Int J Adv Manuf Technol (2008) 36:83–96
DOI 10.1007/s00170-006-0805-y

ORIGINAL ARTICLE

Integrated approach to modularize the conceptual product family architecture

Liu Zhuo · Wong Yoke San · Lee Kim Seng

Received: 5 May 2006 / Accepted: 11 September 2006 /

Abstract To meet the increasingly heterogeneous market, a product family strategy is needed to determine how customized variants can be derived from a common product platform within acceptable cost and time. Toward this, a suitably conceived and developed product family architecture (PFA) is important for implementing mass customization. In this paper, we present a systematic approach to modularize PFA at the early design phase, which is the conceptual design stage. The PFA can be viewed as a conceptual structure with the following three interrelated elements: module, variant, and coupling interface. Identifying variant as the external driver of architectural variation, this paper develops a variety index (VI) method to estimate effects of customization on the conceptual modules. Rather than just identification of module boundary in the product architecture, the proposed modularization method translates the variety source generated from requirements analysis into a dynamic configuration of the conceptual PFA, involving variety analysis, product modularization, and generation of product portfolio architecture. An example of a power tool design is used to demonstrate the proposed method.

Keywords Functional modeling · Mass customization · Modularization · Product family architecture

Abbreviations

AMM	attribute module matrix
NRE	nonrecurring engineering
PBPF	platform-based product family
PFA	product family architecture
PPA	product portfolio architecture
PPP	product portfolio planning
QFD	quality function deployment
VI	variety index

1 Introduction

Appropriately implemented mass customization (MC) can provide clear competitive advantage for a manufacturing enterprise [1, 2]. To achieve the economy of scale and accommodate an increasingly customized product variety across diverse market niches, platform-based product family (PBPF) strategy has been widely adopted as an effective means to implement mass customization [3]. Toward this, a suitably developed product family architecture (PFA) is important for organizing a series of products and managing the complexity of product platform. A PFA can be viewed from different domains along the product life cycle: customer, function, physics and process, and has different technical and managerial implications, respectively.

Many works with regard to PFA have been reported in the literature. Most of them extend the concept of traditional product architecture to manage the complexity of product family and model product family architecture in different domains, involving the mapping process across the functional, design and process perspectives [1, 4, 5].

Decisions in the early stage of the product development can have tremendous effect on the downstream activities. Variations in customer-perceived value always result in design changes and related process variations [6]. Thus it becomes imperative to incorporate customized require-

L. Zhuo · W. Yoke San (✉) · L. Kim Seng
Department of Mechanical Engineering,
National University of Singapore,
21 Lower Kent Ridge Road,
Singapore 119077, Singapore
e-mail: mpewys@nus.edu.sg

ments into PFA and construct a flexible and robust product architecture to accommodate variations. Within this context, the implication of PFA can be viewed as a conceptual structure with the following three interrelated elements: module, variant, and coupling interface. Identifying variant as the external driver of architectural variation, this paper develops a variety index (VI) method to estimate effects of customization on modules in the early design phase, and presents a top-down approach to modularize the conceptual PFA, involving variety analysis, product modularization and generation of product portfolio architecture.

In the next section of the paper, the background related to this research is outlined. Then we investigate the principle of PFA and formulate the VI to estimate the variation effects on product architecture, and then present an integrated method to accomplish the modularization of the conceptual PFA, followed by an application of the method to a practical example, which is a power tool design.

2 Literature review

2.1 Product architecture

The development of a product architecture, assigning forms to functional elements, is a critical phase in the design process because the choice generated will strongly influence the product performance in several aspects, including detailed design, manufacturability, product variety, and so on [7]. Ulrich and Eppinger [8] define a product architecture as consisting of two elements: (1) mapping relation between functions and physical elements, and (2) coupling interface. Modularity is referred to as the most important characteristics of a product architecture. Accordingly, there are two types of architecture: modular and integral. A modular product architecture can easily create product variants by combinations of components or building blocks [8, 9]. On the other hand, integral architecture can acquire advantages of performance by individual design and optimization. Several existing reports point out benefits of modular design and integral design from different aspects [5, 8, 10].

Moreover, Van Wie et al. [11] address the embodiment issues of module interface based on assembly costs and later he and his co-authors [12] develop a representation scheme to capture the design issues related to product architecture. Holtta et al. [13] present a method to measure redesign effort based on analysis of material, energy, and information flows. In addition, Fixson [14] investigates the multi-facets of modularity and product interface in three aspects: interface type, reversibility, and standardization.

2.2 Product variety

Variety source can come from space (multiple-context) and time (changing context) [15]. Platform-based product variety strategy has been recognized as an effective means to implement mass customization. Its implication is different along the product life-cycle. For the customer, a product variety can be viewed as variants with customized values for certain product attributes or features [16]. In the engineering design domain, a product variety can be decomposed into three levels: system structure, configuration, and module instantiation level [17]. To provide variety for the increasingly heterogeneous market, three platform leveraging strategies [3] can be identified: (1) horizontal leveraging, (2) vertical leveraging, and (3) beachhead approach, which combines the preceding two. The choice of leveraging approach determines the platform development method. While most horizontal leveraging strategies take advantage of modular platforms, scale-based platform design can be used for vertical leveraging strategies [18].

2.3 Product family architecture

A product family refers to a group of similar products that are derived from a common platform and possess specific features/functionality to satisfy a variety of market inches [3]. Thus, to organize a series of products instead of an individual product, the concept and implication of product architecture have to be extended to manage the complexity of a product family. Accordingly, a product family architecture (PFA) can be defined as the logical organization of a product family [19]. Like product variety, PFA can be viewed from different domains along the product life cycle. Erens and Verhulst [20] assert that the development of a product family requires a product architecture in three domains: function, technological realization, and physical realization. Jiao and Tseng [1] present a method to rationalize product family development for mass customization from three aspects of functional, technical and physical views. Fixson [14] develops a multi-dimensional framework to simultaneously assess a product architecture in the three domains of product, process and supply chain.

Additionally, Du et al. [19] investigate some fundamental issues regarding the architecture of a product family (APF) from both sales and engineering perspective. They view an APF as a conceptual structure that includes common bases, differentiation enablers, and configuration, and also logical organization for generating a family of products. Muffatto and Roveda [21] propose a conceptual framework to model a product architecture and its platform. In their paper, the relation between product architecture, platform and product family are analyzed. As for PFA-based product family development, Martin and Ishii [15] present a design–for-variety method to develop a product family. This method uses two indices to measure the product architecture. Hsiao and Liu [22] use quality function deployment approach to identify the exterior variation source and then apply an interpretive structural model technique to construct the hierarchy of component interactions within a product.

3 Modular design for mass customization

3.1 Principle of PFA

The success of mass customization lies in the manufacturer's ability to cater for the potential market niches by providing suitably customized varieties based on a rationally technical framework in an effective and timely manner. Since most relevant decisions about the cost and schedule of components or parts are made in the design phase, it is believed that mass customization can be approached from the perspective of design, particularly the early stage of design [23].

Here we look at the PFA as a conceptual structure consisting of three elements: module, variants, and coupling interface. Figure 1 illustrates the interrelation among the three elements of the PFA. The traditional product architecture only consists of module and interface in the mass production environment. However, the paradigm of mass customization poses new requirements on the form of product architecture. Being the external factors of a product architecture, variants in the form scattered customer preference result in the spatial and generational varieties and act as a new source of product complexity in PFA. Because of variants some module boundaries have to be redefined or reconfigured to form new modules corresponding to the variant attributes of products. Accordingly, some modules can become a common platform to support the whole product family, and some modules need to be customized in the dimension of scale variable for variety generation. So, the modularization process for a family of products includes not only the identification of module boundary, but also the classification of modules according to customized requirements as well as quantification of the effects of customization on the product architecture.

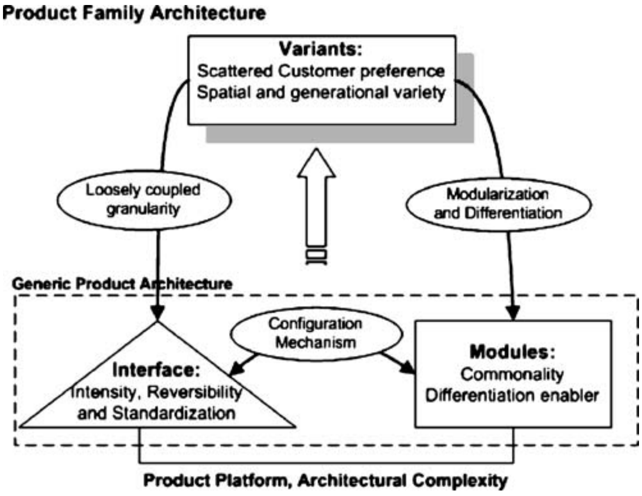

Fig. 1 Three element of product family architecture

At the same time, variants impose certain standardization of the coupling interface between modules and loosely coupled granularity on architectural elements so that the product family architecture can accommodate customization requirements without too many changes of the modules and their interface. Therefore, interface strategy is another important issue in modular product architecture, especially for a family of products [24].

As a result, the realization process of product variants moves toward the modules/components configuration design based on the common platform and differentiation enabler [19]. Meanwhile, combining variants into a product architecture and forming a PFA at the conceptual design stage not only can benefit early product family evaluation and conceptual design, but also address other issues regarding product planning and organization [14], such as supply chain, collaborative development, variety management, etc.

3.2 Variety analysis

In the axiomatic design introduced by Suh [25], product design can be viewed from different domains: the customer, the functional, the physical, and the process domain. Each domain is characterized by the needs or attributes which provide the solution for the preceding domain while giving new requirements for the next domain. To conceptualize the solution, we need the mapping process between the domains and also can mathematically model this mapping process in terms of the characteristic vectors that define the design goals and design solution [25]. Similarly, this method can be applied in the modularization of a PFA to capture the architectural variation due to the customized variants.

The mapping relationship between the customer domain and functional domain can be written as:

$$\{M\} = \{C\}[AMM] \tag{1}$$

where $C=[C_1, C_2 \ldots C_n]$ is the vector of product attribute requirements in the customer domain with a finite number of levels for each attribute, and $M=[M_1, M_2 \ldots M_p]$ is the conceptual module vector in the functional domain. $[AMM]$ is the attribute-module matrix that characterizes the relation between product attribute requirements in the customer domain and the conceptual module in the functional domain. If one product attribute is implemented or affected by one or more modules, there will be a relationship between this attribute and its corresponding modules. The attribute-module matrix has following form.

$$AMM = \begin{bmatrix} a_{11} & \dots & a_{1i} & \dots & a_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ a_{j1} & \dots & a_{ji} & \dots & a_{jp} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{ni} & \dots & a_{np} \end{bmatrix} \qquad (2)$$

$$i = 1, \dots, p \quad j = 1, \dots n$$

Usually the products catering for heterogeneous market niches will be launched at different levels along the dimension of product attributes and results in various product offerings. When the various requirements ($\varDelta C$) occur in the customer domain and needs relating realization, the variation will spread to the functional domain and generate the variant module configuration $\{\varDelta M\}$. This customization process can be written as:

$$\{\varDelta M\} = \{\varDelta C\}[AMM] \qquad (3)$$

Here we develop a variation mapping method, namely *variety index (VI),* to investigate how variants affect the conceptual modules. Similar to quality function deployment (QFD) method in transferring customer requirements into design characteristics, *VI* transfers variance from customer requirements into architectural elements $\boldsymbol{M}=[M_1, M_2 \dots M_p]$ in the function domain as shown in Fig. 2. The range of the product family is a collection of product variants $\boldsymbol{V}=[V_1, V_2 \dots V_m]$ with customized value or level for each product attribute $\boldsymbol{C}=[C_1, C_2 \dots C_n]$.

VI can be represented as *VI*: $\varDelta C \rightarrow \varDelta M$, where $\varDelta C=[\varDelta c_1, \varDelta c_2 \dots \varDelta c_n]$ is variance of attribute and $\varDelta M=[\varDelta M_1, \varDelta M_2 \dots \varDelta M_p]$ is variance of module. To balance the attribute in customer choice, preferences for different attribute are normalized and given through assigning weights by experienced designer. For module *k*, *VI* ($\varDelta M$) can be mathematically represented as follows:

$$VI_k = \sum a_{jk} \varDelta c_j w_j \qquad (4)$$

where w_j is the weight for attribute j to normalize the customer preference.

For all modules, *VI* can be represented as matrix form.

$$\begin{Bmatrix} \varDelta M_1 \\ \varDelta M_2 \\ \dots \\ \varDelta M_p \end{Bmatrix} = \{\varDelta C_1 w_1 \ \varDelta C_2 w_2 \dots \varDelta C_n w_n\} \begin{bmatrix} a_{11} & \dots & a_{1i} & \dots & a_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ a_{j1} & \dots & a_{ji} & \dots & a_{jp} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & \dots & a_{ni} & \dots & a_{np} \end{bmatrix} \qquad (5)$$

$$i = 1, \dots, p \quad j = 1, \dots n$$

Variety index provides a straightforward tool to analyze the variation of the conceptual product architecture due to the customized requirements. The index generated from variety analysis can help the design team to estimate the customization design efforts and focus on the most critical areas in developing the physical architecture.

4 Proposed method for modularizing conceptual PFA

To fully understand the customization effect on product architecture and support platform-based product family development, the modularization of the conceptual PFA should be studied. In our proposed method, there are primarily three steps to modularize the conceptual PFA, as illustrated in Fig. 3. Step 1 involves product family planning. Step 2 utilizes a functional modeling method to identify conceptual modules and generate the variety index (VI) for each module based on the attribute-module matrix. Step 3 generates variants for each module, and then integrates them into a product portfolio architecture (PPA). The three steps are described in the detail in the following sections.

4.1 Product family planning

Since variety source comes from various market segments, it is very necessary to rationally plan the whole product family offered to the customer. In this step, product specifications are collected and prepared for analysis. To finalize the range of target products, the company must choose the optimal level of product attributes for each

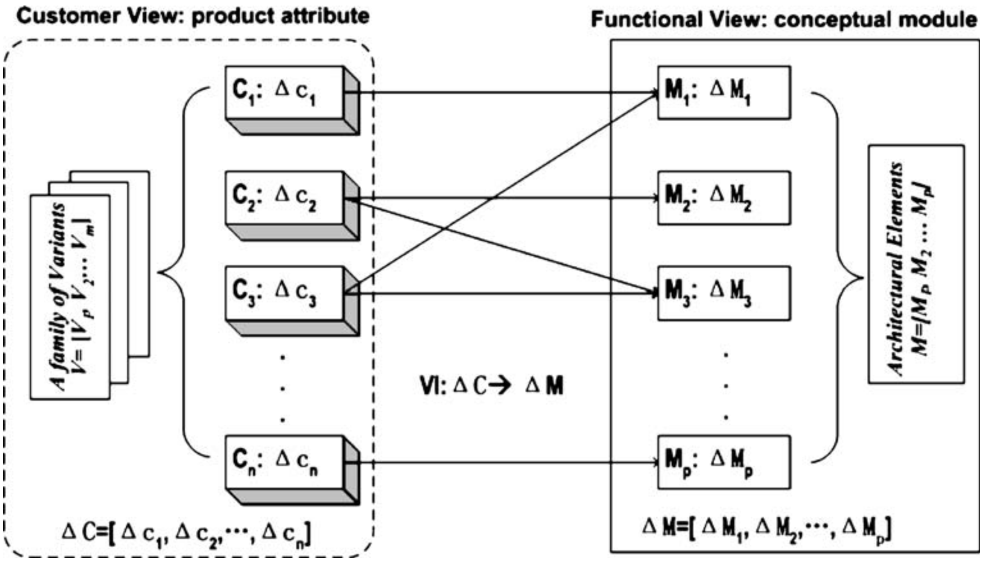

Fig. 2 Illustration of variety index

product variant and the optimal amount of product offerings. Such decisions need the implementation of product portfolio planning (PPP). Quality function deployment (QFD) and conjoint analysis can help decide the choice of level/attribute from the aspects of maximizing customer satisfaction and maximizing profit/sales, respectively [26]. In practice, the PPP problem can be approached from two different perspectives [27]. One is to select a product portfolio from a finite set of candidate items. The other one is to construct a product portfolio directly from part-worth utility data. Jiao et al. [27, 28] develop a heuristic genetic algorithm to solve the PPP problem when the number of attributes and the number of attribute levels become very large, by integrating customer choice and engineering cost.

However, platform strategy evaluation is needed to determine whether the collection of product specifications can be developed on the same platform, considering economic factor and technological feasibility. Some studies [29, 30] have provided methods to optimize this process according to desired requirements.

From the collection of product specifications, we can use Eq. (6) to estimate the variance degree of product attribute ΔC, the source of variety requirement.

$$[\Delta c_j,\ j=1,2 \dots n]=\begin{cases}(N_j-1)/m & \text{, for the discrete attribute}\\ 0 & \text{, for the binary attribute}\end{cases} \tag{6}$$

where Δc_j is the variance degree for product attribute j, N_j is the number of levels offered for product attribute j and m is the number of variants offered in the product family. For example, the voltage attribute need to be customized at the four different levels for six variants and thus the variance degree is $(4-1)/6=0.5$. For binary attribute (i.e., yes or

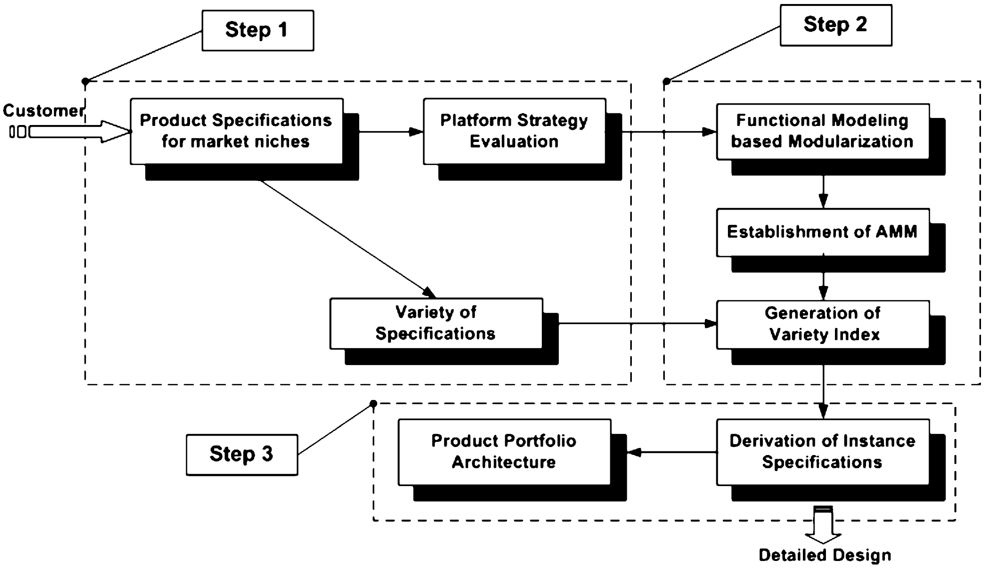

Fig. 3 Three steps for modularizing the conceptual PFA

no type), variance degree is always zero since it can be viewed as an auxiliary feature. So if variance degree of specific product attribute is closer to 1, this attribute will have more customization and require a greater effort to implement.

4.2 Function-based product modularization

For product modularization process, there are three different classes of approaches reported in the literature, customer, structure and function-based approaches as shown in Table 1. Each class of modularization process occurs at a different product development stage and defines its own rule in its scope.

Since the definition of product architecture begins with the arrangement of functional element, function-based modularization method seems to provide a schematic view of architectural exploration for product family early while linking customer needs with engineering design. Meanwhile, the implication of modularity here is to form basic configurable elements for functional share and variety generation. Therefore we apply the functional modeling method to modularize the product.

Functional modeling [31, 32] is a key step at the conceptual design stage, whether original or redesign. It provides an engineering view of how the sub-functions work together to achieve the desired functional requirements, and is independent of how the function is performed. This model uses a graph-based functional design language to form the product conceptual structure, where the product function is characterized in a verb-object (function-flow) format and decomposed further into sub-functions, as shown in Fig. 4. Using the functional model can significantly contribute to the product architecture development by moving the product architecture decision earlier in the conceptual design stage. Then a modular architecture is formed by grouping sub-functions together to form modules based on three heuristic methods: dominant flow, branching flow, and transmission/conversion [39]. The modules identified can be used for concept generation and module instantiation. The function-based modularization method can gain early design commonality in the conceptual product architecture, particular for a series of similar products.

4.3 Establishment of attribute-module matrix and variety index

To fully understand the engineering relation between product attributes and conceptual modules for VI, the source of architectural variation should be identified first. Several research works existing in the literature investigate the mechanism of variation transmission between segmented market and product architecture. Martin and Ishii [15] and Hsiao and Liu [22] have both recognized the different customer requirements as the exterior drivers of variation, and the coupling interactions among components as the interior variation propagation. Similarly, this paper also approaches the generation of the attribute-module matrix (AMM) from two perspectives: namely specification implementation and specification propagation.

4.3.1 Specification implementation

Specification flows can be viewed as the design information that must be passed among designers to design their respective modules. Generally, each product attribute has its specification metric to be customized for different requirements and each conceptual module has the specification output to implement the corresponding product attribute. Thus, by directly mapping the product attribute to the conceptual module implementing the specification, we can generate the AMM from the perspective of specification implementation. The following table shows the relation between specification metrics and conceptual modules in the case study. Table 2

Then we use a 9/6/3/1/0 rating system for estimating the relation values, as shown in Table 3. For each relation node in the matrix, the design team estimates the implementation degree. For example, voltage and charger time are completely implemented by elec. supply module and accordingly their relation values are both 9.

Table 1 Comparison of modularization processes

Orientation	Approach	Methodology	Scope	Case used
Customer	Moore et al., 1999 [16]	Conjoint analysis	Product family	Electrical equipment
	Yu et al., 1999 [34]	Market analysis	Product family	Leg room of car
Function	Stone et al., 2000 [32]	Functional modeling and heuristic method	Product	Electrical equipment
	Dahmus et al., 2001 [35]	Functional modeling & heuristic method	Product family	Power tools
	Kurtadikar et al., 2004 [36]	Functional modeling & heuristic method	Product family	Shop vacuum
Structure	Newcomb et al., 1998 [37]	Modularity measure	Product	Center console
	Gershenson et al., 1999 [38]	Modularity measure	Product	Mechanical pencil
	Hsiao and Liu, 2005 [22]	Interpretive Structural Model	Product family	Coffee maker

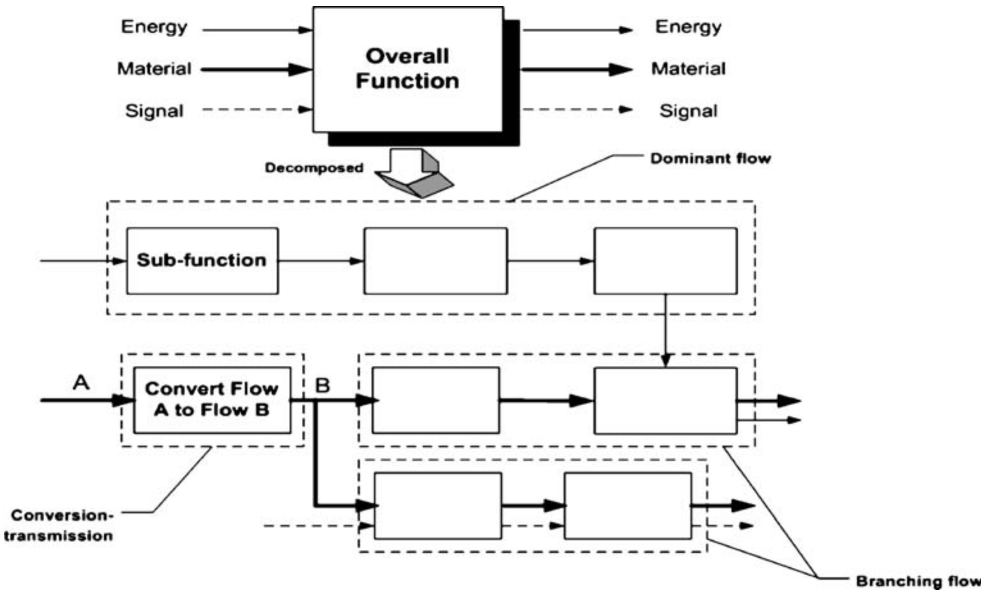

Fig. 4 Illustration of function modeling

4.3.2 Specification propagation

The coupling interaction among modules in term of specification flow is another important aspect of product architecture according to the definition by Ulrich [7]. Thus the specification change sourcing from the various customer requirements may spread within the product architecture by module interaction. Here since we focus on the functional analysis of product architecture, the functional modeling can provide a visual tool to identify the specification propagation within the product architecture by tracing the flows.

For the specifications propagated among modules, the design team should estimate the sensitivity of each module to a change in those propagated specification flows. Here we also use 9/6/3/1/0 rating system to quantify the sensitivity. If a small change in the propagated specification requires a large change in the realization of the module, this module has a high sensitivity to the change of the attribute with that specification and thus their relation is given a rating of 9. For example, the electrical flow associated with voltage is propagated to actuator module. Although the electrical flow can range from 12 V to 24 V, a small change occurs to the actuator (switch) since most switch designs can accommodate the different voltage settings from 12 to 24 V. So the relation between voltage and actuator module is rated 1. However, for elec.-to-torque module (motor), the change of electrical flow will require a moderate design change and thus the relation between voltage and elec.-to-torque module is rated 6.

4.3.3 Variety index

Traditionally individual product design captures customer needs and transfers them into engineering design. However current multi-product development usually involves design-

Table 2 Specification Implementation between modules and attribute

Module and flow output	Product attributes and specification metrics							
	Voltage (V)	Max. torque (in-lbs)	# of variable speed	Speed (rpm)	Hammer capacity	Chuck size (mm)	Charger time (mins)	Clutch setting
Elec. supply (Elec. V)	X						X	
Actuator								
Elec.-to-torque (Torque, in-lbs &rpm)		X		X	X			
Transmission (Torque, in-lbs &rpm)		X	X	X				
Hammer					X			X
Secure						X		
Handle								

Table 3 VI rating system

Rating	Description
9	Has a crucial relation between attribute and module
6	Has a strong relation between attribute and module
3	Has a partial relation between attribute and module
1	Has a minor relation between attribute and module
0	No relation exists

ing a series of similar products/components synchronously and continuously. This compels designers to understand the variation effect from customers on product architecture and later detailed design in the early design phase. Thus it is very necessary to evaluate and quantify the variation of product architecture after identifying the basic module boundary.

Given a collection of the specifications for a product family to meet different market niches, product architecture, and attribute-module relation, we can estimate its *VI* for each module according to Eq. (4), which can also be an indicator of the extent of efforts we need to implement module instantiation process in the subsequent detailed design stage.

4.4 Instance derivation and product portfolio architecture

The modules with low *VI* can be the common modules of product platform and will be used across the entire product family. On the other hand, the modules with high *VI* can be the differentiating modules, which need more instantiation for the specific variant product. Differentiating modules and their instances can be viewed as class-member relations. The instantiation design usually involves the scale-based parameter optimization along the different attributes of differentiating modules [18].

In order to facilitate product development, product variants with their corresponding module instances in terms of engineering specification should be generated. Du et al. [19] have investigated this process, namely variant derivation, which involves four important steps: selection constraints, parameter propagation, include condition, and variety generation. In this paper, parameter propagation is mainly used to accomplish the derivation of module instances.

At the end of this step, a product portfolio architecture (PPA) is formed to guide subsequent design activities and early product evaluation, which provide information for the instantiation of differentiating modules. PPA is an engineering view of product family configuration characterized by combinations of the common modules $\{CM_j^*, j=1, 2 \ldots k\}$, and a set of differentiating modules $\{DM_i, i=1, 2 \ldots n\}$ associated with their customized instances. To describe a family and its product variants, a decomposition /classification structure can be adopted to represent PPA as shown in Fig. 5. In the vertical direction, PFA is constructed in a hierarchical form with the decomposition in each sub-level and with instances attached to the end modules. In the horizontal direction, PPA is organized into two categories of conceptual modules: common and differentiating.

In summary, the PPA develops the targets for the further development of product family and provides an engineering

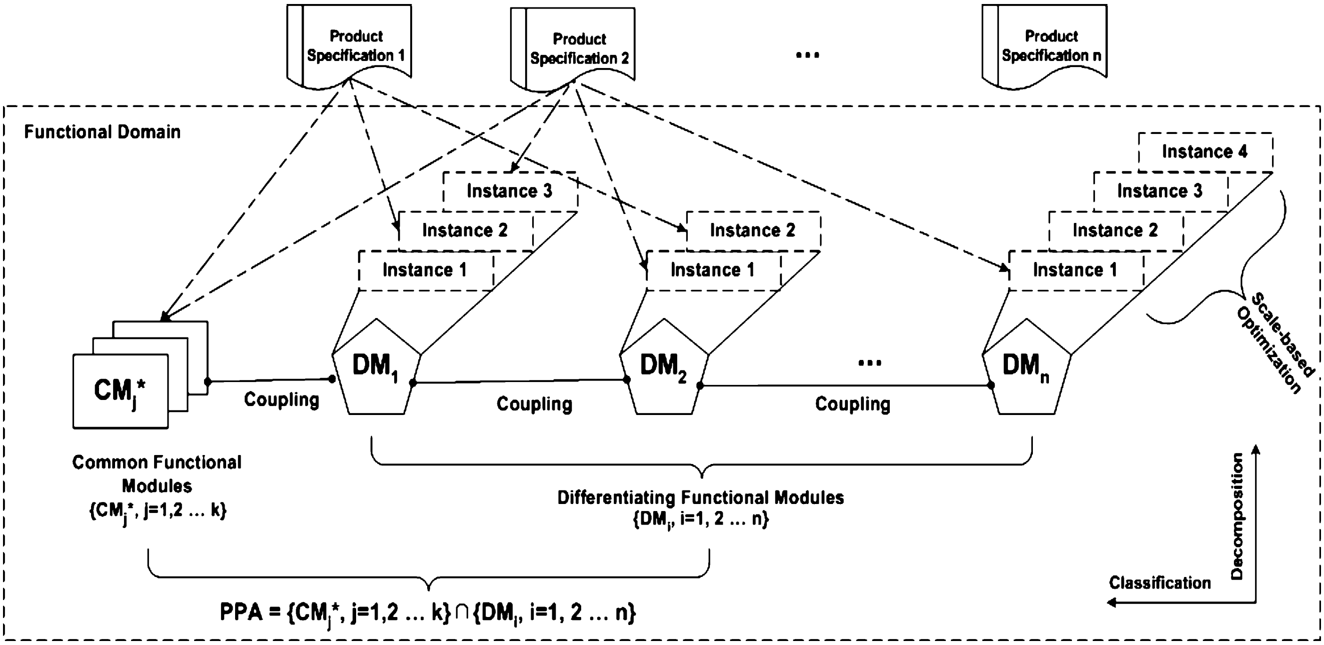

Fig. 5 Product portfolio architecture of engineering view

insight to understand product variety in terms of conceptual module configuration.

5 Case study

For the case study, a family of power tools is used here to illustrate the aforementioned approach. In addition, vertical leveraging strategy is adopted to provider a variety of similar products, which have the same functionality but different price-performance levels.

5.1 Power tool family

Usually power tool families are identified according to functional features and performance requirements. The investigated power tools are centered on a cordless drill series. Based on the market analysis, eight product attributes that are important to the customers are identified: voltage, maximum torque, the number of variable speed, no load speed, clutch settings, chuck size, charging time, and hammer capacity. To balance the attribute importance in customer choice, preferences for different attribute are

Table 4 Specification collections of product variants

Common attributes		
Clutch settings (C8)	23	For all variants
Charging time (C7)	1 hour	
Clutch size (C6)	13 mm	
Differentiating attributes for variants		
Voltage (C1)	12 V	V1
	14.4 V	V2, V4
	18 V	V3, V5
	24 V	V6
Max. torque * (C2)	350 in-lbs	V1
	400 in-lbs	V2
	450 in-lbs	V3, V4
	500 in-lbs	V5
	550 in-lbs	V6
Number of variable speeds (C3)	2	V1,V3,V4
	3	V2,V5,V6
No load speed (C4)	500 rpm	For all variants
	1500 rpm	V1,V2,V3,V4,V5
	2000 rpm	V2,V5,V6
Hammer (C5)	yes	V4,V5, V6
	no	V1,V2,V3

*In-lbs=0.109 Nm

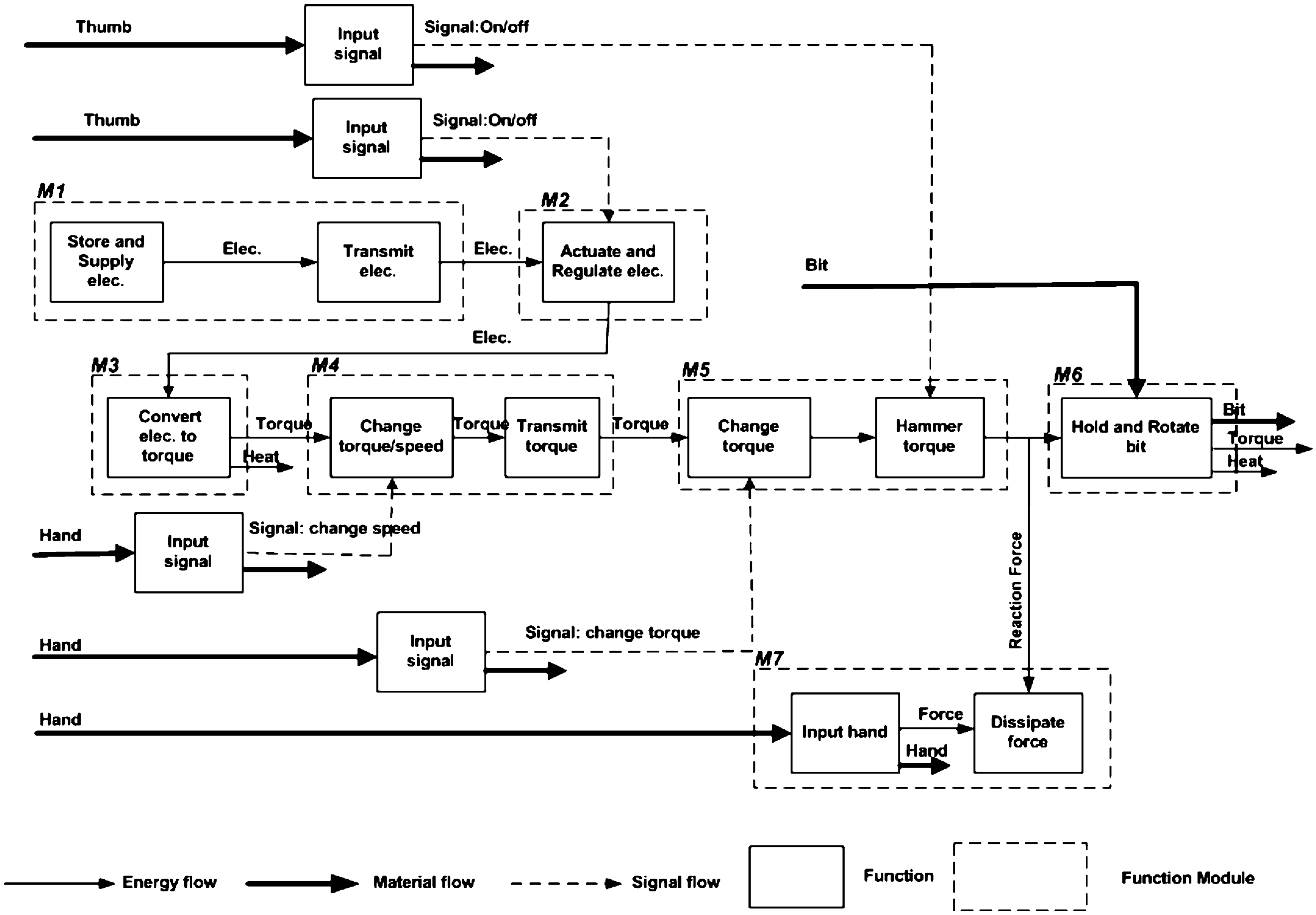

Fig. 6 Functional modeling of power tool family

Table 5 Modules of the cordless drill family

Module	Module name	Used heuristic method
M1	Elec. supply module	Dominant flow
M2	Actuator module	Dominant flow
M3	Elec.-to torque module	Transformation/conversion
M4	Transmission module	Dominant flow
M5	Hammer module	Dominant flow
M6	Secure module	Dominant flow
M7	Handle module	Branching flow

normalized and given through assigning weights by experienced designer. The weight of each attribute is listed in Table 4. To meet different market niches in the vertical direction, the levels of each attribute are customized for six variants [V1... V6] as the following through conjoint analysis and selection constraints.

Product family planning and strategy evaluation is not employed in this example since several related methods exist in the literatures [27–30].

According to Eq. (5), the vector of variance degree for each product attribute in the family of cordless drills is ΔC= [3/6, 4/6, 2/6, 2/6, 0, 0, 0, 0].

5.2 Function-based modularization

For the family of cordless hammer/drill listed previously, the function model of the power tool family is shown in Fig. 6 and Table 5 lists the candidate modules identified by functional modeling. By tracing the flow status through the whole function chain, we can divide the cordless drill into four sub-systems: electrical, conversion, mechanical, and support sub-systems. Each sub-system is decomposed into modules, which address the implementation of related product attribute. In this example, electrical sub-system includes elec. supply module and actuator module. Conversion sub-system only has elec.-to-torque module. Mechanical sub-system contains transmission, hammer, and secure module. Handle module is accessorial support sub-system.

Since the target product family has the same product attributes but different levels of attributes, they can share a common functional model. However, for each product variant, these modules in the functional model differ in the flow intensity.

5.3 Attribute-module matrix and variety index

In this example, AMM is determined by interviewing the experienced designers. Two perspectives of attribute-module relation will be explored according to the method mentioned earlier. Figure 7 illustrates the final mapping relation result regarding specification implementation and specification propagation. Table 6 lists the assigned value of AMM.

Then the *VI* can be computed by multiplying the relation value between the attribute and the module by variance degree and attribute weight. For instance, electricity supply module contributes two attributes: voltage and charging time. Voltage and charging time will vary for product family with variance degree 3/6 and 0. Thus according to Eq. (4),

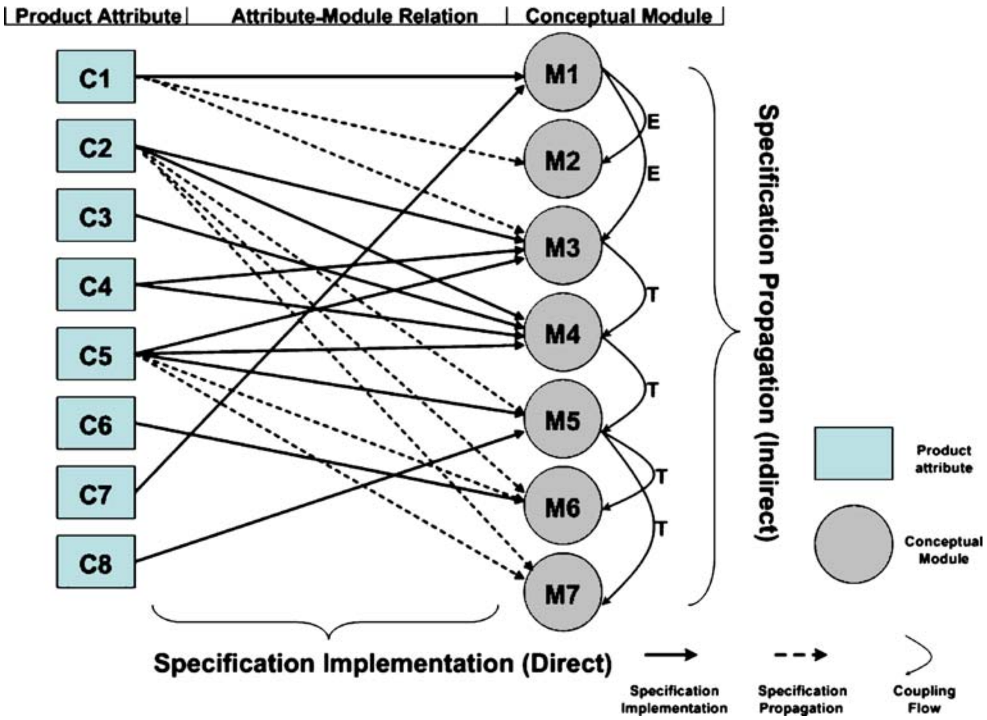

Fig. 7 Two perspectives of attribute-module relation

Table 6 Attribute-module matrix and variety index

Sub-system	Module	Product attributes								VI (ΔM)
		Voltage (C1)	Max. torque (C2)	# of variable speed (C3)	Speed (C4)	Hammer capacity (C5)	Chuck size (C6)	Charger time (C7)	Clutch Settings (C8)	
Electrical	Elec. Supply module (M1)	9	0	0	0	0	0	9	0	**0.63**
	Actuator module (M2)	*3*	0	0	0	0	0	0	0	**0.21**
Conversion	Elec.-to torque module (M3)	*6*	9	0	6	3	0	0	0	**2.08**
Mechanical	Transmission module (M4)	0	6	9	9	6	0	0	0	**1.84**
	Hammer module (M5)	0	*1*	0	0	9	0	0	9	**0.17**
	Secure module (M6)	0	*1*	0	0	*3*	9	0	0	**0.17**
Support	Handle module (M7)	0	*1*	0	0	*3*	0	0	0	**0.17**
Weight of attribute (w)		0.14	0.25	0.20	0.08	0.14	0.03	0.08	0.08	
Variance degree (ΔC)		3/6	4/6	2/6	2/6	0	0	0	0	

VI for electricity supply module will be $VI_1(\Delta M_1) = \sum a_{j_1}\Delta c_j w_j = 9 * 0.14{*}3/6 + 9 * 0.08 * 0 = 0.63$. Table 6 lists the attribute-module relation and result of VI for each module.

Variety index is helpful for identifying crucial modules to meet the customer's various requirements while providing an indicator of platforming in product architecture. In this case, modules with lower VI<0.5 (M2, M5, M6 and M7) can become fixed for all variants to form the common platform. Modules with higher VI>0.5 (M1, M3 and M4) need to be instantiated to meet differentiating requirements.

5.4 Instance derivation and product portfolio architecture

In order to identify the modules and their corresponding components/assembly on which we should focus in the latter design, module-component categorization is given in Table 7 and plotted in Fig. 8. This process also lists the estimated nonrecurring engineering (NRE) costs, which are indicators of designing the differentiating components. In addition, the feasibility of over-design for each module is evaluated by the design team and represented as "bubble" as shown in Fig. 8, to indicate how well the component design can accommodate the specification change through over-design without greatly increasing the cost and causing much performance lost. If the component has a higher VI and NRE cost, balance needs to be made to decrease the developing cost incurred from the increasing number of instances.

Table 7 Module-component categorization of cordless drill

Module	Component/ assembly	VI	Feasibility of over-design	NRE (*$)
M1	Battery	0.63	Low	3000
M2	Switch	0.21	High	500
M3	D.C motor	2.08	Low	5000
M4	Gear assembly	1.84	Medium	18 000
M5	Clutch (Cam)	0.17	High	7000
M6	Chuck	0.17	High	1500
M7	Handle	0.17	High	1000

*NRE is nonrecurring engineering cost and cannot be repetitive

Although a very small variation exists for those modules with lower VI, they can be considered "fixed" for the product family through over-design. This implies designing these modules so that they can accommodate a moderate range in the specification without increasing cost and affecting performance. For example, the actuator module (switch) may require different working voltage for each variant. However designing the switch that can work at different voltages raging from 12 to 24 V can keep one instance for all variants since there is little cost increase and performance lost. In this example, actuator (M2), hammer (M5), secure (M6), and handle (M7) modules compose the platform and may be used across the whole family. On the other hand, those modules with higher VI will be differentiating modules and need more instances to meet customization. Similarly, effective over-design can help reduce the number of instances. In this example, elec. supply (M1), elec.-to-torque (M3), and transmission (M4)

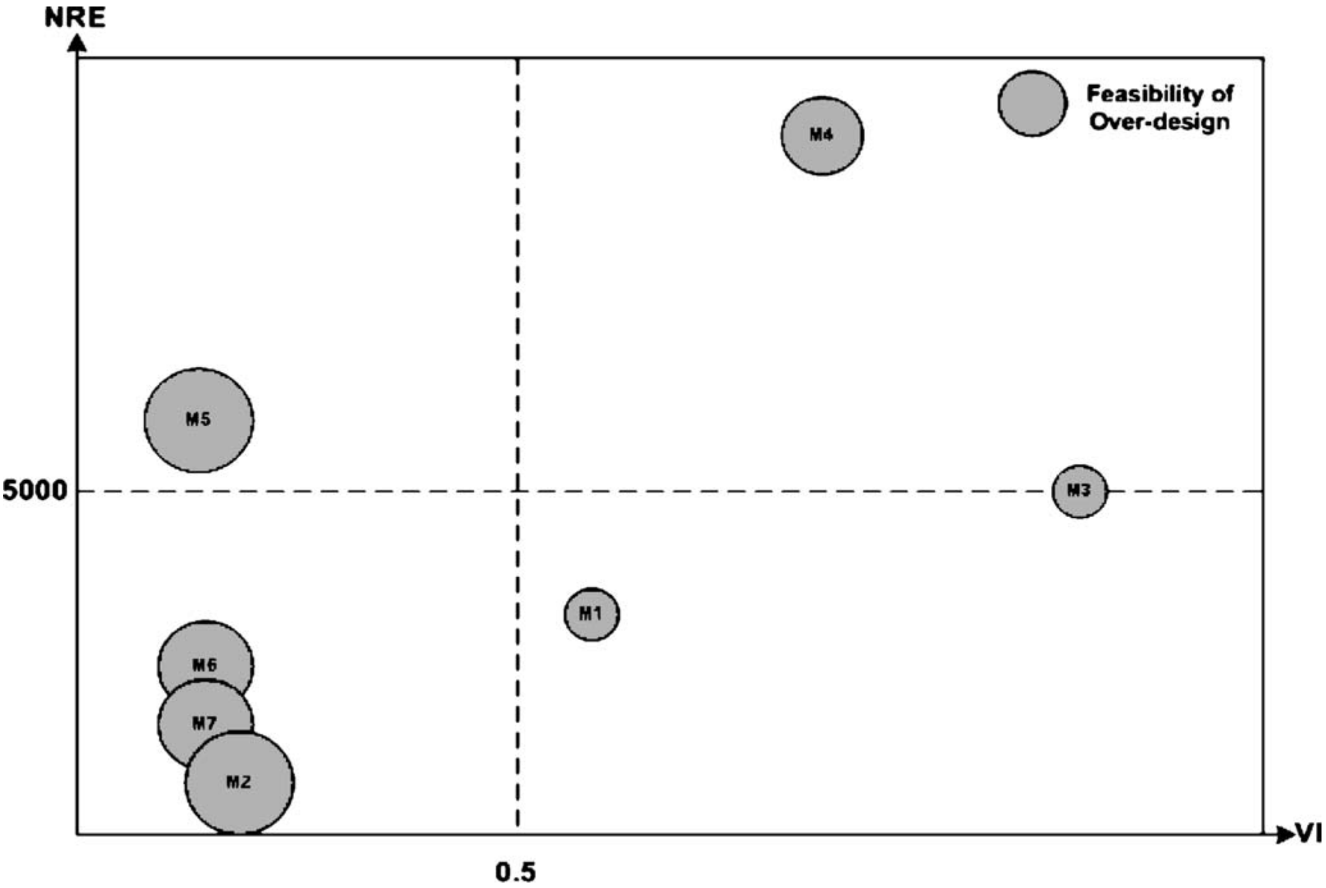

Fig. 8 VI versus NRE versus feasibility of over-design

modules have larger VI values and constitute the differentiating modules, and will be instantiated at a latter detailed design stage to meet different specifications. Gear assembly (M4) has a highest VI and NRE cost. To reduce cost saving from repetitive design, only three instances will be developed through over-design of gear dimension to accommodate more stress requirements, instead of six instances generated for each variant.

Table 8 Variant specification for differentiating modules (±10%)

	Module/Component	Instance	Specification Description	Variants
Differentiating	M1/Battery	I_{11}	Output:12 V; Capacity: 1.7 Amp-Hour	V1
		I_{12}	Output:14.4 V; Capacity: 1.7 Amp-Hour	V2,V4
		I_{13}	Output:18 V; Capacity: 1.7 Amp-Hour	V3,V5
		I_{14}	Output: 24 V; Capacity: 1.7 Amp-Hour	V6
	M3/D.C. motor	I_{31}	Input:12 V; Max. Speed:20 k rpm, Torque (stall):954 mNm	V1
		I_{32}	Input:14.4 V; Max. Speed:20 k rpm, Torque (stall):1090 mNm	V2
		I_{33}	Input:14.4 V; Max. Speed:20 k rpm, Torque (stall):1226 mNm	V3
		I_{34}	Input:18 V; Max. Speed:20 k rpm, Torque (stall)=1226 mNm	V4
		I_{35}	Input:18 V; Max. Speed:20 k rpm, Torque (stall)=1362 mNm	V5
		I_{36}	Input:24 V; Max. Speed:20 k rpm, Torque (stall)=1500 mNm	V6
	M4/Gear assembly	I_{41}	Output:Speed=500/1500 rpm, Torque (stall)=50 Nm	V1,V3,V4
		I_{42}	Output:Speed=500/1500/2000 rpm, Torque (stall)=55 Nm	V2,V5
		I_{43}	Output:Speed=500/2000 rpm, Torque (stall)=60 Nm	V6
Common	M2/Switch	I_{21}	D.C; Voltage:12–24; Switch type: on/off/Variable speed	All
	M5/Clutch(Cam)	I_{51}	Coil Clutch; (Number of cam tooth:18)	All
	M6/Chuck	I_{61}	Chuck Size:13 mm; Keyless; Single sleeve; Metal	All
	M7/Handle	I_{71}	Material: plastic	All

Over-design can be an effective design strategy to reduce the number of instances and increase commonality across a family of products by designing components to accommodate the specification variation. However some disadvantages can be incurred from over-design. One of them is that the individual performance may be impaired. Also the increased material costs due to over-design dimension of a component may limit the commonality of this component. In addition, the quantity distribution among the required instances is another factor to determine the use of over-design [33].

After variety analysis for each module, the suitable instance specification for each module should be determined according to variant requirements. This process is always based upon a set of selected options from customers [19]. Selected attribute levels are transformed to the variety parameters of the end-product and then propagated down the hierarchy of product architecture. Through this parameter propagation, all parameters of module instances obtain specific values. However the allocation of parameter value to each instance requires domain knowledge about mapping functional requirements to design parameters [25].

Table 8 lists the xxsi variants at the end-product level and the corresponding instances for each module generated by design team.

6 Conclusion

Recognizing the necessity to explore the customization effect on product architecture, this paper investigates the modularization of the conceptual product family architecture. The development of conceptual PFA impacts on later configuration design and module instantiation of the platform-based product family. This paper extends the concept of the traditional product architecture and models the product family architecture as a conceptual structure with three important interrelated elements: module, variant, and coupling interface. Variants in term of different customer requirements act as the exterior drivers of architectural variation and meanwhile variation is propagated within product architecture through module interaction. Thus, variety index (VI) is developed to estimate variance degree of each module due to different customer requirements and indicate the architectural area of platforming at the early stage of product family development. Although the generation of VI depends on the specific design context, VI can serve as an indicator of how much modularization design effort is needed to implement customization spreading from customer requirements to engineering design.

The product modularization process for a series of related products is an important aspect in developing product family. It not only involves identifying the module boundary, but also includes the classification of modules and quantification of the customization effect on generic product architecture. Within this framework, a step-by-step method to systematically modularize the PFA has been proposed. This method begins with product portfolio planning, and then adopts the functional modeling method to identify module boundary. The generation of VI for each module is accomplished by establishing the attribute-module relation from two perspectives: specification implementation and specification propagation. Finally, instance specifications for all modules are derived and integrated to form product portfolio architecture, which provides an engineering view of variety generation and develops targets for further product family design. An industrial example of a power tool family design is used to show the application of this method.

Although this study highlights some important issues regarding the variety analysis and modularization of the conceptual product family architecture, there are still some limitations that need to be solved in future research. Interface is another important aspect in product family architecture. How to evaluate or quantify the degree of coupling interfaces existing among the modules and their effect on product family development may be a challenge for further research.

Acknowledgements The authors will like to thank Mr. Zhu Shiyan of Greapo Power Tools (Suzhou), China for his assistance in the case study.

References

1. Jiao JX, Tseng MM (1999) A method of developing product family architecture for mass customization. J Intel Manuf 10:3–20
2. Silveira GD, Borenstein D, Fogliatto FS (2001) Mass customization: literature review and research directions. Int J Prod Econ 72:1–13
3. Meyer M, Lehnerd A (1997) The power of product platforms, building value and cost leadership. Free Press, New York
4. Simpson TW (2004) Product platform design and customization: status and promise. Artif Intell Eng Des Anal Manuf 18:3–20
5. Jose A, Tollenaere M (2005) Modular and platform methods for product family design: literature analysis. J Intell Manuf 16:371–390
6. Jiao JX, Tseng MM (2004) Customizability analysis in design for mass customization. Comput Aided Des 36:745–757
7. Ulrich K (1995) The role of product architecture in the manufacturing firm. Res Policy 24:419–440
8. Ulrich K, Eppinger SD (2000) Product design and development. McGraw-Hill, Boston
9. Gershenson JK, Prasad GJ, Zhang Y (2003) Product modularity: definitions and benefits. J Eng Des 14(3):295–313
10. Mikkola JH, Gassmann O (2003) Managing modularity of product architectures: toward an integrated theory. IEEE Trans Eng Manage 50(2):204–218
11. Van Wie MJ, Greer JL, Matthew IC, Stone RB, Wood KL (2001) Interfaces and product architecture. ASME, Conf Proc DETC2001, Pittsburgh, PA, USA

12. Van Wie MJ, Rajan P et al (2003) Representing product architecture. ASME, Conf Proc DETC2003, Chicago, IL, USA
13. Holtta KM, Otto KN (2005) Incorporating design effort complexity measures in product architectural design and assessment. Des Stud 26:463–485
14. Fixson K (2005) Product architecture assessment: a tool to link product, process and supply chain design decisions. J Oper Manag 23:345–369
15. Martin MV, Ishii K (2002) Design for variety: developing standardized and modularized product family architectures. Res Eng Des 13:213–235
16. Moore WL, Louviere JJ, Verma R (1999) Using conjoint analysis to help design product platforms. J Prod Innov Manag 16:27–39
17. Fujita K, Yoshida H (2004) Product variety optimization simultaneously designing module combination and module attributes. Concurr Eng: Res Appl 12(2):105–118
18. Simpson TW, Jonathan RA, Farrokh M (2001) Product platform design: method and application. Res Eng Des 13:2–22
19. Du XH, Jiao JX, Tseng MM (2001) Architecture of product family: fundamentals and methodology. Concurr Eng: Res Appl 9 (4):309–325
20. Erens F, Verhulst K (1997) Architectures for product families. Comput Ind 33:165–178
21. Muffatto M, Roveda M (2002) Product architecture and platforms: a conceptual framework. Int J Technol Manag 24(1):1–16
22. Hsiao SW, Liu E (2005) A structural component-based approach for designing product family. Comput Ind 56:13–28
23. Jiao JX, Tseng MM (2000) Fundamentals of product family architecture. Integr Manuf Syst 11(7):469–483
24. Chen KM, Liu RJ (2005) Interface strategies in modular product innovation. Technovation 25:771–782
25. Suh NP (2001) Axiomatic Design: Advances and Application. Oxford University Press, New York
26. Pullman ME, Moore WL, Wardell DG (2002) A comparison of quality function deployment and conjoint analysis in new product design. J Prod Innov Manag 19:354–364
27. Jiao JX, Zhang YY, Wang Y (2005) Product portfolio planning with customer-engineering interaction. IIE Trans 37(9):801–814
28. Jiao JX, Zhang YY, Wang Y (in press) A heuristic genetic algorithm for product portfolio planning. Comput Oper Res
29. Gonzalez-Zugasti JP, Otto KN, Baker JD (2000) A method for architecture product platforms. Res Eng Des 12:61–72
30. Weck OL, Suh ES, Chang D (2003) Product family and platform portfolio optimization. ASME, Conf Proc DETC2003, Chicago, IL, USA
31. Pahl G, Beitz W (1996) Engineering design: A systematic approach. Design Council, London
32. Stone R, Wood K (2000) Development of a functional basis for design. J Mech Des 122(4):359–370
33. Krishnan V, Gupta S (2001) Appropriateness and impact of platform-based product development. Manage Sci 47(1):52–68
34. Yu JS, Gonzalez-Zugasti JP, Otto KN (1999) Product architecture definition based upon customer demands. J Mech Des 121 (3):329–335
35. Dahmus JB, Gonzalez-Zugasti JP, Otto KN (2001) Modular product architecture. Des Stud 22:409–424
36. Kurtadikar RM, Stone RB, Van Wie MJ, McAdams DA (2004) A customer needs motivated conceptual design methodology for product portfolios. ASME2004, Conf Proc DETC, Salt Lake City, UT, USA
37. Newcomb PJ, Bras B, Rosen DW (1998) Implications of modularity on product design for life cycle. J Mech Des 120:483–490
38. Gershenson JK, Prasad GJ, Allamneni S (1999) Modular product design: a life-cycle view. J Integr Des Process Sci 3(4):13–26
39. Stone R, Wood K, Crawford R (2000) A heuristic method for identifying modules for product architectures. Des Stud 21:5–31