck, O. and Chang, D., 2002, "Architecture trade methodology for LEO personal communication systems", *AIAA 20th Intl Comm Satellite Systems Conference*, Montreal.
9. Ericsson, A. and Erixon, G., 1999, *Controlling design variants: Modular product platforms*, ASME press, New York.
10. Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2002, "Platform selection under performance loss constraints in optimal design of product families", *ASME Design Engineering Technical Conferences*, Montreal.
11. Ferdinand, A., 1993, *Systems, Software, and Quality Engineering*, Van Nostrand Reinhold.
12. Fujita, K., Sakaguchi, H., and Akagi, S., 1999, "Product variety deployment and its optimization under modular architecture and module communalization", *ASME Design Theory and Methodology conference*, Las Vegas.
13. Gershenson, J. and Ishii, K., 1992, "Design for serviceability", in Kusiak (ed), Concurrent engineering: theory and practice, John Wiley, New York. Pp. 362-384.
14. Gonzalez-Zugasti, J., and Otto, K., 2000, "Modular platform-based product family design", *ASME Design Engineering Technical Conferences*, Baltimore.
15. Graedel, T., Allenby, H., and Comrie, P., 1995, "Matrix approaches to abridged life cycle assessment", *Environmental Science and Technology*, **29**, 3, pp. 134A-139A.
16. Holtta, K. and Otto, K., 2003, "Incorporating design complexity measures in architectural assessment", *ASME Design Engineering Technical Conferences*, Chicago.
17. Hölttä, K., Tang V. and Seering, W., 2003, "Modularizing Product Architectures Using Dendrograms" *International conference on engineering design*, Stockholm.
18. Kota, S., Sethuraman, K., and Miller, R., 2000, "A Metric for Evaluating Design Commonality in Product Families", *Journal of Mechanical Design*, **122**, pp. 403 – 410.
19. Martin, M. and Ishii, K., 2002, "Design for variety: developing standardized and modularized product platform architectures", *Research in Engineering Design*, **13**, 4, pp. 213 - 235.
20. Nayak, R., Chen, W., and Simpson, T., 2000, "A variation-based methodology for product family design", *ASME Design Engineering Technical Conferences*, Baltimore.
21. Nelson, S., Parkinson, M., and Papalambros, P., 2002, "Multicriteria optimization in product platform design", *Journal of Mechanical Design*, **123**, pp. 199-204.
22. Newcomb, P., Bras, B., and Rosen, D., 1998, "Implications of modularity on product design for the life cycle", *Journal of Mechanical Design*, **120**, pp. 483-490.
23. Otto, K. and Wood, K., 2001, "Product Design: techniques in reverse engineering and new product development", Prentice Hall, Upper Saddle River.
24. Raiffa, H. and Keeney, R., 1993, *Decisions with multiple objectives: Preferences and value tradeoffs*, Wiley, New York.
25. Rajan, P., Van Wie, M., Campbell, M., Otto, K., and Wood, K., 2003, "Design for flexibility – measures and guidelines", *International conference on engineering design*, Stockholm.
26. Saaty, T., 1988, *The analytic hierarchy process*.
27. Siddique, Z. and Rosen, D., 2000, "Product family configuration reasoning using discrete design spaces", *ASME Design Engineering Technical Conferences*, Baltimore, Maryland.
28. Smith, J. and Duffy, A., 2001, "Modularity in support of design for re-use", *International conference on engineering design*. Glasgow.
29. Sosa, M., Eppinger, S., and Rowles, G., 2000, "Understanding the Effects of Product Architecture on Technical Communication in Product Development Organizations", MIT Sloan School of Management. Working paper # 4130.
30. Sudjianto, A. and Otto, K., 2001, "Modularization to support multiple brand platforms", *ASME Design Engineering Technical Conferences*, Pittsburgh.
31. Suh, N., 2001, *Axiomatic Design: Advances and Applications*, Oxford University Press, New York.
32. Tanaka, M., 1989, "Cost planning in the design phase of a new product", Japanese management accounting, Y. Mondem, M. Sakurai (Eds), Productivity Press, Cambridge.
33. Ulrich, K. and Eppinger, S., 2000, *Product Design and Development*. 2^{nd} ed, McGraw-Hill, New York.
34. Whitney, D., *Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development*, to be published by Oxford University Press.
35. Zamirowski, E. and Otto K., 1999, "Identifying Product Family Architecture Modularity Using Function and Variety Heuristics", *ASME Design Theory and Methodology conference*, Las Vegas.