Product Configuration Flow from Obtaining Customer Requirement to Providing the Final Customized Product

Yanhong Qin
Chongqing Jiaotong University, Chongqing, 400074, China
Email: qinyanhong24@163.com

Guangxing Wei
Chongqing Jiaotong University, Chongqing, 400074, China
Email: wgx777@126.com

***Abstract*—The paper analyzes the modular structure of product family by decomposing the product family into generic modules. Then, the module model denoted by attribute variable is established for each generic module. Based on the classification of customer requirement and erection of product decision tree, the description and explanation type of customer requirement can be used to restrict the sub branch of decision tree in order to find out the certain product family satisfying customer. In approach of quality function deployment, the customer requirement is mapped to the module attribute, which can determinate the value and weight of module attribute variable. After retrieving the candidate set of modules which are nearest to the customer requirement on the module accordingly, the candidate modules are combined efficiently under the constraints relative to modules and attributes, In this way, many invalidate module combinations can be avoided and hence the efficiency of product configuration is promoted.**

***Index Terms*—modular product family, product decision tree, attribute variable, customer requirement classification, product configuration**

I. Introduction

In an age when consumers demand customized products of high quality and low price, the competition among firms has been aiming at product variety and speed of response to market. Mass customization aiming at delivering an increasing product variety, which best serves customer requirements while keeping mass product efficiency, has recently received numerous attention and popularity in industry and academia alike [1]. For the product configuration is aiming at satisfying customer requirement, while various customers will adopt different methods to express their own requirement, it is important to classify customer requirement by translating it into attribute information of product module. Literature [2] classified the customer requirement into three types: binary, optional, parameter. Furthermore, Literature [3] set a taxonomy model of customer requirement information, which classified the customer requirement into five types: binary, optional, parameter, description and explanation, and based on the taxonomy model of customer requirement information, the corresponding formal semantic description approaches to these five types of the information about customer requirement were discussed. Literature [4] analyzed the current research situation of customer requirement in mass customization and developed the matter-element model. Literature [5] pointed out the limitation of mass customization, such as small optional scope of customized user and long delivery time and so on, and then proposed the model of prototype of dynamic customization system. A methodology of product family architecture (PFA) was developed to rationalize product development for mass customization, by which the diverse customer requirements were matched with the capabilities of a firm through systematic planning of modularity in functional, technical and physical views. Besides, mapping of functional requirements to specific modules was considered [6], which classified the product variety optimization into three degree of optimization problems, i.e. attribute assignment, module combination, and simultaneous design of both. When the probabilities of computational optimization for product variety design under fixed product architecture were explored, the contents of modules and their combination under fixed product architecture was determined in approach of optimization [7]. Besides, literature [8] proposed an optimization method of module combination for products in a family, and the unified modeling language (UML) was used to describe a product family [9], but the method still focused on how the customer's functional requirement be translated into a selection of specific modules in the product family.

The functional variety is used broadly to denote any differentiation in the attributes related to a product functionality from which the customer derives a benefit.

Manuscript received Dec. 1, 2010; revised Jan. 5, 2011; accepted Jan. 12, 2011.
Project number: KJ100412

© 2012 ACADEMY PUBLISHER
doi:10.4304/jsw.7.2.308-315

On the other hand, the technical variety such as diverse technologies, design methods, manufacturing process, components and assemblies, etc., is necessary to achieve some functionality of a product required by the customer [10]. While the functional variety is often related to the customer satisfaction, the technical variety usually involves the manufacturability and costs. An important target of product configuration in mass customization is to increase functional varieties, meanwhile reduce technical varieties.

Design in approach of modular product family is an efficient method to realize mass customization. Literature [11] established modular structure of product family, and explained the constraints among module types, between module types and attributes, between attributes in different module types and internal constraints, among different attributes in the same module type. A module type is a model of the set of modules, which are interchangeable, perhaps with some restriction. But the problem of how to choose the exact modules for the specific customer requirements or configuring product based on decomposing customer requirements is yet not solved. Literature [12] developed an optimization method based on simulated annealing technique for the assemble-to-order manufacture paradigm, but neither considered the constraints defined in the module and attribute, nor formulated how to match modules efficiently when there were a large number of various modules in the module case database.

To solve the problems above, this paper is organized as follows. Based on functionality analysis of each product in the same product family, section 2 decomposes the product family into various generic modules from top to down, and then the module model for each generic module is established and denoted by attribute variables. In section3, based on customer requirement classification and product decision tree where the product family matching customer requirements will be filtrated, and then, by reference to quality function deployment (QFD), the mapping relations of customer requirement to technique requirement and continuously, technique requirement to module attribute requirement are formed, which can determinate the value and weight of attribute variables in module model. So, the candidate set of modules which are nearest to the customer requirement on the certain module model can be retrieved accordingly, and the candidate modules are combined efficiently under the constraints relative to modules and attribute. In section 4, the configuration flow is illustrated for the specific customer requirement. In this way, many invalidate module combinations can be avoided and hence the efficiency of product configuration is promoted. And the conclusion is drawn in section 5.

II. MODULAR PRODUCT FAMILY

A. *Module model and generic module*

Our research as in reference [2] and [3] illustrated the modular structure of a single product in a product family. Each tree has a root node denoting a certain product having some child nodes, which denote modules. Some child nodes involve their child nodes further, but others have not, which are called as leaf nodes. In approach of the design principle of product family and modularity, the product family is decomposed into a series of generic module (GM) based on analyzing the functionality of various products in the same product family.

If a module model (shorten as M) is established for each GM, each M can be regarded as a configurable unit. Modules derived from the same M belong to a certain kind of GM. The M can be described by multi-attributes. When some attributes can be evaluated by many values, the attributes can be treated as variables, i.e. attribute variables in module model, shorten as module attribute variables. The differentiations of the different modules in the same GM derives from different values of partial or all module attribute variable.

B. Module attribute variable

For the specific modular product, value of each module attribute variable is certain. But the values of all the module attribute variables in a modular product family are not decided, while only a domain corresponding to each attribute variable can be reached. For some GMs, they involve only one module, so the value of module attribute variable is certain. To keep consistence, however, we denote the corresponding module model with the module attribute variables, but there is only one element in the domain of each attribute variable.

The module attribute variable of the module model in the middle level can be denoted by the module attribute variable belonging to the module in the lower level, but the attribute variable in the same combination is irrelevant, by which the random variable in the set can't be reasoned by another variable in the same set. For the fig.1 as an instance, there are three module models, and M_1 is decomposed into M_2 and M_3. M_2 is denoted by attribute variable A_1 and A_2, and M_3 is denoted by attribute variable A_3 and A_4, by which the value of A_1 can be reasoned by the value of A_3. So, the variable A_1, A_2, A_4 or A_2, A_3, A_4 is irrelevant. M_2 and M_3 constitute M_1, so the module attribute variable of M_1 can be denoted by the attribute variable of M_2 and M_3, but the variable A_1 and A_3 are relevant. Then, the attribute variable of M_1 is A_1, A_2, A_4 or A_2, A_3, A_4.

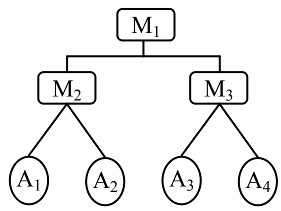

Figure1. Reasoning of module attribute variable in the middle level

Sometimes, some attributes of certain module may be relevant to attributes of another certain module, for which attributes of module are not independent with others, some attribute variables of certain module are the

© 2012 ACADEMY PUBLISHER

attribute variables of other modules at the same time. Because the attribute variable is pre-defined in product configuration, maybe they are available material parts or nonfigurative object or concept, e.g. attribute, character, etc. to express the performance and functionality of module or product, some modules or parts can be substituted by some attributes. For example, whether a personal computer (PC) has video camera, where video camera is an available attribute, and whether a PC can be set a video camera, where, video camera is a module, expresses the same meanings. So the course of product configuration is the mixed course of attribute definition and module selection to satisfy customer requirement.

Some attribute variables can express the module model, and all independent attribute variables can represent the product family. Then, we can denote the product family as $\mathbf{A}(MPF)=\{A_{a1},A_{a2},A_{d1},A_{e1},A_{e2},A_{c1},A_{c2},A_{c3}\}$. When each attribute variable in the product family is assigned with a value subjected to constraints, the specific product can be decided and configured. When each attribute variable of certain module model is assigned with a value subjected to constraints, a specific module can be decided. For example, engine of automobile, which is a module model of the automobile, can be expressed by attribute variables such as cylinder number of engine, array form of cylinder, delivery capacity, the number of valve, the maximum power output, etc. While each of them are assigned with a appropriate value subjected to some constraints of customer requirements, the specific engine of automobile can be decided and configured to satisfy customer requirements. Definition of the attribute variable should be simply under the precondition of design, manufacture and assemble constraints. The domain of the attribute variable should be fixed according to customer requirements about the attribute variables and design specifications of product structure.

Sometime, as a result of the complex structure and functionality of product and module, we can set more attributes for the module in order to search the special module case matching customer requirements more exactly. According to the domain form of module attribute variable and customer requirement classification in the next section, i.e. binary, optional, parameter which includes continuous and discrete parameter, description and explanation type, where the description and explanation type is applied to search the product family matching customer requirement, we will classify the attribute variable as binary, optional, parameter which includes continuous and discrete parameter. For example, the domain of the cylinder number of engine is $\{3,4,6,8,\cdots\}$ and the domain of color of automobile body is {white, black, blue, red, ⋯ }, where the cylinder number of engine is a discrete parameter variable, and color of automobile body is an optional variable, while the maximum power output is a continuous parameter variable.

In general, let M denote the module model, then $\{M_1,M_2,\cdots,M_m\}$ can denote the product family, where M_i $(i=1,2,\cdots m)$ is a random module model in $\{M_1,M_2,\cdots,M_m\}$, and the set of attribute variable $\mathbf{A}(M_i)=\{A_j(M_i)|j=1,2,\cdots,n_i\}$ represents M_i . For the number of attribute variables in each module model may be not equal, let n_i denote the number of attribute variable in the i th module model. Analogously, the product family can be denoted by the set of attribute variables $\mathbf{A}(MPF)=\{A_j(M_i)|i=1,2,\cdots m;\ j=1,2,\cdots,n_i\}$. If $a_j(M_i)$ implies value of the j th variable $A_j(M_i)$ of M_i , then $D_j(M_i)$ denotes the domain of $A_j(M_i)$. If $A_j(M_i)$ is a continuous parameter variable and $l_j(M_i),h_j(M_i)$ imply the minimum and maximum value of $D_j(M_i)$ respectively, then $D_j(M_i)=[l_j(M_i),h_j(M_i)]$ with restriction $l_j(M_i)<h_j(M_i)$.

III. Customer Requirement Translation

A. *Customer requirement analysis*

Customer requirement on product involves functionality, performance, appearance, price and dimension of the product, etc. That is to say, the customer requirement is denotation of unsolved precept in customer domain and it includes the essential characters of the product in need. In the course of product customization practically, for the difference in the customer themselves or the knowledge relative to product, different customers will choose different expression mode to represent their own requirement expediently. For those customers with more product knowledge, they can represent their own requirement more exactly, because they can communicate with sale staffs in technology traits and engineering characteristics. So, the requirement information can be translated to the value of module attribute variable directly, which is easy for enterprises in mass customization to deal with. For example when a customer is planning to purchase a PC, he can speak out some engineering characteristic, such as type of CPU, capacity of memory, rotate speed of HD. But for some customers who are lack of product knowledge, maybe it is hard for them to represent their requirements by choosing given options about technology traits or engineering characteristics. For they don't know the function and performance of characteristic parameter about CPU, memory, HD, etc., it is difficult for them to decide purchase correctly. In the condition, the enterprise in mass customization will allow them to represent requirements by some fuzzy or illustrative language. They can explain the main use purpose of PC and the endured price to achieve the PC product customization, but the requirement information must be deal with firstly and then be translated to value of module attribute variable indirectly. In a word, we must classify the customer requirement in order to obtain and deliver

© 2012 ACADEMY PUBLISHER

various requirements effectively and exactly. Therefore, the customer requirement can be classified as binary, optional, parameter, description and explanation type.

(1) Binary type. Binary type represents the additional function or attribute of configurable object, and formally there are two elements or values in the domain, $\{1,0\}$, where 1 imply the object in need, 0 has the opposite meaning, and the elements are mutually exclusive. That is to say, the function or attribute is dispensable in the view of customers.

(2) Optional type. Optional type defines available finite options, each of which can substitute each other but represent different character or performance of product or parts. For example, motor color may include four states: red, yellow, green and blue. Of course, the option can be set as default value. The difference between the optional and the binary is that optional represents the replaceable function or characteristic but not additional function.

(3) Parameter type. Parameter type is numerical, which can describe the quantity variety of component and numerical attribute. On the variety character, parameter type can be classified into continuous and discrete type.

(4) Description type. Usually, the customer is allowed to choose description language handily to describe attributes or function, by which the customer can adopt fuzzy description semantic manner to illuminate the product elemental characteristics, which may be product function, part function, price, appearance, structure and technique parameter, etc. So, this type can further be divided into functional and technological descriptions.

(5) Explanation type. The customer can represent the requirement by informal and natural language according to their own thinking. It can be obtained by inputting requirement sentence in communicative interface on internet.

B. Product decision tree

(1) Product decision tree illustration. When customers input their requirements in online shopping malls and comparison sites on the internet, by referring to classified customer cluster, the requirements will be classified into similar customer cluster in order to search certain product family that can satisfy the customer requirements. As well known to us, there are many attributes in product, such as function, performance, appearance, price and size, then the character set representing a certain product family is different from another. For each attributes, there exist some levels of classification, and for each level, the classification is complete and mutually exclusive. So, it is reasonable to denote certain product kind as character set, where some of the characters are in common, and others can be selected to differentiate each product family. Generally, the expert system in mass customization defines some characters that can distinguish product type distinctly, and hereby establish a decision tree for product classification. For an enterprise with many product families, it is effective to differentiate each product family by decision tree. The decision tree is similar to a tree structure of flow figure, where each non-leaf node denotes a test on the product attribute, each branch denotes a test result, i.e. the lower level node restricted to attribute value, and each leaf node denote a certain product family (shorten as PF). Taking the motor as an example, the attribute of motor type and motor price can distinguish product type distinctly, while motor type is classified as three types, i.e. ride type, footplate type and crook type. Besides, the price can be divided into low grade, medium, and top grade. Based on the knowledge on customer psychoanalysis, the product can be classified from top to down, which is a clustering of distance decreasing. The price hierarchy precedes motor type, by which the price grade is on the first level of the decision tree, motor type is on the second level, and the third level is a certain PF. In the decision tree, the classification on each level is complete and mutually exclusive.

(2) Attribute ranking. For the difference of each attribute contributing to product sale, different customer may prefer different attributes, even for the same customer, he or she may prefer different attributes in different purchasing time. If the salesmen think the customer prefer the motor use and motor structure, then the grade of motor type precede price. Of course, the method of order ranking is subjective to some extent. So, it is necessary to find out certain order ranking method fit for most customers.

Here, the information gain (IG) of each attribute is calculated and ranked in degrading order, by which the attribute of maximal IG can discriminate the products to maximal extent. The bigger the IG of the attribute is, the more obvious the discrimination degree is. Then, the level of the attribute in decision tree is higher, for which it precedes more attributes. If there are K attributes of $C_1, C_2, \cdots, C_K$ and $K < \sum_{i=1}^{m} n_i$, which can divide the products into K product families, i.e. $PF_1, PF_2, \cdots, PF_K$. If $PF_k (k = 1,2,\cdots,K)$ occurred s_k in history transaction, then the total number of this kind of products occurred is $S = \sum_{k=1}^{K} s_k$, for given sample classification, it needs $I(s_1, s_2, \cdots, s_K) = -\sum_{k=1}^{K} p_k \log_2(p_k)$, where p_k denotes the probability of random product k belonged to PF_k , which can be estimated by s_k / S . We can suppose that random attribute $C_k (j = 1,2,\cdots,K)$ has V_k different available values $\{C_k^1, C_k^2, \cdots, C_k^{V_k}\}$, where these V_k values are discrete otherwise the continuous value must be divided into exclusive range and become discrete. The attribute C_k can divide S into V_k subset $\{S_1, S_2, \cdots, S_{V_k}\}$, and in S_v $(v = 1,2,\cdots,V_k)$, there are some products which have C_k^v in attribute C_k . If C_k is a test attribute in decision tree, its subset is corresponding to a sub branch of a node. If s_{kv} equals the number of PF_k occurred in S_v , the entropy of the subset divided by attribute C_j is

© 2012 ACADEMY PUBLISHER

$E(C_k)=\sum_{v=1}^{V_k}\frac{s_{1v}+s_{2v}+\cdots+s_{Kv}}{S}I(s_{1v},s_{2v},\cdots,s_{Kv})$, in which $\frac{s_{1v}+s_{2v}+\cdots+s_{Kv}}{S}$ denotes the weight. For each given subset S_v, $I(s_{1v},s_{2v},\cdots,s_{Kv})=-\sum_{k=1}^{K}p_{kv}\log_2(p_{kv})$, where $p_{kv}=s_{kv}/S_v$ means the probability of S_v belonging to product family PF_k. Therefore, the IG of the sub branch corresponding to attribute C_k can be denoted as $Gain(C_k)=I(s_{1v},s_{2v},\cdots,s_{Kv})-E(C_k)$. The attribute of maximal IG will be selected as the first level of test attribute in decision tree to classify the given products. With the IG decreasing, the attribute of less IG will hold the higher level in decision tree. Consequently, we can decide the product family subjected to each constraints of decision tree.

According to description and explanation type of customer requirement, we can cut the sub branch of decision tree to find out the certain product family satisfying customer, the scope of retrieving product shrinking. The binary, optional, and parameter type of customer requirement can be translated into the value of attribute variable according to the product planning matrix and module deployment matrix, as referred to [2]. For easy illustration, here, we assume that the final product family PF_k is selected according to description and explanation type of customer requirement, and PF_k is denoted as PF.

(3) Translation of customer requirement to module attribute variable. As a matter of fact, the individual customer requirements of the product are the special request on the value of each attribute variable in all module models in a certain product family. The system in mass customization must set up platform based on internet for collecting information on customer requirements. The customer requirements should be understood exactly at all, and they should be inducted appropriately. After the customer requirements are restricted to certain available product family according to the description and explanation type of customer requirement, the binary, optional and parameter type of requirement should be translated into technique requirements of the product, i.e. the information about general technique character of the product, the certain technique requirements should be translated into value information about the attribute value of each module model. Here, the product in need for customer is called target product and analogously, the module that can be combined with other modules to satisfy customer requirements are called target module.

For modular configuration of product, the customer requirements should be analyzed in the first place. The approach of QFD is an important method for enterprise in mass customization to analyze customer requirements. When QFD is used to translate the customer requirements, there are some assumptions as following. There are no unreasonable or opposite requests in the customized customer requirements, but some relativity in the customer requirements is allowed. The enterprise has implemented strategy of mass customization and has established complete modules case base or components base. The cost for the enterprise to analyze individual customer requirements by QFD is very low, to some extent, i.e. it can be regarded as zero. By reference to the approach of QFD, the mapping relation of customer requirements to technique requirements, and that of technique requirements to module attributes can be established, by which the value and weights of attribute variable in each module model can be decided according to customer requirements. Now, we can illustrate product planning matrix and module deployment matrix, as shown in reference [2].

C. Module Search

After we compute out the value and weight of all attribute variable in each module models, the mapping relation of target module, which is the module that can satisfy customer requirements, and the module case in the module base will established, i.e. the similar module cases to target module will be searched in the module case base and these similar module cases belong to a GM. The similarity degree of module case and target module on the same module model can be computed quantitatively. The module case and target module belonging to same GM ensure that module case and target module have same attribute variables, from which it is reasonable to compute the similarity degree of them. For the search space is restricted in generic modules, it is helpful to reduce the search time and promote search efficiency. For example, the search algorithm on the module model M_i is as following:

Step 1. According to the requirement translation on the above, the customer requirement of the attribute variable of each module model can be gained. For random module model M_i, the set of its attribute variable is $\mathbf{A}(M_i)=\{A_1(M_i),A_2(M_i),\cdots A_{n_i}(M_i)\}$. Consequently, the customer requirement on the M_i can be translated into the value requirement of attribute variable, i.e. $\mathbf{a}^r(M_i)=(a_1^r(M_i),a_2^r(M_i),\cdots a_{n_i}^r(M_i))$ with the weight vector $\mathbf{w}^r(M_i)=(w_1^r(M_i),w_2^r(M_i),\cdots,w_{n_i}^r(M_i))$, which denote the target module.

Step 2. If a module case u is a random case of GM_i and value vector of attribute variable is $\mathbf{a}^u(M_i)=(a_1^u(M_i),a_2^u(M_i),\cdots,a_{n_i}^u(M_i))$, $u=1,2,\cdots,v_i$, where v_i denote the number of module cases in GM_i.We search all the module cases of GM_i, i.e. compare the target module and each module case of GM_i.

Step 3. The similarity of the target module and each module case in GM_i can be computed out, i.e. $s^u(M_i)=1-\sum_{j=1}^{n_i}w_j^r(M_i)\delta_j^u(M_i)$, in which $\delta_j^u(M_i)$

© 2012 ACADEMY PUBLISHER

denotes some distance value relative to the type of attribute variable, i.e. binary, optional and parameter. $\delta_j^u(M_i)$ can be computed out by the following formulas: if the attribute variable is binary type and $a_j^r(M_i)=a_j^u(M_i)$, then $\delta_j^u(M_i)=0$, else $\delta_j^u(M_i)=1$. If the attribute variable is optional type, then the distance between random two options will be provided by expert system. Here, the matrix $(\delta_{pq})_{Q\times Q}$ can denote the distance between them. $(\delta_{pq})_{Q\times Q}$ is a symmetrical matrix, i.e. $\delta_{pq}=\delta_{qp}$, Q is the number of option available to customer. δ_{pq} denotes the distance between the option p and option q, if $p\neq q$, then $\delta_{pq}\in(0,1]$; but if $p=q$, then $\delta_{pq}=0$. Summarily, $\begin{bmatrix} 0 & \delta_{12} & \cdots & \delta_{1Q} \\ \delta_{21} & 0 & & \delta_{2Q} \\ \vdots & & \ddots & \vdots \\ \delta_{Q1} & \delta_{Q2} & \cdots & 0 \end{bmatrix}$. If attribute variable is parameter type and when it is discrete variable, then the distance is $\delta[a_j^r(M_i),a_j^u(M_i))=1-\dfrac{\min[a_j^r(M_i),a_j^u(M_i)]}{\max[a_j^r(M_i),a_j^u(M_i)]}$, but when it is continuous variable, then the distance can be computed as following: we assume that $a_j^r(M_i)$ is $[b_1,b_2]$, $a_j^u(M_i)$ is $[e_1,e_2]$, and $l_j\le b_1\le b_2\le h_j, l_j\le e_1\le e_2\le h_j$. Here, b_1,b_2,e_1,e_2 is a certain value respectively. Because M_i is a candidate module to meet customer requirement, so there must be some overlap range in the range $[b_1,b_2]$ and $[e_1,e_2]$, and four relationship exist: $b_1\le e_1\le b_2\le e_2$, $e_1\le b_1\le e_2\le b_2$, $b_1\le e_1\le e_2\le b_2$ or $e_1\le b_1\le b_2\le e_2$:

$$\delta(a_j^r(M_i),a_j^u(M_i))=\delta([b_1,b_2],[e_1,e_2]) = \frac{\int_{e_1}^{e_2}\int_{b_1}^{b_2}|x-y|/(h_j-l_j)d_y d_x}{(e_2-e_1)(b_2-b_1)}$$

Then when $b_1\le e_1\le b_2\le e_2$ or $e_1\le b_1\le e_2\le b_2$,

$$\frac{(b_1-e_1)(b_2-e_1)}{2(e_2-e_1)(\beta-\alpha)}+\frac{(e_2-b_1)^3+(e_2-b_2)^3+(b_2-b_1)^3}{6(e_2-e_1)(b_2-b_1)(\beta-\alpha)};$$

But when $b_1\le e_1\le e_2\le b_2$ or $e_1\le b_1\le b_2\le e_2$,

$$\frac{b_2^2+b_1^2+b_2b_1}{3(e_2-e_1)(\beta-\alpha)}+\frac{e_2^2+e_1^2-(e_2+e_1)+(b_2+b_1)}{2(e_2-e_1)(\beta-\alpha)}.$$

We can estimate the similar degree of the target module and module case in GM_i according to $s^u(M_i)$ quantitatively. $s^u(M_i)=0$ means that the module case of GM_i is different to target module absolutely, i.e. the module case can't satisfy the customer requirement on M_i at all. $s^u(M_i)=1$ denotes that the target module and module case in GM_i are the same, i.e. the module case can fully satisfy customer requirements on M_i. If $s^u(M_i)$ is closer to 1, then it means that the module case is more similar to the target module, i.e. the module case can better satisfy the customer requirement. When $s^u(M_i)$ is compared with ε, which denotes some degree of satisfaction, the candidate modules subjected to $s^u(M_i)\ge\varepsilon$ can be searched out. If the number of the candidate modules is z_i, then the set of candidate modules on M_i is $\{M_i^1,M_i^2,\cdots,M_i^{z_i}\}$. For various and customized customer requirement, sometimes, the module cases can't satisfy the individual customer requirement, i.e. there is no module case meeting $s^u(M_i)\ge\varepsilon$. So it is necessary to modify the module with maximum $s^u(M_i)$ or set up new module case to satisfy customer requirement on M_i and then store the modified or new case into module case base. Although there are some constraints on the different attribute variables, they must subject to these constraints.

Step 4. According to above 3 steps, after computing the similarity degree of each module case in different GM and target modules and then filtering the set of candidate modules of each GM is $\{M_1^1,M_1^2,\cdots,M_1^{z_1}\}$, $\{M_2^1,M_2^2,\cdots,M_2^{z_2}\}$, …, $\{M_m^1,M_m^2,\cdots,M_m^{z_m}\}$. Designers can combine the candidate modules selected from different set to get a project of configuration. The module combination should subject to some constraints as follows. Firstly, the constraints between different module models which are global constraints and consist in the knowledge base but not inside the unit of module models, such as AND, OR, XOR, NOT. Secondly, the constraints between module model and attribute variables of another module models. Thirdly, the constraints exit between the attribute variables from different module models. When the attribute variable is defined, the attribute variable in different modules should be defined separately, but different module models are granted to share some common attribute variables in order to decrease the number of attribute variables, e.g. the relation between the color of front light and color of back light in the same automobile. Besides the values of some attribute variables can be computed out on the logic relation to some other attribute variables. For example, the logic relation between X and Y may be Y= F(X), so if Y is certain, then X can be decided, vice versa.

Module combination is not only the simple integration of different module, which is subjected to some constraint. When the customer requirement is decomposed, the module which is most near or similar to corresponding customer requirement will be retrieved and combined to achieve product configuration. The simplest way to solve module combination is enumeration, i.e try all possible module combinations, and then find out the combination meeting all constraints and compute the similar degree of each combination and customer requirement, finally range the similar degree from big to small, the satisfactory configuration will be obtained

© 2012 ACADEMY PUBLISHER

based on the price requirement. But the method is only suit for the few number modules, when the number of candidate module is bigger, and the computation will become difficult and error-prone. Some scholars have introduced the genetic algorithm to solve the problem of module combination. The merit of genetic algorithm is the relatively small search space, because the algorithm can search more feasible solutions and approximate optimal solutions, thus the search time is reduced and the efficiency of the module combination is improved. But the disadvantage of the algorithm is that genetic algorithm can't search the feasible solutions and approximately optimal solutions with the constraints of module and attributes.

As a matter of fact, product configuration for modular product is the course that customer evaluates the value of each attribute variable so as to choose exact modules to achieve combination. The module combination can be regarded as a constraint satisfaction problem, i.e. a generic module denotes a variable and the corresponding set of candidate modules denote the domain of variable. Each element in the domain denotes a module case, the constraints on module models and attribute variable restrict the legal module combination. For the modular structure of product family, we can find out the module combination subjected to above constraints from down to top in the modular structure of product family. By doing that, we can simplify the problem, i.e. reduce time and promote the efficiency of configuration. The deposition of modular product family is from top to down, in contrary, the combination of modules in model of modular product family is from down to top. The candidate modules subjected to some constraints relative to modules and attributes, in this way, many invalidated module combinations can be avoided and hence the efficiency of product configuration is promoted.

VI. FLOW CHART OF PRODUCT CONFIGURATION

The flow of modular product configuration which is expressed by attribute variables is as following (Fig.2):

Step 1, input customer requirement and search the appropriate product family which can satisfy the customer requirement.

Step 2, translate customer requirement into the specific value and weight of each attribute variable, i.e. $\mathbf{a}^r(M_i)$ and $\mathbf{w}^r(M_i)$, $i=1,2,\cdots,m$.

Step 3, estimate the product case whether it satisfy customer requirement. Product case is the product configured by the other customers in the past and stored in the product case base. If $\mathbf{a}^r(M_i)$ and $\mathbf{a}^u(M_i)$ $i=1,2,\cdots,m$ are the same, then go into the final step, or shift to the next step.

Step 4, filter out the candidate modules. According to the similarity degree of target module and module case, by which there is $(a_1^r(M_i), a_2^r(M_i), \cdots a_{n_i}^r(M_i))$ and $(a_1^u(M_i), a_2^u(M_i), \cdots, a_{n_i}^u(M_i))$ where $u=1,2,\cdots,v_i$, if there are some module cases subjecting to $s^u(M_i) \ge \varepsilon$, then choose these modules as candidate modules and shift to step6, or shift to next step.

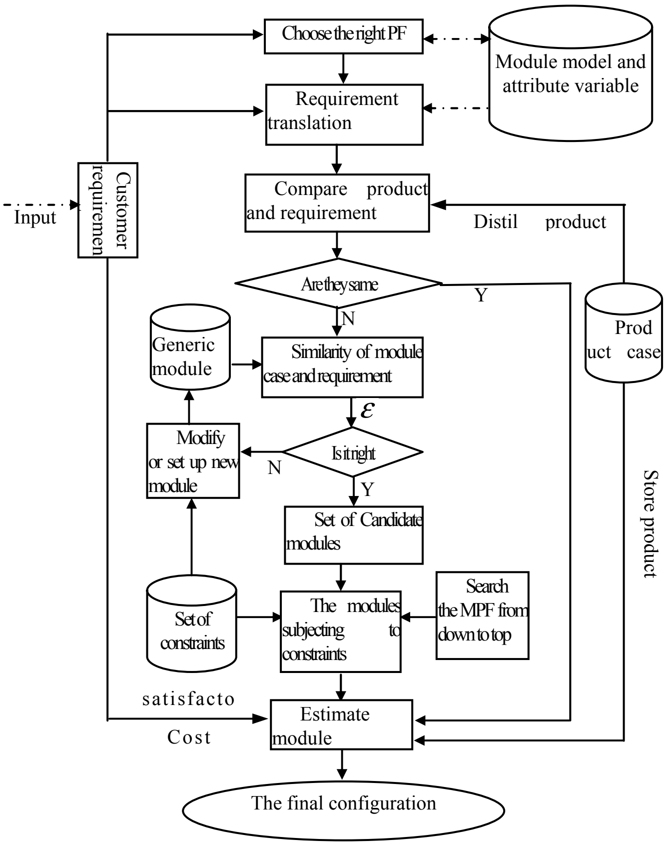

Figure2. Flow chart of product configuration

Step 5, modify modules or set up new modules. If there is no module subjecting to $s^u(M_i) \ge \varepsilon$, then choose the module with maximum $s^u(M_i)$ for modification or set up new module which will become candidate module.

Step 6, combine module. If the set of candidate modules in all GMs are decided, then check combination from down to top in the modular structure of product family and filter out the legal module combination.

Step 7, estimate the project of product configuration, because there may be many projects of legal module combinations. It is necessary to estimate each projects and make sure the best module combination according to satisfactory degree and price request of customer. The final project of configuration will be stored in the product case base for future use.

V. CONCLUSION

For various customers will adopt different method to express their own requirement, it is important to classify customer requirements and translate the various requirements into attribute information of product module. This paper classifies the customer requirements into five types: binary, optional, parameter, description and explanation. There are many attributes on product, such as function, performance, appearance, price and size. The character set representing a certain product family is

© 2012 ACADEMY PUBLISHER

different from another. Generally, the expert system in mass customization defines some characters that can distinguish product type distinctly and hereby establish a decision tree, by which it is effective to differentiate each product family. According to description and explanation type of customer requirements, we can cut the sub branch of decision tree to find out the certain product family. Then, the scope of retrieving product is shrinking.

Based on analyzing function of products that share different functionalities or common functionality but different performance, the modular structure of product family is established and the product family is decomposed into generic modules. Then, the module model represented by attribute variable is established for each generic module. In approach of QFD, the binary, optional, and parameter type of requirements can be translated into the value of attribute variable, i.e. the mapping relation of customer requirement to technique requirement, and continuously, the technique requirement to module attribute can be achieved based on the product planning matrix and module deployment matrix, and the value and weight of each attribute variable in all module models can be decided. After searching the candidate modules set which are nearest to the customer requirements on the module model, the similarity degree of module case and target module on the same module model can be calculated quantitatively. The module case and target module belonging to same GM ensure that module case and target module have same attribute variables. So, it is reasonable to compute the similarity degree between them. For the search space is restricted within generic modules, it is helpful to reduce the search time and promote efficiency. The candidate modules are combined efficiently under the constraints relative to module models and attributes, by which some combinations of invalidate module can be avoided and hence the efficiency of product configuration is promoted.

How to process natural customer language online and how to apply virtual technology and communication tool to obtain customer requirements will be our future research direction. In our work, we translate customer requirements to technique requirements or value of module attribute variable, which is set as a real number. But in the practical product family layout, technique requirements or value of module attribute variable is always a range instead of a single value. So, it is difficult to apply module deployment matrix to compute the value range of variable directly. It is worth to study the optimal decision model about product planning matrix and module deployment matrix.

ACKNOWLEDGMENT

This work was supported by Natural Science Foundation of Chongqing, China under Grant No. KJ100412.

REFERENCES

[1] B. J. Pine Ⅱ. "Mass customization: the new frontier in business competition". *Harvard Business School Press*, 1993, Boston.

[2] Y.H. Qin and G.X. Wei, "Product configuration based on modular product family modelling," *Journal of Computational Information Systems*, vol. 6, pp. 2321-2331, July, 2010.

[3] B. Dan, Y. H. Qin, and J. P. Wang, "Product configuration method based on an authentic real case and a constraint satisfaction problem," *Journal of Chongqing University (Natural Science Edition)*, vol. 3, pp.511-514, May, 2008.

[4] B. Dan, J. P. Wang, "Taxonomy model and representation approaches of customer needs information for mass customization," *Computer Integrated Manufacturing Systems*, vol.14, pp.1504-1511, August, 2008.

[5] S.T.Yan, W. J. Wu, and H.X. Huang, "The recognition method of customer's demand based on matter-element theory," *Science Technology and Engineering*, vol.10, pp.582-585, February, 2010.

[6] L. Y. SU, and M. Dong, "Service product configuration of mass customization based on ontology," *Application Research of Computers,* pp. 483-487, February, 2010.

[7] Y. Wang, and J. Lin, "Research on mass dynamic customization system," *Computer Engineering and Applications*, pp. 193-196, April, 2010.

[8] S. k. Wei, S. j. Li, and Y. Li, "Confirming weight of voice of the customer in QFD using the method of AHP," *Machinery Design and Manufacture*, pp.55-61, June, 2006.

[9] K. Fujita, "Product variety optimization under modular architecture," *Computer - Aided Design*, vol.34, pp. 953 - 965, December, 2002.

[10] K. A. Jørgensen, "Product configuration and product family modeling," unpublished working psper, 2005.

[11] J. D. Fruto, and D.Borenstein, "A framework to support customer-company interaction in mass customization environments," *Computers in Industry*, vol. 54, pp. 232-246, February, 2004.

[12] H. L. Zheng, F. Liu, "Process optimization model oriented to integrated control of CAPP/PPC," *China Mechanical Engineering*, vol.12, pp.1188-1191, August, 2006.

[13] G.X.Li, Y.Z.Ci, et al, "Study of module solution for product configuration," J*ournal of National University of Defense Technology*," vol.26, pp.85-89, May, 2004.

[14] P. J. Newcomb, B. Bras, and D. W. Rosen, "Implication of modularity on product design for the life cycle. Journal of Mechanical Design," Transactions of ASME, vol.120, pp. 483-490, March, 1998.

[15] D. Alford, P. Sackett, "Mass customization: an automotive perspective," *International Journal of Production Economics*, vol.65, pp. 99－110, February, 2000.

YanHong Qin Sichuan Province, China. Birthdate: Nov., 1981. Work as associate professor in Management School of Chongqing Jiaotong University, China. Her research interests include on mass customization, product configuration, etc.

GuangXing Wei Chongqing Province, China. Birthdate: Aug., 1977. Work as professor in Management School of Chongqing Jiaotong University, China. His research interests include on mass customization, supply chain management, etc.

© 2012 ACADEMY PUBLISHER