10.6977/IJoSI.201007_1(2).0003
Yao-Tsung Ko / Int. J. Systematic Innovation 27-43 (2010)

An Innovative Matrix-Based Approach for Designing Product Variety

Yao-Tsung Ko
Department of Industrial Design ,Tunghai University, Taiwan
Corresponding author, E-mail: a614_ko@hotmail.com
(Received 4 May 2010; final version received 20 September 2010)

Abstract

New product development (NPD) and innovation are key factors that affect a company's long-term survival and growth. The design process is an important stage in new product development (NPD). Based on graph theory and the weighting concept, this paper presents a Quantified Design Structure Matrix (QDSM) which is a systematic planning method of optimizing design priorities and product architecture for managing product variety from an informational structure perspective. Focusing on product variety and the design process in concurrent engineering (CE), the planning model is divided into two phases: global planning and local planning. The proposed method helps designers optimize the design planning and plan better design strategies for product variety. A case study is used to illustrate this method. The results verify that designers may concurrently create variant design solutions in a product family that can meet different market needs without extra effort being spent on redundant design loops.

Keywords: new product development (NPD), graph theory, concurrent engineering (CE), quantified design structure matrix (QDSM)

1. Introduction

Design for variety (DFV) is a design strategy and methodology that helps designers reduce the impact of variety on the life-cycle costs and time of a product (Martin and Ishii 1997). Various investigations have explored issues dealing with the strategic benefits of developing product platforms and the management of product families.

Suh (1990) viewed product variety as the proper selection of design parameters that satisfy variant functional requirements. Fujita and Ishii (1997) formulated the task structure of product variety design. Erens (1996) developed product variety under functional, technological, and physical domains. Martin and Ishii (1997) proposed Design for Variety (DFV), which is a series of methodologies with quantifying indices for reducing the influence of product variety on product life-cycle costs, and thus helping design teams develop decoupled product architectures. These studies have established a basis for product variety management.

Product variety is another orthogonal axis against the design process and product architecture and requires strategic design synthesis. Second, although all these studies provided some insight into the dependent relationships of a complex product for product variety design, they failed to expose and explore the logic behind these dependencies. Moreover, the operation process of the proposed tools is complex and inefficient. The tools are not easily applied to computational programming. Therefore, this paper focuses on optimizing product architecture by identifying the attributes of product components for design variety and on design priorities of product components for concurrent engineering (CE).

To deal with this problem, this paper proposes a structural matrix-based method called Quantified Design Structure Matrix (QDSM) based on the design structure matrix (DSM) (Steward 1981). For instance: (1) the traditional path searching method (Weinblatt 1972) adopted in the partitioning procedure is computationally inefficient; it is difficult to solve large design matrix. (2) Although many researchers (Kusiak and Wang 1993, Rogers 1989) have tried to improve the tearing algorithm, no optimal method exists for tearing. (3) The dependency strength between two product components cannot be really reflected using a binary matrix with "1 "and "0 ". The information is insufficient to dispose the coupled components for further analysis. Thus, this study attempts to solve these problems using the QDSM model.

QDSM can reduce complex system interactions into a logically oriented graph. This paper employs QDSM to establish a hierarchical component interaction structure, which can help designers determine component commonality, variety, and design priorities for design strategies. QDSM can help designers develop a product family. We expect that paper can provide a planning model for new product design and that the results can help designers concurrently create variant design solutions in a product family that can meet different market needs without extra effort being spent on redundant design loops.

2. Methodology: information structure analysis

2.1 Extended directed graph (EDG)

Once decomposed, the design process and product architecture can be described as a directed graph based on graph theory (Roberts 1976). The directed graph consists of a set of nodes, representing the design components, and a set of directed lines connecting these nodes. The directed lines or linkages reflect a dependency or a relationship between the connected components. Assume that $G = \langle V, E \rangle$ is a directed graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of nodes denoting n components, and $E = \{e_1, e_2, \ldots, e_n\}$ is a set of directed lines denoting the path and direction of information linkages. Each element of E corresponds to two nodes in V. However, there are some disadvantages to directed graphs. For instance: (1) Simple relationships. Most directed graphs can only describe sequential relationships. However, there are also parallel relationships and coupled relationships in the design process and product architecture. A directed graph cannot describe these relationships completely. (2) Scattered structure and difficulty to operate in computer language. Since directed graph models are described in a graphical and illogical way, it is not convenient to work with them on a computer. (3) The dependency strength between the product components cannot be described. This is a disadvantage when decomposing the design components, in particular, disposing coupled components for design priorities. (4) The hierarchical relationships of the design components cannot be clearly represented. An excellent plan and strategy for the design process and product architecture is thus difficult to make. (5) Furthermore, if information flows are complex or information content is great, the directed graph model will be messy.

Thus, we propose an extended directed graph (EDG) to present the original information model of a complex design process by quantifying the dependency strength between the product components. Furthermore, mapping EDG to DSM is proposed to describe a complex design process and product architecture. We are able to obtain an excellent plan for design priorities and product variety after analyzing the information flows hidden in DSM. In the next subsection, we introduce the basic theory of DSM.

2.2 Design structure matrix (DSM)

According to graph theory, the relationships between design components can be mapped to a matrix. The matrix is called a Design Structure Matrix (DSM) (Steward 1981), in which the rows and columns correspond to the design components. A DSM associated with a directed graph is a binary square matrix with m rows and columns, and n non-zero elements, where m is the number of nodes and n is the number of directed lines connecting these nodes in the directed graph. If there exists a directed line from node j to node i, then the value of element a_{ij} (column j, row i) is unity (or marked with an X). Otherwise, the value of the element is zero (or left empty). The DSM can be defined as follows:

Definition 1. Let A be a DSM with a $n \times n$ square matrix, where n denotes the number of components. The DSM is a binary Boolean matrix $A = [a_{ij}]_{n \times n}$. Its elements, a_{ij}, can only be “0” or “1”. Thus, it can be defined as:

$$a_{ij} = \begin{cases} 0 & (i = j \;\; or \;\; a_j \not\to a_i) \\ 1 & (a_j \to a_i) \end{cases} \qquad (1)$$

In the matrix, the element $a_{ii} = 0$ is on the diagonal. “$a_j \to a_i$” denotes that component a_j input information to component a_i. Then, $a_{ij} = 1$, otherwise $a_{ij} = 0$. Figure 1 shows a classical DSM.

The matrix representation of a directed graph provides a systematic mapping among design components that is clear and easy to read regardless of size. It can be shown that an empty row represents a node without inputs, and that an empty column represents a node without outputs. Off-diagonal marks in a single row of the DSM represent all of the components whose output is required to perform the component corresponding to that row. Similarly, reading down a specific column reveals which components receive information from the component corresponding to that column. If one interprets the component ordering in the matrix as the execution sequence, then marks below the diagonal represent forward information transfer to later (i.e. downstream) components. This kind of mark is called a forward mark or a forward information link. Marks above the diagonal depict information fed back to earlier listed components (i.e. feedback mark or information link) and indicate that an upstream component depends on a downstream component. Figure 2 (Smith 1992) shows three configurations that characterize a system mapped from a directed graph to a DSM representation.

Backward

Feed forward

$$\begin{array}{c|c} & \begin{array}{cccc} a_1 & a_2 & \cdots & a_n \end{array} \\ \hline \begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_n \end{array} & \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \end{array}$$

Fig. 1. Design Structure Matrix.

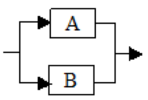
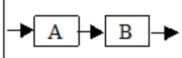
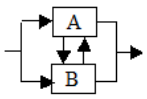
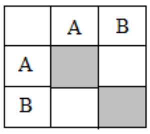
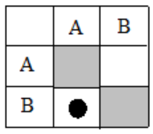
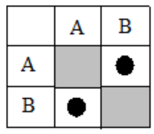
Three Configurations that Characterize a System			
Attribution	Independent	Dependent	Interdependent
Relationship	Parallel	Sequential	Coupled
Graph Representation			
DSM Representation			

Fig. 2. Characterizing a system by DSM and directed graph representation.

2.3 Mapping from EDG to QDSM

There are many vague and uncertain relationships within design components when product configurations are considered. The traditional DSM cannot express fuzzy and uncertain interdependent relationships with "1" and "0". We utilize a simple weighting method to represent the complete dependency structure profile and dependency uncertainty of the design process and product architecture.

We not only use the directed lines to describe the relationship between the product components, but also quantify the dependency strength between product components in EDG. In order to assign weights to the relationships between design components, we apply a weighting scale with linguistics variables to define the degree of the dependency strength. After mapping EDG to DSM, the evaluation value a_{ij} of the dependency strength will be used instead of a "1" in DSM. The matrix will become a numerical DSM. It is called a quantified design structure matrix (QDSM).

Based on the weighting concept, we can employ linguistics variables to describe the degrees of the dependency strength within the product components. A variable is represented using a linguistic variable V, which is based on the linguistic scale: S_v = EL, VL, L, M, H, VH, EH where EL: Extremely Low (0); VL: Very Low (0.1); L: Low (0.3); M: Medium (0.5); H: High (0.7); VH: Very High (0.9); and EH: Extremely High (1). The element a_{ij}

presents quantitatively the dependency strength between component a_i and component a_j and is defined as follows:

$$a_{ij} = \begin{cases} 0 & (i = j \quad or \quad a_j \nrightarrow a_i) \\ K & (a_j \rightarrow a_i) \end{cases} \qquad (2)$$

where $K \in \{0,\ 0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.9,\ 1\}$.The element a_{ij} is associated with a real number in the interval [0; 1]. To establish the universal weighting scale of linguistics variables, the linguistic variable set R_v is defined as:

$$R_v = \left\{ \frac{0}{extremely\ low}, \frac{0.1}{very\ low}, \frac{0.3}{low}, \frac{0.5}{medium}, \frac{0.7}{high}, \frac{0.9}{very\ high}, \frac{1}{extremely\ high} \right\} \qquad (3)$$

We can obtain an EDG by assigning weights to the relationships between each pair of components; the EDG can then be mapped to QDSM for further analysis. Figure 3 shows the mapping procedure from EDG to QDSM.

3. Re-engineering process based on QDSM

An important challenge of CE is making sound decisions at very early stages of product development where budgeted costs are low. All components in the downstream design should be considered at early stages, so that the potential problems can be found as early as possible.

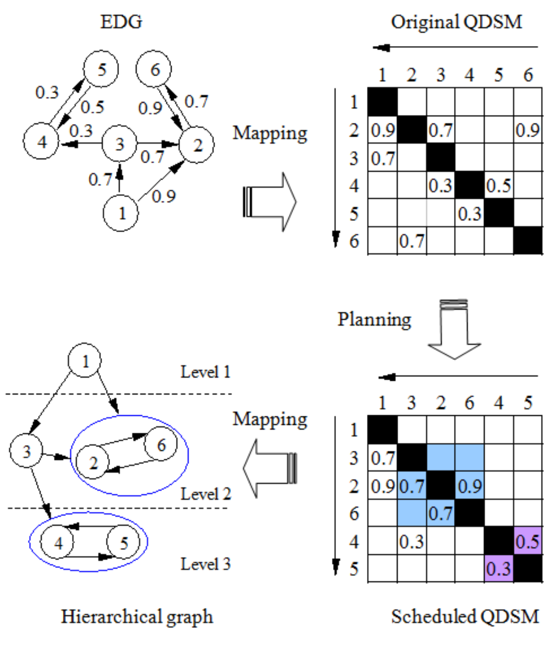


Fig. 3. Mapping from EDG to QDSM.

To achieve its aim, concurrent engineering uses the small local iterations to avoid the large scope iterations of the traditional sequential design process. From a microcosmic view, the early stages of concurrent engineering are focused on coupled phases which often arise from the small local iterations and can be expressed by the coupled relationship model. From a macroscopic view, the structure of the decoupled circuitry serves as an ideal model of the concurrent design process which emphasizes “do it right first”. If one interprets the component ordering in

QDSM as the design sequence, the elements $a_{ij} = 1$ ($i > j$) below the diagonal represent the forward information transfer to later (i.e. downstream) components; and the elements $a_{ij} = 1$ ($i < j$) above the diagonal depict information fed back (or iteration) to earlier (i.e. upstream) components. Thus, the QDSM of the ideal concurrent design process and optimal product architecture will become a lower triangular form. However, a complex design process and product architecture include many information loops in coupled mode that lead to iterations of product components, delaying the design period. The purpose of re-engineering is to reduce the iteration time as much as possible. Because the above QDSM is based on components, its re-engineering can be realized by the partitioning and tearing of QDSM. In the next subsection, we introduce the proposed planning method based on QDSM. The method includes two phases: global planning and local planning.

3.1 Global planning of the design process

QDSM can be considered as the transpose of the incidence matrix corresponding to EDG. The partitioning algorithm is adopted to identify the coupled components. The upper-diagonal marks of QDSM signify feedback and iterations of components. The purpose of partitioning is to transform QDSM into a lower triangular matrix in the global planning phase of the design process and product architecture. The Interpretative Structural Modeling (ISM) method (Warfield 1973, 1990) is adopted to realize and improve the partitioning algorithm of QDSM in the global planning phase. There are three main steps in the global planning phase: (1) sorting independent components, (2) identifying coupled components, and (3) arranging the ranks of the uncoupled components. We first introduce some definitions which will be used in the partitioning algorithm. The procedures of the partitioning algorithm are as follows:

Procedure 1. Sorting independent components.

The purpose of partitioning is to push forward the process of each component and recognize the coupled components in the design process. It is a gradually decreasing process. The gradually decreasing analysis of partitioning includes the sorting of independent components and also the recognition of coupled components. An independent component is defined in Definition 2.

Definition 2: In the fuzzy design structure matrix A, the components with a zero row-sum or a zero column-sum are called independent components. We take the condition $a_{ij} \in R$, if $\sum_{j=1}^{n} a_{ij} = 0$ or $\sum_{i=1}^{n} a_{ij} = 0$, and then we define the corresponding component of a_i, a_j as the independent component.

In this paper, we develop a simple and efficient procedure for finding a logical order of the components using the matrix form when no loops exist. The proposed algorithm starts by finding the input-degree of component i(Ii), which is the row sum of that component. Then, we rank the component with a zero row-sum, if it exists, to be the first component in the QDSM. This component with all its corresponding marks is deleted from the QDSM and the above process is repeated to find another component with a zero row-sum. If there are no components with a zero row-sum and the QDSM is not empty, then the design process contains cyclic flows of information and the procedure is terminated. Similarly, if we find a component with a zero column-sum, we can place it to the last position in the QDSM.

Procedure 2. Identify the coupled components.

The problem of identifying the coupled components set is translated into the problem of seeking strongly connected components in QDSM. Based on the algebraic technique of ISM, we can deduce a reachable matrix and a strongly connected matrix for identifying the coupled components from the incidence matrix of QDSM.

Definition 3 (Warfield 1990, Xiao 1997). Let A be the incidence matrix of QDSM and let In be the n-dimensional Boolean unity matrix; then, the transitive closure of $(A \cup I_n)^n$ is defined as the reachable matrix P of this QDSM.

The reachable matrix $P=(A\oplus I_n)^n=(p_{ij})_{n\times n}$ is deduced from incidence matrix A if a Boolean n-multiple product of $A\oplus I_n$ uniquely converges to P for all integers $n>n_0$, where n_0 is an appropriate positive integer, I_n is a n-dimensional Boolean unity matrix, and $\oplus$ is the logic Sum operator in Boolean sense (Warfield, 1990). Matrix P represents all direct and indirect linkages between components. Relationship transitivity is a basic assumption in ISM.

Definition 4 (Xiao 2001). Let Q be a strongly connected matrix. Matrix Q is the strongly connected judgment matrix of the reachable matrix P. Q is defined as follows:

$$Q=P\bigcap P^T=\begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & {}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}\bigcap\begin{bmatrix} p_{11} & p_{21} & \cdots & p_{n1} \\ p_{12} & p_{22} & \cdots & p_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ p_{1n} & p_{2n} & \cdots & p_{nn} \end{bmatrix}=\begin{bmatrix} p^2{}_{11} & p_{12}\cdot p_{21} & \cdots & p_{1n}\cdot p_{n1} \\ p_{21}\cdot p_{12} & p^2{}_{22} & \cdots & p_{2n}\cdot p_{n2} \\ \cdots & \cdots & \ddots & \cdots \\ p_{n1}\cdot p_{1n} & p_{n2}\cdot p_{2n} & \cdots & p^2{}_{nn} \end{bmatrix} \quad (4)$$

where the matrix $P=(p_{ij})_{n\times n}$ is reachable, and P^T is the transpose of P. Matrix Q is denoted as

$$P\bigcap P^T=(p_{ij})_{n\times n}=(p_1,p_2,\cdots,p_n)^T \quad (5)$$

where p_i is a n-dimensional row vector. Let the set composed by any of the unequal p_i be $\{p_1',p_2',\cdots,p_m'\}$ $(1\le m\le n)$, Then:

(1) The number of coupled components in QDSM is $m'(m'\le m)$, where m' is the total number of row vectors that have at least one component whose value is equal to 1 in $\{p_1',p_2',\cdots,p_m'\}$.

(2) If p_i' is the row vector that has at least one component whose value is equal to 1 and all the components whose value is equal to 1 are $p_{ik1},p_{ik2},\cdots,p_{ikp},(2\le p\le n)$, then $C=\{C_{ik1},C_{ik2},\cdots,C_{ikp}\}$ is a coupled components set.

If the path is reachable from component i to component j, then $p_{ij}=1$. If the path is reachable from component j to component i, then $p_{ji}=1$. Thus, component i and component j are reachable from each other, if and only if $p_{ij}\cdot p_{ji}=1$. In matrix Q; if the non-zero elements of the ith row are in the $j1$th, $j2$th…, $j$kth columns, then, component i, component $j1$, component $j2$,…, component jk form a strongly connected component. The components corresponding to these components are in a coupled set.

Procedure 3. Arrange the ranks of the uncoupled components.

Definition 5 (Cui *et al.* 1997). The reachable matrix P becomes a reduced matrix P', if every coupled component set is merged into one component, and the rows and columns corresponding to the coupled component set have been merged into one row and column.

Definition 6 (Xiao 1997). Let $P'=(p_{ij}')_{m\times m}$ be the reductive matrix of a QDSM. $P'E_{l-1}=(p_1,p_2,\cdots,p_m)^T$, where $l\ge 1$, $1\le m\le n$, the m-dimension vector $E_0=(1,1,\cdots,1)^T$, $E_l=(e_1,e_2,\cdots,e_m)^T$, where

$$e_i=\begin{cases} 0, & p_i\in\{0,\ 1\}; \\ 1, & p_i\notin\{0,\ 1\}; \end{cases} \quad (i=1,\ 2,\ \cdots,m). \quad (6)$$

Then, for component C_i, $p_i=1$ is the necessary and sufficient condition of $L_l=\{C_i\}$,where L_l means that the level of component C_i is l in QDSM.

Definition 6 can be easily realized on a computer to arrange the level of coupled components sets. According to the above method, the partitioned QDSM of the design flow can be easily obtained. The execution of design components becomes sequential. The rank of the design components indicates the priority level of all the components. The design process is in a lower triangular form, and there are no large-scale or whole iterations.

3.2 Local planning of the design process

Creating a lower triangular form by partitioning avoids large-scale iterations, but loops in coupled blocks still exist. It is thus necessary to break apart these loops and plan them. To reduce the feedback and iterations caused by coupled information flow, we use a removing coupling method called tearing to make certain the original iteration sequence of coupled components by analyzing the relationships between components. The basic principle of the tearing algorithm is to cut off the loops at the weakest point and to design the components with the least information-dependent intensity. Here, we propose a simple and efficient method to eliminate the coupled component sets.

No optimal method exists for tearing, but many researchers (Kusiak and Wang 1993, Rogers 1989) have identified two important criteria for tearing procedures.

(1) Minimal number of tears: the motivation behind this criterion is that tears represent an approximation or an initial guess to be used; we should reduce the number of these guesses.

(2) Confine tears to the smallest blocks along the diagonal: the motivation behind this criterion is that if there are to be iterations within iterations (i.e. blocks within blocks), these inner iterations are performed more often. Therefore, it is desirable to confine the inner iterations to a small number of components.

In this paper, we propose a simple and efficient method to decouple the coupled components sets. We now look at tearing each block separately. For each block in the partitioned QDSM, the block information input-degrees (*IIi*) and the block information output-degrees (*IOi*) are calculated for all the components within that block. Note that *IIi* and *IOi* are the row and column sums of component *i*, respectively; however, only the subset of components and marks contained within the block is considered. Next, we calculate the ratio *Ri* = *IIi*/*IOi*, which is a relative importance index. Another issue to consider is the relative importance of input and output information. In a QDSM, the elements above the diagonal denote the iteration of design information. The feedback information of more downstream components will cause more large-scale iterations. We want to have the least amount of feedback information during the design process in concurrent design. In order to identify the weights for the element $E_{ij}(i < j \le n)$ above the diagonal, we can adopt the related distance from E_{ij} to the corresponding element E_{ii} on the diagonal to denote the relative importance. The weight of the element $E_{ij}(i < j \le n)$ above the diagonal can be defined as $W_a = |j - i|$. For element $E_{ij}(j < i \le n)$ below the diagonal, we define its weight as $W_b = 1$. Both *IIi* and *IOi* can be defined as follows:

$$II_i = \sum_{j=1}^{i-1} C_{ij} \cdot W_b + \sum_{j=i+1}^{n} C_{ij} \cdot W_j \qquad (i, j \le n) \qquad (7)$$

$$IO_i = \sum_{i=1}^{i-1} C_{ij} \cdot W_i + \sum_{i=j+1}^{n} C_{ij} \cdot W_b \qquad (i, j \le n) \qquad (8)$$

where, *n* denotes the number of coupled components, and W_i and W_j are the weights corresponding to the elements. The steps of the tearing procedure are listed as below:

(1) Calculate the *IIi* and *IOi* of component *i* , where *i* = 1 to *n*.

(2) Calculate the ratio *Ri* = *IIi*/*IOi*.

(3) Compare each *Ri* corresponding to component *i*. Component *i* with the minimum *Ri* value is scheduled first within the block, since it requires minimum input and delivers maximum output.

(4) After choosing the top-priority component, the scheduled component and all its corresponding marks are removed from the block. Next, we check if the loop was broken by the removal of the scheduled component using the above procedure. If an information loop is encountered again within the block, the process of finding new *Ri* values is repeated. After ranking all the components within a block, we tear all the feedback marks in the block.

4. A case study

4.1 Object product

This study employs the variant design of a PLC (Power Line Communication) product to illustrate the proposed methodology. This case study involves a Taiwanese electronic appliances manufacturer (Company A). Ninety percent of the products of this company are Original Design Manufactured (ODM), and are mainly exported to America, Europe and Japan. Based on their experiences and manufacturing technologies, Company A aims to develop a series of products to simultaneously meet the requirements of each segmented market, and to provide variety in mass customization.

4.2 Identify market-driven variety

At present, the position of the PLC products of Company A belongs to cost driven market segmentation with unrefined style and low-tech. Company A hopes that their PLC product can be developed toward high-value market segmentation with high-style and high-tech in the future (Figure 4). In this case study, market planning is performed by the product development team, which includes the marketing, planning, and design departments of Company A. The market planning is aimed at two different markets (technology variety) with two different appearances (style variety), so four products need to be concurrently developed.

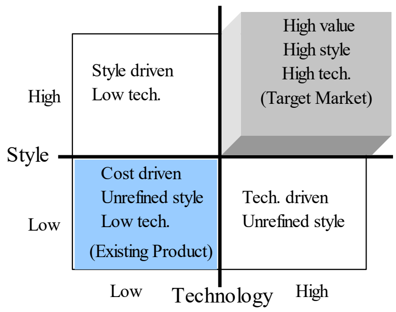


Fig. 4. Market segmentation and position map of PLC Product.

Finally, the design team identifies the initial product specifications (Table 1) for concurrently developing four variant PLC products for the different segmented markets.

Product Spec.	Product 1	Product 2	Product 3	Product 4
Main Function	200 Mega PLC	85 Mega PLC	55 Mega PLC	55 Mega PLC
Extensional Functions	Audio, Video, VoIP	Audio, Video	X	X
Security device	Electronic Key	X	Electronic Key	X

Table 1. Initial PLC product specifications.

Based on the existing PLC product of Company A and the initial product specifications, the design team identifies all required physical components, as shown in Table 2.

10.6977/IJoSI.201007_1(2).0003
Yao-Tsung Ko / Int. J. Systematic Innovation 27-43 (2010)

Table 2. Components list for PLC product.

1. Key PCBA	9. Functional Base Cover
2. Functional PCBA	10. Power Plug
3. Main System PCBA	11. Power Button
4. Key Front Cover	12. Key Button
5. Key Back Cover	13. Led Lens
6. Main Top Cover	14. Main IO Plate
7. Main Base Cover	15. Functional UI Plate
8. Functional Top Cover	

4.3 Build QDSM for PLC product

Next, we represent the interdependent relationships of 15 product components from an EDG mapping to a 15 x 15 square QDSM using the proposed weighting method (Equation 3) which assigns weights to the dependency strength between each pair of product components. This numerical DSM becomes a QDSM (Figure 5).

Part Name	No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Key PCBA	1					0.3							0.2			
Functional PCBA	2									0.3						
Main System PCBA	3							0.5				0.2		0.2		
Key Front Cover	4	0.5		0.5		0.5					0.6		0.3			
Key Back Cover	5	0.5			0.3											
Main Top Cover	6			0.8				0.4				0.3		0.2		
Main Base Cover	7			0.9			0.6									
Functional Top Cover	8		0.6	0.8						0.2						
Functional Base Cover	9		0.5						0.6							0.4
Power Plug	10			0.9			0.5									
Power Button	11			0.8			0.7									
Key Button	12	0.8			0.9											
Led Lens	13			0.9			0.8									
Main IO Plate	14			0.4				0.3								
Functional UI Plate	15		0.8												0.7	

Fig. 5. Original QDSM for PLC product components.

4.4 Global planning

4.4.1 Identifying coupled components sets

The original QDSM can be clustered and reordered using the improved partitioning algorithm illustrated in section 3.1. The incidence matrix, reachable matrix, and strongly connected matrix can be deduced. First, we can transform the original QDSM into a binary Boolean matrix. The matrix is called incidence matrix A and is shown below.

$$A=\begin{bmatrix}
0&0&0&0&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&1&0&1&0&0\\
1&0&1&0&1&0&0&0&0&1&0&1&0&0&0\\
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&1&0&1&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
1&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&1&0
\end{bmatrix}$$

Second, according to Definition 3, the reachable matrix P can be obtained as below.

$$P=\begin{bmatrix}
1&0&1&1&1&1&1&0&0&1&1&1&1&0&0\\
0&1&1&0&0&1&1&1&1&0&1&0&1&1&1\\
0&0&1&0&0&1&1&0&0&0&1&0&1&0&0\\
1&0&1&1&1&1&1&0&0&1&1&1&1&0&0\\
1&0&1&1&1&1&1&0&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&0&0&1&0&1&0&0\\
0&0&1&0&0&1&1&0&0&0&1&0&1&0&0\\
0&1&1&0&0&1&1&1&1&0&1&0&1&1&1\\
0&1&1&0&0&1&1&1&1&0&1&0&1&1&1\\
0&0&1&0&0&1&1&0&0&1&1&0&1&0&0\\
0&0&1&0&0&1&1&0&0&0&1&0&1&0&0\\
1&0&1&1&1&1&1&0&0&1&1&1&1&0&0\\
0&0&1&0&0&1&1&0&0&0&1&0&1&0&0\\
0&0&1&0&0&1&1&0&0&0&1&0&1&1&0\\
0&1&1&0&0&1&1&1&1&0&1&0&1&1&1
\end{bmatrix}$$

Third, according to Definition 4, the strongly connected matrix Q can be deduced as follows:

$$Q=P\cap P^T=\begin{bmatrix}
1&&&1&1&&&&&&&1&&&\\
&1&&&&&&1&1&&&&&&1\\
&&1&&&1&1&&&&1&&1&&\\
1&&&1&1&&&&&&&1&&&\\
1&&&1&1&&&&&&&1&&&\\
&&1&&&1&1&&&&1&&1&&\\
&&1&&&1&1&&&&1&&1&&\\
&1&&&&&&1&1&&&&&&1\\
&1&&&&&&1&1&&&&&&1\\
&&&&&&&&&1&&&&&\\
&&1&&&1&1&&&&1&&1&&\\
1&&&1&1&&&&&&&1&&&\\
&&1&&&1&1&&&&1&&1&&\\
&&&&&&&&&&&&&1&\\
&1&&&&&&1&1&&&&&&1
\end{bmatrix}$$

From matrix Q, we can find that the strongly connected components include $\{C_1, C_4, C_5, C_{12}\}$, $\{C_2, C_8, C_9, C_{15}\}$, $\{C_3, C_6, C_7, C_{11}, C_{13}\}$, $\{C_{10}\}$, and $\{C_{14}\}$. The coupled components sets are $\{C_1, C_4, C_5, C_{12}\}$, $\{C_2, C_8, C_9, C_{15}\}$, and $\{C_3, C_6, C_7, C_{11}, C_{13}\}$.

According to Definition 5, the reduced matrix P' of the reachable matrix P is:

$$P'=\begin{array}{c}
\begin{matrix} s_1 & s_2 & s_3 & s_4 & s_5 \end{matrix}\\
\begin{bmatrix}
1&&1&1&\\
&1&1&&1\\
&&1&&\\
&&1&1&\\
&&1&&1
\end{bmatrix}
\end{array}$$

where s_1 denotes coupled set $\{C_{1,4,5,12}\}$, s_2 denotes coupled set $\{C_{2,8,9,15}\}$, s_3 denotes coupled set $\{C_{3,6,7,11,13}\}$, s_4 denotes $\{C_{10}\}$, and s_5 denotes $\{C_{14}\}$. Based on the Definition 6, the order levels of all product components can be deduced as:

$E_0=(1,\ 1,\ 1,\ 1,\ 1)^T$, $P'E_0=(3,\ 3,\ 1,\ 2\ ,2)^T$, $L_1=\{C_{3,6,7,11,13}\}$.

$E_1=(1,\ 1,\ 0,\ 1,\ 1)^T$, $P'E_1=(2,\ 2,\ 0,\ 1,\ ,1)^T$, $L_2=\{C_{10},\ C_{14}\}$.

$E_2=(1,\ 1,\ 0,\ 0,\ 0)^T$, $P'E_2=(1,\ 1,\ 0,\ 0,\ 0)^T$, $L_3=\{C_{1,4,5,12},\ C_{2,8,9,15}\}$

According to the above order levels of product components, the re-engineered QDSM can be obtained as shown in Figure 6.

Level	Part Name	No.	3	6	7	11	13	10	14	1	4	5	12	2	8	9	15
1	Main System PCBA	3	■		0.5	0.2	0.2										
1	Main Top Cover	6	0.8	■	0.4	0.3	0.2										
1	Main Base Cover	7	0.9	0.6	■												
1	Power Button	11	0.8	0.7		■											
1	Led Lens	13	0.9	0.8			■										
2	Power Plug	10	0.9	0.5				■									
2	Main IO Plate	14	0.4		0.3				■								
3	Key PCBA	1								■		0.3	0.2				
3	Key Front Cover	4	0.5					0.6		0.5	■	0.5	0.3				
3	Key Back Cover	5								0.5	0.3	■					
3	Key Button	12								0.8	0.9		■				
3	Functional PCBA	2												■		0.3	
3	Functional Top Cover	8	0.8											0.6	■	0.2	
3	Functional Base Cover	9												0.5	0.6	■	0.4
3	Functional UI Plate	15							0.7					0.8			■

Fig. 6. A partitioned QDSM for PLC product components.

4.5 Local planning

We next decouple the coupled components sets. We take coupled block 1 as an example. According to section 3.2, we can calculate the ratio index *Ri* = *IIi*/*IOi* as shown in Table 3.

Table 3. The calculation of the tearing procedure.

Activity i	*II i*	*IO i*	*R i*	*Rank*
C_3	2.4	3.4	0.71	1
C_6	2.4	2.1	1.14	3
C_7	1.5	1.4	1.07	2
C_{11}	1.5	1.2	1.25	5
C_{13}	1.7	1.4	1.21	4

From the above analysis, we can obtain the new order of the product components of the coupled set from [*C3, C6, C7, C11, C13*] to $[C_3 \Rightarrow C_7 \Rightarrow C_6 \Rightarrow C_{13} \Rightarrow C_{11}]$. The other coupled sets can be decoupled in the same manner. After the tearing procedure, we can obtain the final component sequence in a QDSM (Figure 7); it can be mapped to a hierarchical graph automatically. Figure7 shows the interaction matrix after an appropriate rearrangement of the order. Three chunks form in the PLC product, namely C1: Main module, C2: Key module, and C3: Functional Module. The precedence of the three chunks is determined by the inter-chunk interactions. Based on concurrent engineering, we can assign these three modules to three designers, respectively, to reduce the product development time. Finally, according to the
hierarchical graph, we can figure out the optimal design process and product architecture for PLC product development.

The identified relationships represent design constraints and incidence between product components that cope with the design knowledge of the specific product. The bottom row in Figure 7 shows the S value (sum of rows), indicating the degree to which each component influences others and the third-last column lists the R value (sum of columns), indicating the degree to which each component is influenced by the others. The last two columns of Figure 7 list the values of (S + R) and (S –R), respectively. The (S + R) value indicates the sum of interactions of a component, including the ‘supplying’ and ‘requiring’ interactions. The (S – R) indicates the difference between the influencing and influenced interactions of a component; a higher value indicates that the component is dominant. For example,

Figure 7 shows that the highest (S + R) value is 6.9 for component 3, namely the Main System PCBA. The two highest (S – R) values are 1.6 and 5.1 for component 2 and component 3, respectively, namely the Functional PCBA and Main System PCBA. Figure 8 shows the (S – R) plotted against (S + R). This graph is an overall indicator of how interactive/dominant a component is. For example, a high (S – R) value indicates that changes to the component have a relatively high propagation strength. A high (S + R) value indicates an interface component; changes to which affect or refer to numerous components.

Chunk	Module	Level	Component	No.	3	7	6	13	11	10	14	1	5	4	12	2	8	9	15	*R-value*	*S+R*	*S-R*
C1	Main Module	1	Main System PCBA	3	■	0.5		0.2	0.2											0.9	*6.9*	*5.1*
		1	Main Base Cover	7	0.9	■	0.6													1.5	*2.7*	*-0.3*
		1	Main Top Cover	6	0.8	0.4	■	0.2	0.3											1.7	*4.3*	*0.9*
		1	Led Lens	13	0.9		0.8	■												1.7	*2.1*	*-1.3*
		1	Power Button	11	0.8		0.7		■											1.5	*2*	*-1*
		2	Power Plug	10	0.9		0.5			■										1.4	*2*	*-0.8*
		2	Main IO Plate	14	0.4	0.3					■									0.7	*1.4*	*0*
C2	Key Module	3	Key PCBA	1								■	0.3		0.2					0.5	*2.3*	*1.3*
		3	Key Back Cover	5								0.5	■	0.3						0.8	*1.6*	*0*
		3	Key Front Cover	4	0.5					0.6		0.5	0.5	■	0.3					2.4	*3.6*	*-1.2*
		3	Key Button	12								0.8		0.9	■					1.7	*2.2*	*-1.2*
C3	Functional Module	3	Functional PCBA	2												■		0.3		0.3	*2.2*	*1.6*
		3	Functional Top Cover	8	0.8											0.6	■	0.2		1.6	*2.2*	*-1*
		3	Functional Base Cover	9												0.5	0.6	■	0.4	1.5	*2*	*-1*
		3	Functional UI Plate	15							0.7					0.8			■	1.5	*1.9*	*-1.1*
			S-value		6	1.2	2.6	0.4	0.5	0.6	0.7	1.8	0.8	1.2	0.5	1.9	0.6	0.5	0.4			

Chunk 1
Chunk 2
Chunk 3

Fig. 7. A re-engineered QDSM for PLC product design.

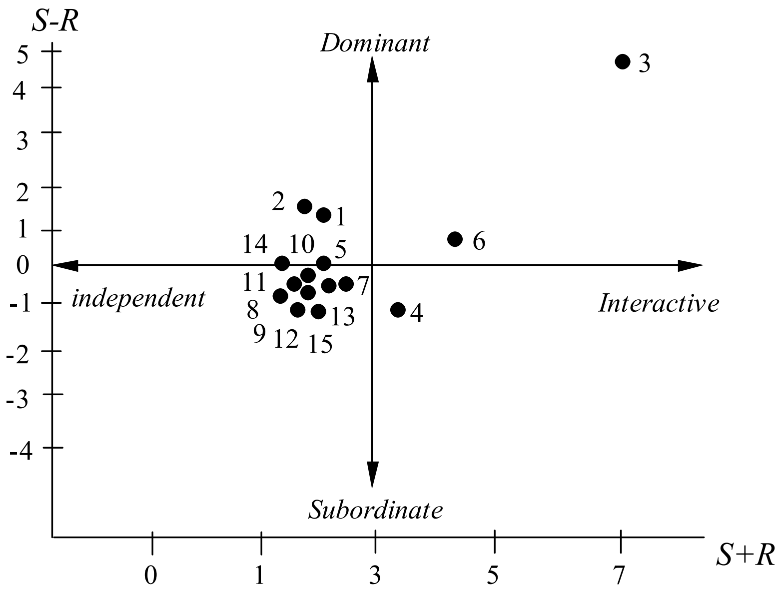


Fig. 8. Plotted diagram of component interaction.

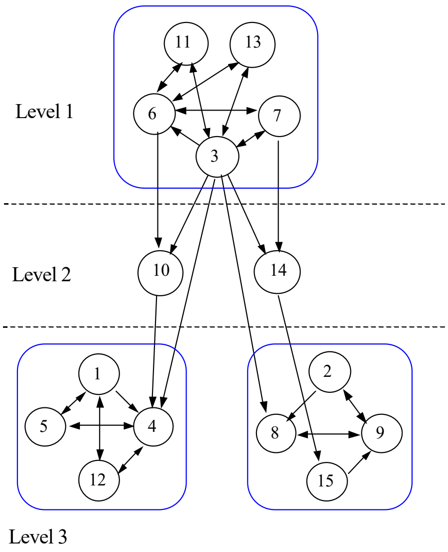


Fig. 9. Hierarchical graph of component interaction.

Figure 9 shows the hierarchical graph of the design constraint flow derived from the re-engineered QDSM. In this graph, the circles represent components, the oriented lines are design constraints provided by the source components, and the rounded rectangles indicate a set of mutually interactive components, which are integrated as a module. These modules and other components then are further grouped into chunks according to the frequency of their interactions.

4.6 Identifying the attributes of product components for design strategies

It is very important to develop a series of products with different depth and width dimensions for design variety. We need to identify the attribute of each component for the variant design and cost down (Halman et al, 2003). For example, we must define which component can be developed to be a platform, a module, or a standardized part for commonality in our product family. In general terms, the goal of the design team is to design the product platform architecture so that as much of the design as possible is standardized across generations and across the product family (Jose and Tollenaere, 2004). The design team tries to modularize parts of the design that cannot be standardized. Definitions of these terms are listed below.

(1) *Modularized*: this is a grouping concept for product design. Components are designed as building blocks which can be grouped together to form a variety of products (Salvador et al., 2002**)**. This concept promotes standardization and the re-use of existing modules to develop a product family. There are some interdependent relationships between these modularized components. This implies that these components have strongly connected relationships and they will become a functional modular design. We can identify the modularized parts using global planning analysis.

(2) *Standardized*: it is expected that the components will not change across generations and across the product family. These standardized parts will become commonality parts within the product family. This implies that a product can meet all the market requirements without having to be redesigned (Ulrich and Eppinger 2000). These components have higher independence. We can identify these standardized parts by their position in the independent- dominant or independent- subordinate quadrant in Figure 8.

(3) *Platform*: this is a design architecture concept of compromising interface definitions and key-components. It helps the design team make decisions on how to rearrange the mapping between the physical components and functions, and how to define interfaces. This implies that the platform is the main technological base for deriving different product families (Du et al., 2001). These components have higher dominance. We can identify these platform parts by their position in the interactive-dominant or independent -dominant in Figure 8.

(4) *Variety*: this is the most popular attribute for product components, especially in identifying appearance parts (Dahmus et al., 2000). We can identify these variety parts by position in the independent-subordinate or interactive-subordinate quadrant in Figure 8.

Besides the above the criteria, we must synthetically consider the other factors including appearance parts and structural parts, for identifying the attribute of each component. If a product design has better configurations using modularized, standardized, and platform parts, the development costs including mold costs and parts costs will be reduced. The main cost reduction criterion is to use as many standardized and modularized parts as possible across the product family.

From the above analysis, we only establish the optimal design process for CE and determine the attributes of product components for designing a product family. According to the segmented market requirements and the analysis results of QDSM, we can illustrate the different requirements of components and define the attribute of each component for concurrently developing four variant PLC products. Figure10 shows the individually required components for four variant PLC products in hierarchy graph. Finally, based on design variety and cost reduction criteria, we define all attributes of product components in Figure 11.

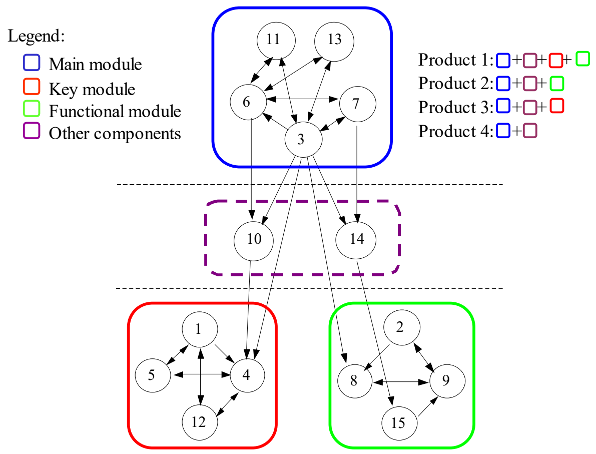


Fig. 10. Individual requirements of components for four variant PLC products.

4.7 Developing a product family

According to the above analysis and design strategies, the designers of company A create four variant products to meet two different market needs and design objective. The product proposals are shown in Figure 12.

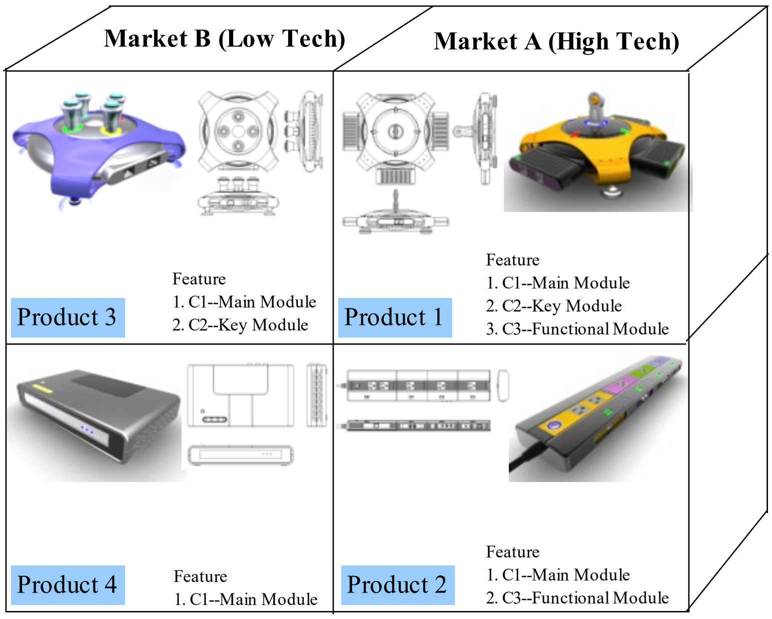


Fig.12. The product proposals for four variant PLC products.

Chunk	Module	Component	No.	Attribute	Product 1	Product 2	Product 3	Product 4
C1	Main Module	Main System PCBA	3	Platformization	V	V	V	V
		Main Base Cover	7	Variety	V	V	V	V
		Main Top Cover	6	Variety	V	V	V	V
		Led Lens	13	Variety	V	V	V	V
		Power Button	11	Variety	V	V	V	V
		Power Plug	10	Standardization	V	V	V	V
		Main IO Plate	14	Standardization	V	V	V	V
C2	Key Module	Key PCBA	1	Platformization	V		V	
		Key Back Cover	5	Variety	V		V	
		Key Front Cover	4	Variety	V		V	
		Key Button	12	Variety	V		V	
C3	Functional Module	Functional PCBA	2	Platformization	V	V		
		Functional Top Cover	8	Variety	V	V		
		Functional Base Cover	9	Variety	V	V		
		Functional UI Plate	15	Standardization	V	V		

Fig. 11. The attribute of each component of the PLC products

5. Conclusions

This research proposed a new system approach for design configurations that considers the optimal design process and product architecture for product variety based on an existing product. QDSM is a compact representation of the information structure of the design process and product architecture. It is a design configuration method that shows the order in which the design components are performed, and what components need to be verified. Our proposal is an enhanced structural model which can be used to visualize the hierarchical structure of product components and optimize the design process for CE. The proposed methodology is divided into two phases: global planning and local planning. The global planning phase focuses on identifying the coupled components sets and rearranges the uncoupled sets using an improved partitioning algorithm. In the local planning phase, a new tearing algorithm is proposed to decouple the coupled components for an optimal design sequence. The procedures of global planning and local planning are presented to re-engineer a design process and product architecture. The proposed approach helps designers and managers optimize the design configurations and plan better design strategies for designing a product family. A case study in PLC product family design was conducted to demonstrate the feasibility and effectiveness of the proposed design configuration approach.

Characteristics of the proposed approach are summarized as follows:
(1) By applying the fuzzy linguistic variables to quantify the degree of dependency between product components, EDG can be carried out and mapped to the proposed QDSM model for further analysis.
(2) By modeling the global planning method, including the reachable matrix, strongly connected matrix, and hierarchical analysis based on the Boolean algebraic operation, the strongly connected components and hierarchical level of product components can be determined. It is a computable method for grouping strongly connected components and a visual hierarchical structure of product components.
(3) By modeling the local planning method, including the calculations of the information input-degrees (IIi), the information output-degrees (IOi), and the ratio $Ri = IIi/IOi$, the optimized design priorities and product architecture for design strategies can also be determined.
(4) By identifying the attributes of product components including modularization, platformization, standardization, and variety based on the analysis results of QDSM, better design strategies for concurrently product family design can be obtained.

Acknowledgements

We would like to thank the R&D division of the ST&T Electronic Company for their assistance in developing the examples presented in this paper. The proposed product 2 in Figure 12 won a Germany 2007 iF Product Design Award (Ko and Li 2007) for the ST&T Electronic Company.

References

Blackenfelt M., Profit maximisation while considering uncertainty by balancing commonality and variety using robust design – the redesign of a family of lift tables, *In: Proc 2000 ASME Design Engineering Technical Conferences*, Baltimore, USA. 2000, ASME, New York.

Claesson A., H. Johannesson , and S. Gedell , Platform product development: product model – a system structure composed of configurable components, *In: Proc 2001 ASME Design Engineering Technical Conferences*, Pittsburgh, USA, 9–12 September. 2001, ASME, New York.

Cui N., R. Chen, and S. Ma, Structured modeling of business process of an enterprise, *Journal of HuaZhong University of Science and Technology*, 1997, 25 (2), 59-62, (in Chinese).

Dahmus J. and Otto K., Incorporating life-cycle costs into product architecture decisions, *In: Proc 2001 ASME Design Engineering Technical Conferences*, Pittsburgh, USA, 9–12 September. 2001, ASME, New York.

Dahmus, J., Gonzalez-Zugasti, P. and Otto, K., Modular product architecture, presented at *2000 ASME Design Engineering Technical Conferences*, DFM-14021, Baltimore, MD, 10–13 September 2000.

Du, X., Jiao, J. and Tseng, M. M., 2001, Architecture of product family: fundamentals and methodology, Concurrent Engineering: Research and Application, 2001, 9, 309–325.

Erens F.J., The synthesis of variety: developing product families, *Dissertation Eindhoven University of Technology*, 1996.

Fujita K. and K. Ishii, Task structuring toward computational approaches to product variety design, *In: Proceedings of the ASME Design Engineering Technical Conferences*, 1997, DETC97/DAC-3766.

Fujita K., H. Sakaguchi, and S. Akagi, Product variety deployment and its optimization under modular architecture and module communalization, *In: Proceedings of the ASME Design Engineering Technical Conferences*, 1999, DETC99/DFM-8923.

Halman, J.I.M., Hofer, A.P. and van Vuuren, W., Platform driven development of product families: linking theory with practice. *J. Product Innovation Management*, 2003, 20(2), 149–162.

Huang G.Q. and K.L. Mak, Re-engineering the product development process with “design for X”, *Proceedings of the Institution of Mechanical Engineers, Part B,* 1998, 212, 259-268.

Jose A. and Tollenaere M., Modular and platform methods for product family design: literature analysis, *Journal of Intelligent Manufacturing*, 2004, 16(3), 371-390.

Kitamura T., A computer-aided approach to the structural analysis and modification of a large circulatory system model, *IEEE Transaction on Biomedical Engineering* , 1999, 46, 485–493.

Ko Y.T. and T.L. Li, *Product design: Multimedia AV Power Cable*, Advanced studies catalogue, iF Yearbook Product 2007, iF international Forum Design GmbH press, Hannover, 2007, Germany: 820.

Kusiak A. and J. Wang, Efficient organizing of design activities, *International Journal of Production Research*, 1993, 31, 753-769.

Maier J. and G. Fadel, Strategic decisions in the early stages of product family design, *In: Proc 2001 ASME Design Engineering Technical Conferences*, Pittsburgh, USA, 9–12 September, 2001, ASME, New York.

Martin M.V. and K. Ishii, Design for variety: development of complexity indices and design chart, *In: Proceedings of the ASME Design Engineering Technical Conference*, 1997, DETC97/DFM-4359.

Pine B.J., *Mass Customization: The New Frontier in Business Competition*, Harvard Business School Press, Boston, 1993.

Roberts F.S., *Discrete Mathematical Models*, Prentice-Hall Englewood Cliffs, NJ, 1976.

Robertson D. and K.T. Ulrich, Planning for product platforms, *Sloan Management Review*, 1998, 39, 19–31.

Rogers J., A knowledge-based tool for multilevel decomposition of complex design problem, NASA TP2903, May, 1989.

Salvador F., Forza C. and Rungtusanatham M., Modularity, product variety, production volume, and component sourcing: theorizing beyond generic prescriptions, 2002, Journal of Operations Management, 20(5), 549-575

Sharma H.D., A.D. Gupta, and Sushil, The objectives of waste management in India: a futures inquiry, *Technological Forecasting and Social Change* , 1995, 48, 285–309.

Smith R.P., *Development and verification of engineering design iteration models*, PhD Thesis, Massachusetts Institute of Technology, 1992.

Steward D.V., The design structure system: a method for managing the design of complex systems, *IEEE Transactions on Engineering Management,*1981, 28(3), 71-74.

Suh N.P., " *The Principles of Design*.", Oxford University Press, NY.

Ulrich, K. and Eppinger, S., *Product Design and Development*, New York: McGraw-Hill, 2000.

Warfield J.N., On arranging elements of a hierarchy in graphic form, *IEEE-SMC3*, 1973, 121–132.

Warfield J.N., A Science of Generic Design: Managing Complexity Through System Design, *Intersystems*, Salinas, CA, 1990, l.1.

Weinblatt H., A new search algorithm for finding the simple cycles of finite directed graphs, *Journal of ACM*, 1972, 19, 43-56.

Xiao R.B., and Q. Fei, Application of the improved method in structural modeling to comprehensive management of the Mine Bereaus, *System Engineering: Theory and Practice*, 1997, 17(3), 57-62. (in Chinese).

Xiao R.B., A new approach to modeling the complex system structure.", *Process in Natural Science* , 2001, 11, 25–32. (in Chinese).

Yassine A. and D. Braha, Complex Concurrent Engineering and the Design Structure Matrix Method, *Concurrent Engineering*, 2003, 11(3), 165-176.

Yassine A. and R.S. Sreenivas, Managing the exchange of information in product development., *European Journal of Operational Research*, 2008, 184(1), 311-326..

AUTHOR BIOGRAPHIES

Yao-Tsung Ko is an Assistant Professor in the Department of Industrial Design, Tunghai University, Taiwan, Republic of China. He received his PhD and Masters degree in Industrial Design from National Cheng Kung University in 2009 and 1995. His research interests include industrial design, fuzzy set theory, computer-aided design, concurrent engineering, design management, innovation design, and new product development.

Manuscript Received: May. 24, 2010
Revision Received: Jun. 20, 2010
and Accepted: Jun. 26, 2010