Logic workflow structure modeling of product variant design and activity path generating

Xinsheng Xu, Tiequn Huang, Cheng Wang, Jun Yuan and Fanfan Zhu
Department of Industrial Engineering, China Jiliang University, Hangzhou, China

Abstract

Purpose – Product variant design process consists of a series of asynchronous activities. These activities and the logic relations among them are important in constructing general logic workflow structure, which is the foundation of deriving an activity path for variant design business. Traditional process modeling approaches have not defined activities for product variant design and cannot describe the complex relations among these activities because of the lack of logic express elements. Thus, logic workflow structure modeling method is anticipated to meet the requirements of logic description and path generation in product variant design application. This paper aims to address these issues.

Design/methodology/approach – The paper identifies the variant design modes of different types of parts and defines their variant design activities. The procedure of constructing general logic workflow structure of product variant design is proposed. Simultaneously, the principles of inferring logic relations among activities are put forward based on their adjacency information and connectivity probability. A general logic workflow structure of product variant design is constructed. Based on this, activity path corresponding to a variant design business can be generated. The algorithm of generating activity path is designed as well. In addition, Boolean vectors of activity path, based on the functional contour matrix of polychromatic set theory, can be inferred, which denotes the functional character of activity path.

Findings – A general logic workflow structure for product variant design has been established, which comprises variant design activities and basic process logic nodes. The logic relations among activities can be inferred based on their in-degree/out-degree and connectivity probability. The function character of activity path can also be expressed based on the polychromatic set theory.

Originality/value – The combination of variant design activity and basic process logic node makes diverse variant design business descriptions possible in a general workflow structure. The proposed approach provides evidences for designer to plan and develop product variant design system effectively.

Keywords Mass customisation, Activity path planning, General logic workflow, Process logic, Variant design activity

Paper type Technical paper

1. Introduction

Mass customization aims at best satisfying individual customer needs with nearly mass production efficiency (Pine, 1993). It is becoming one of popular production modes in the twenty-first century. Variant design, as a valid means to achieve the goal of mass customization, has been well recognized in both academia and industry alike (Tu and Xue, 2008). The strategy of variant design is to derive individualized products based on existing successful designs so as to relieve designers from iterating similar design processes, shorten product development cycle and reduce cost (Wang, 2001). The advantages of variant design in efficient and effective aspects motivated a large body of researches. For example, Wilkes and Leonard (1988) asserted that variant design is a method of automating the mechanical artifacts design process. Fowler (1996) discussed the key approaches used in product variant design with the goal of rapid responding to individual customer requirements. Forster *et al.* (1997) used rule constraints among function parameters to enable intelligent variant design. Germani and Mandorli (2004) investigated the product variant development based on self-configuring component. Wang *et al.* (2005a, 2005b) investigated the assembly variant design based on the dimension constraint relations of assembly at the manufacturing feature level. They formulated assembly variant design as a mixed integer linear programming problem. Khajavirad and Michalek (2008) investigated the platform selection and variant design methods from the perspective of partial component sharing. Sambhoos *et al.* (2009) extracted the assembly mating graphs at the dimension layer for assembly variant design. Feng *et al.* (2010) presented a variant design approach for mechanical parts based on an extensible logic theory. Lo *et al.* (2010) proposed an approach of supporting product variant design through one-step quality function deployment (QFD)-based three-dimensional (3D) morphological charts. Nie *et al.* (2011) researched a rapid

The current issue and full text archive of this journal is available on Emerald Insight at: **www.emeraldinsight.com/0144-5154.htm**

Assembly Automation
36/3 (2016) 333–348

[DOI 10.1108/AA-02-2015-007]

This project was supported by the National Natural Science Foundation of China (NSFC) under contract 51175486 and 51405462 and Zhejiang Province Fund of Application Technology Research for Public, China under contract 2013C31132 and 2014C31117.

Received 4 February 2015
Revised 8 May 2015
5 January 2016
11 April 2016
Accepted 6 May 2016

locking assembly variant design based on product configuration model and case-based reasoning (CBR). Lu and Liu (2011) and Wang *et al.* (2012) presented product variant design methods based on tabular layouts of article characteristics. Yang and Li (2013) proposed a variant design method of series products based on skeleton. Trivedi *et al.* (2013) studied the 3D parametric variant design of the inner ring of spherical roller bearing. Schuh *et al.* (2014) explored the product configuration design based on similarity. Liu *et al.* (2015) proposed a variant design method based on products genes and physical description. Qiao *et al.* (2015) presented a method of generating adaptive assembly from predicting change propagation. In addition, Xu *et al.* (2012) presented an approach of numerical control (NC) programming for mass customization product. It can implement the variant design of NC program based on template. Chen *et al.* (2015) analyzed the change impact in variant product design through an attribute-based and object-oriented approach. Du *et al.* (2015) discussed the joint optimization issues of product variant design from the perspective of product family configuration and scaling design, respectively. However, researches in this field have typically used an artifact perspective (Xu and Jiao, 2009), namely, focusing on the contents of design. The main theme is about decision-making regarding how the details of the product instance are worked out.

Generally, product variant design can be tackled from both artifact and process perspectives. In contrast to the artifact perspective, the process perspective emphasizes on the infra-structure of design activities and their interrelationships. To address these issues, various design process models were proposed such as structured analysis and process description (Xu and Jiao, 2009), meta-model (Wang *et al.*, 2014), hidden Markov-based model (Ning *et al.*, 2014) and agent-based model (Liu *et al.*, 2016). In addition, from the viewpoint of applications, Zhang *et al.* (2007) developed a design process model for the reuse of knowledge. Sause and Powell (2010) proposed a design process model for computer integrated structural engineering. Strömman *et al.* (2011) studied the design process model for the optimizing design of continuous production processes. Tan *et al.* (2011) presented an incremental innovation design process model based on teoriya resheniya izobreatatelskikh zadatch (TRIZ). Liu *et al.* (2012) put forward a design process model which integrates morphological matrix and conflict-resolving principles. Stef *et al.* (2013) investigated the product design process model in the digital factory context. Kuo *et al.* (2016) presented a product attribute-driven eco-design process using depth-first search. However, these models and applications fall short to capture the rich interdependencies, priorities and resource requirements of design activities (Kumar and Ganesh, 1998). Thereupon, Petri net and its extended applications such as timed colored Petri net are investigated on workflow process (Raposo *et al.*, 2000), collaborative product development (Jiang *et al.*, 2008) and design process modeling of product variants (Xu and Jiao, 2009). Unfortunately, these process models and methods still lack activity representation schemes such as information transferring, logic constraint among activities and business paths. A valid method of describing variant design activity and modeling variant design process is not yet developed for complex product.

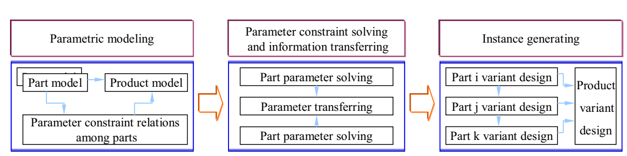
This research is geared toward defining variant design activity of complex product which consists of several types of parts, constructing general logic workflow structure based on variant design activity and basic process logic nodes, deriving activity path of variant design business and describing the function character of activity path. The objective is to provide evidences to designer to insight into the calling logic of variant design activities during product variant design process and to identify the function characters of activity path systematically. In this work, variant design modes of different types of parts are discussed, and the activity elements for complex product variant design process are defined as well. Basic process logic nodes are introduced to express the complex logic relations among activities. Simultaneously, polychromatic sets (Xu *et al.*, 2012a, 2012b, 2012c) are used to describe the multi-attribute characters of basic nodes and variant design activities. On the basis of these, a systematic procedure is proposed to construct a general logic workflow structure of product variant design including logic relation infer among activities and activity path generation.

The remainder of the paper proceeds as follows. The variant design modes for different types of parts and design activity definition are presented in Section 2. Section 3 introduces the basic process logic node and its polychromatic set descriptions. Section 4 presents the methods of constructing the general logic workflow structure of product variant design process. The system framework of product variant design process platform based on the general logic workflow structure is presented in Section 5. The practical implementation of product variant design process platform is investigated in Section 6. Section 7 presents the conclusions about variant design activity and general logic workflow structure for complex product variant design and future researches.

2. Variant design modes and activity definition

Generally, product variant design process can be divided into three stages from the perspective of work contents illustrated in Figure 1, namely, parametric modeling, parameter constraint solving and information transferring and instance generating. Parametric modeling is to build a 3D entity of a part in a parametric computer aided design (CAD) system and establish parameter constraint relations among parts. Then, all these models are assembled into a parametric product model. Parameter constraint solving and information transferring stage is to work out the value of part's parameters by solving parameter constraints according to the design specifications of customized product from customers. Then, these known parameters of parts will be transferred to other parts mating with it directly based on the parameter constraint relations among them (Wang, 2001). As a result, parameters

Figure 1 Overview of product variant design process



of all parts can be determined. Instance generating is to drive the parametric model to generate a new similar part instance. Eventually, product variant design will be achieved after all parts have implemented variant design. These three stages are tightly related through parameters and their information transferring.

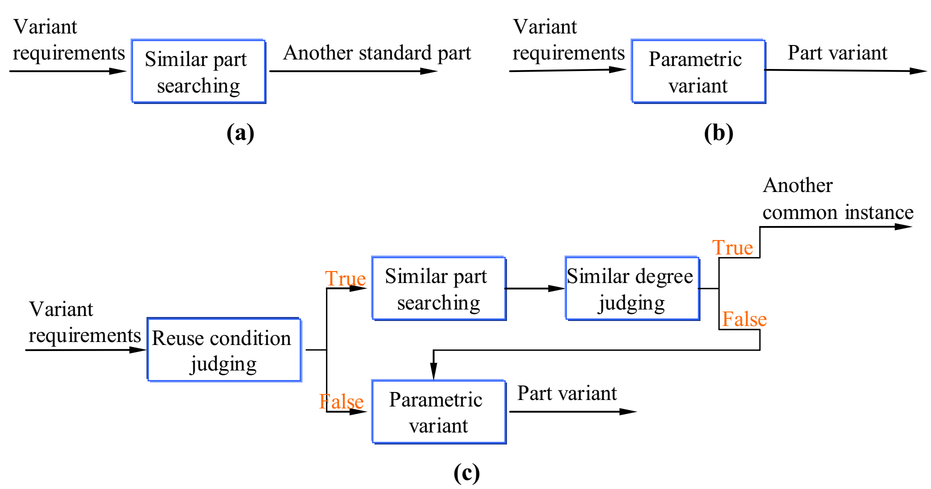
The parametric modeling stage needs lots of human–computer interactions. Usually, it does not have unified activities and workflows because different designers may have different work styles or different work schemes. Therefore, this paper mainly focuses on the later two stages, namely, parameter constraint solving and information transferring and instance generating. To perform our research works, we assume that the parametric models of parts and their parameter constraint network (Xu *et al.*, 2011) have been readied already.

2.1 Part variant design modes

Mechanical product is usually a kind of complex combined system which consists of many types of parts. These types of parts are all involved in product variant design because of the strict assembly mating relations among them (Prebil *et al.*, 1995). According to the source of parts and their utilization status in mass customization product, parts in complex product can be mainly divided into three groups such as standard part, common part and customizable part (Xu *et al.*, 2011). Standard parts are the components whose structure, size, drawing and mark have been completely normalized and are produced by professional manufacturers. Therefore, it does not permit to be changed at all these aspects. Common parts are the components that can be used interchangeably in different series of products or in a family of products. On the context of mass customization, common parts are key contradiction mediators between diversified product requirements and time and cost goal of mass customization production. Customizable parts refer to the components that can be subjected to variant design to meet the individual requirements of customers effectively. Obviously, it is necessary to execute different variant design modes for these three types of parts so as to realize the cost and time goal of mass customization production. For example, the variant design of a standard part can only be achieved by selecting another existing similar standard part to replace the part whose design specifications (namely, parameters) are deduced from the parameter constraint network among mating parts. Otherwise, the design changes of the standard part will lead to a series of production changes and increase the cost and time of customized product drastically in the end. Figure 2 presents the variant design modes of these three types of parts.

It can be seen from Figure 2 that the variant design result of a standard part is still a kind of a standard part. Selecting another existing similar standard part as the variant design result of the current standard part is the fundamental strategy because standard part does not permit to be changed. Otherwise, it will increase the cost of customized product drastically. Therefore, the key activity in a standard part variant design is similar part searching. Common parts are always used in a lots of product instances in a product family. When we attempt to reuse an existing instance of common part to other customized products, we should make sure whether this instance can satisfy with the reuse conditions such as function, structure, etc. If it meets, then we shall select the most suitable instance to reuse in the new customized product. This will reduce a large amount of design and manufacturing cost for the manufacturer. However, customer's requirements sometimes are unique and unpredictable. In many cases, it can not satisfy with the requirements of customer well only by reusing the existing instances of the common part. Thereupon, we would have to modify the common part to satisfy with unique customer's requirements. But, these modifications should be limited with the consideration of cost and time. Therefore, a common part is always designed as a parametric model. When needed, parametric variant design will be implemented to meet the individual requirements of customers. As a result, a common part has two alternative variant design modes. A customizable part is a kind of differentiation enabler that can generate variant instance based on parametric model conveniently, and individual customer's requirements are satisfied well in the end.

Figure 2 Variant design modes of three types of parts



Notes: (a) Standard part; (b) customizable part; (c) common part

On the context of mass customization, classifying part and adopting suitable variant design mode for each class of parts are highly significant. It is the critical foundation to satisfy with the individual needs of customers while achieving the cost and time goal of mass customization production.

2.2 Variant design activity definition

Product variant design process is a series of asynchronous activities in essence (Xu and Jiao, 2009). These activities and their workflows are the foundation of building the product variant design process model. Therefore, the primary task in product variant design process research is to define variant design activity reasonably.

Considering the function characters of activities and the business requirements of product variant design, we, according to the variant design modes in Section 2.1, define variant design activity as a three-tuple structure $E = < E^F, E^D, E^R >$ in which $E^F = \{E_r^F | r = 1, 2, \cdots, n_r\}$ denotes the functional operations of variant design activity, $E^D = \{E_t^D | t = 1, 2, \cdots, n_t\}$ denotes the decision-making operations of variant design activity and $E^R = \{E_u^R | u = 1, 2, \cdots, n_u\}$ denotes the reusable resources for product variant design in mass customization enterprise. The elements in each kind of activities and their relationships are presented in Figure 3. Furthermore, each element is described in detail in Figure 4.

2.2.1 Functional operations

Functional operations consist of units which execute some tasks including $E_{\mathrm{Tr}}^{\mathrm{F}}(\bullet)$, $E_{\mathrm{s}}^{\mathrm{F}}(\bullet)$, $E_{\mathrm{Sim}}^{\mathrm{F}}(\bullet)$ and $E_{\mathrm{V}}^{\mathrm{F}}(\bullet)$. $E_{\mathrm{Tr}}^{\mathrm{F}}(\bullet)$ transfers variant dimensions to other dimensions in the constraint network. Therefore, the parameters' value of variant part can be determined. $E_{\mathrm{s}}^{\mathrm{F}}(\bullet)$ is a kind of searching

Figure 3 Elements of variant design activity and their relationships

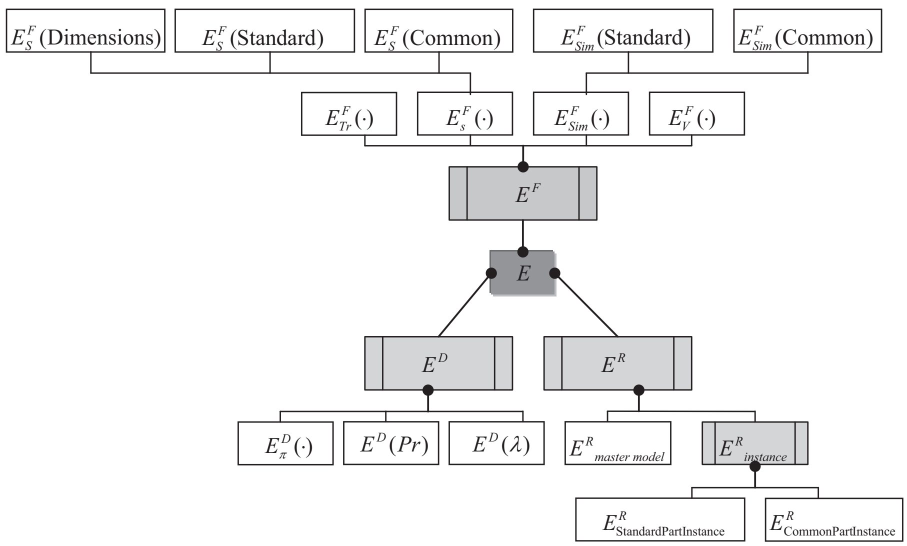


Figure 4 Element descriptions of variant design activities

E^F	Description		Formulations
$E_{Tr}^F(\cdot)$	Variant dimensions transferring (Liu *et al.*, 2007)		$R=\sum_{q=1}^{n_q}(\sum_{u=1}^{n_u}\prod_{v=1}^{n_u} c_{quv,qu(v+1)})d_q$
$E_s^F(\cdot)$	E_s^F(Dimensions)	Searching variable in-degree dimension (Xu *et al.*, 2012)	——
	E_s^F(Standard)	Searching standard part instance (Xu *et al.*, 2011)	——
	E_s^F(Common)	Searching common part instance (Xu *et al.*, 2011)	——
$E_{Sim}^F(\cdot)$	E_{Sim}^F(Standard)	Similarity calculating of standard part (Xu *et al.*, 2011)	$S_{\mathrm{variant}}=\max_{i=1}^{n}(Similarity(s,s_i))$
	E_{Sim}^F(Common)	Similarity calculating of common part (Xu *et al.*, 2011)	$C_{\mathrm{variant}}=\max_{j=1}^{m}(Similarity(c,c_j))\quad Similarity\geq\xi_0$
$E_V^F(\cdot)$	Part parametric variant design		——

(a)

E^D	Description
$E^D(Pr)$	Part type judging
$E^D(\lambda)$	Variable in-degree dimension judging of all dimensions in a part (Xu *et al.*, 2012)
$E_\pi^D(\cdot)$	Part similarity judging (Xu *et al.*, 2011)

(b)

E^R		Description
$E^R_{master\ model}$		Reusing parametric master model of part
$E^R_{instance}$	$E^R_{\mathrm{StandardPartInstance}}$	Reusing standard part instance
	$E^R_{\mathrm{CommonPartInstance}}$	Reusing common part instance

(c)

Assembly Automation
Volume 36 · Number 3 · 2016 · 333–348

operation which contains E^{F}_{S}(Dimensions), E^{F}_{S}(Standard) and E^{F}_{S}(Common). E^{F}_{S}(Dimensions) is to search potential transformable in-degree parameters (Xu *et al.*, 2011) to provide evidences for the variant implementation strategy. E^{F}_{S}(Standard) is to search another existing standard part which is similar to the requirement results of the standard part from dimension transferring in the constraint network and E^{F}_{S} (Common) is to search the existing instance of the common part. $E^{F}_{Sim}(\bullet)$ calculates the similarity between variant part and existing part instance. According to the different requirements of similarity between standard part and common part, E^{F}_{Sim} $(\bullet)$ is also further divided into E^{F}_{Sim}(Standard) and E^{F}_{Sim}(Common). Usually, the similarity between two standard parts is more strict than that of common parts. $E^{F}_{V}(\bullet)$ implements part variant design based on parametric technology.

2.2.2 Decision-making operations

Decision-making operations consist of units that judge which conditions are satisfied with or which condition it belongs to. It includes $E^{D}(Pr)$, $E^{D}(\lambda)$ and $E^{D}_{\pi}(\bullet)$ in which $E^{D}(Pr)$ is to identify part type to select corresponding variant design modes, $E^{D}(\lambda)$ judges whether the potential transformable in-degree parameters of part satisfy with the predefined threshold in practices and $E^{D}_{\pi}(\bullet)$ judges the similarity between variant part and existing part instance, and it should satisfy with the predefined threshold also.

2.2.3 Reusable resources

Reusable resources are the objects which can be used in variant product instances. There are mainly two kinds of part resources, $E^{R}_{mastermodel}$ and $E^{R}_{instance}$. $E^{R}_{mastermodel}$ is a kind of parametric model that can be used to implement parametric variant design to meet the individual requirements from customers, whereas $E^{R}_{instance}$ is the existing part instances that can be reused directly to increase the resource reuse level in the customized product. Furthermore, resource $E^{R}_{instance}$ is divided into $E^{R}_{StandardPartInstance}$ and $E^{R}_{CommonPartInstance}$.

2.3 Integrated variant design process model

Based on the definition of variant design activity in Section 2.2 and the variant modes of different types of parts in Section 2.1, an integrated variant design process for complex product can be established. The primary framework is an integrated variant design process model for parts shown in Figure 5. By executing this process repeatedly, the variant design of complex product which consists of different types of parts can be achieved. It is an intricate structure because of hybrid variant modes and complex connections among variant design activities.

In this process model, rectangular nodes denote the activities defined in Section 2.2. The start node only has output arc but no input arc, whereas the end node only has input arc but no output arc. These characteristics can be used to judge whether a node is a start node or an end node. In addition, the in-degree and (or) out-degree of nodes are different. This means that there are different constraint structures among these activities. These constraint structures describe the mechanisms of triggering next activity. However, even the same constraint structure may have different trigger mechanisms. For example, different output of an activity will trigger a different next activity in a condition judgment. The activity trigger mechanisms in integrated product variant design process and their interpretations are presented in Table I.

Figure 5 Integrated variant design process model

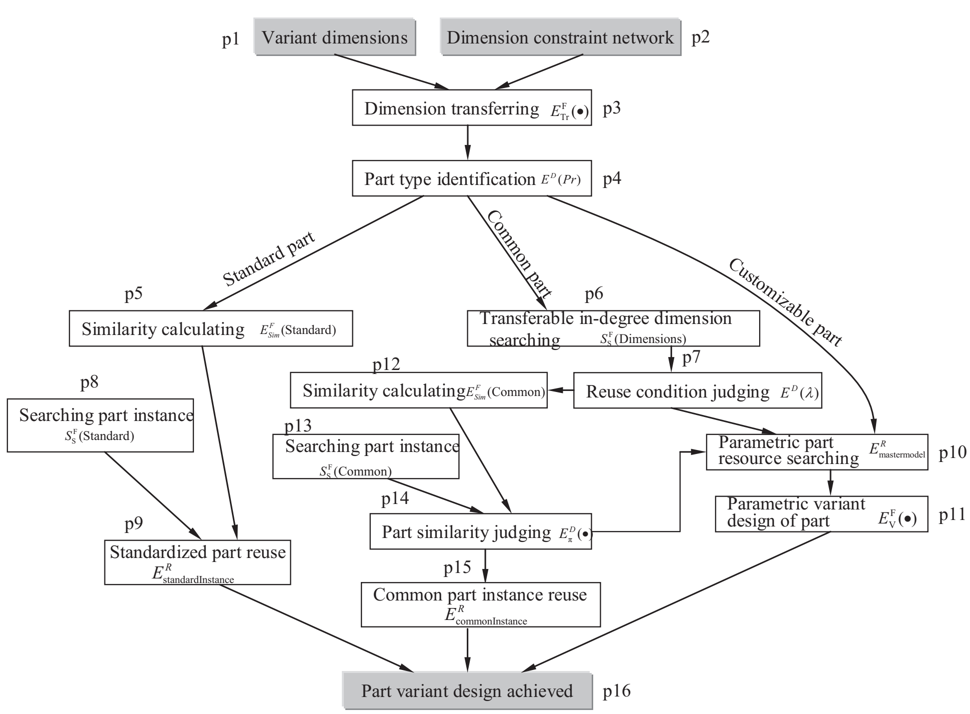
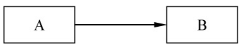
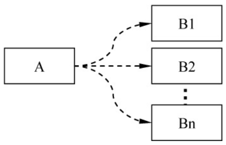
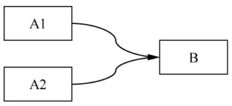
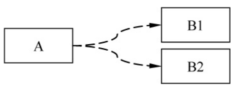
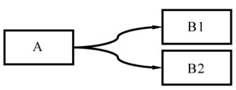
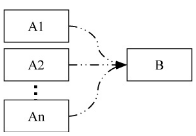


Table I Activity trigger mechanisms and their interpretations

Activity structures	Logic description
	If A is performed, then executing B
	If A is performed, then executing one of (B1, B2, [. . .] Bn) (n ≥ 3) according to the output result of Activity A
	If only A1 and A2 are performed, then activity B will be executed. At the same time, the output results of A1 and A2 are input into Activity B simultaneously
	If A is performed, then executing one of (B1, B2) according to the output result of Activity A
	If A is performed, then B1 and B2 are executed simultaneously
	If one of (A1, A2, . . . An) is satisfied, then executing B

All these trigger mechanisms may be combined or nested in an integrated product variant design process. As a result, it forms intricate logic relations among variant design activities. Obviously, traditional product variant design process model can not express these complex logic characters among variant design activities clearly. Therefore, it is necessary to analyze the constraint relations among variant design activities and to define appropriate logic nodes to describe the trigger mechanisms among variant design activities. The next sections will focus on these issues to present an effective method about activity relation description.

3. Basic process logic nodes of workflow and its polychromatic set descriptions

The basic process logic nodes of workflow which are not the activities needed to be executed are set for describing the logic relations among activities (Xu *et al.*, 2007). Generally, the basic logic nodes for process modeling include condition, selection, synchronization, aggregation, exclusive OR (XOR) branch and XOR join (Zhao and Li, 2008).

Each kind of basic logic node has its predecessor node and subsequent node, and they can simultaneously connect with different variant design activities. Therefore, the in-degree and (or) out-degree of each kind of logic nodes are different, so does the connectivity probability among them. It is a huge challenge to describe these complex characters of basic logic nodes by using traditional theories and methods. Polychromatic set is a newly established system theory (Xu, 2000a; Xu, 2000b; Chaudhry *et al.*, 2000). Its key idea is to use a standard mathematical model to simulate different objects. Because of the availability of the standard mathematical model, the polychromatic theory has made significant progress in problem formalization. The method has a significant advantage which has also been considered as a contribution to theoretical development in systems theory. The theory, techniques and approaches of polychromatic set can play an important role in product life cycle simulation, product conceptual design, concurrent engineering and virtual manufacturing for product modeling, process modeling and process optimization (Li and Xu, 2003).

According to the polychromatic set theory, the polychromatic set descriptions of all basic logic nodes are presented in Table II (Zhu *et al.*, 2006).

When analyzing a product variant design process, this matrix can be used as the evidences to judge the logic relations among activities. Even more important, this kind of quantitative evidence can help us to differentiate these logic relations accurately.

4. General logic workflow structure of product variant design process

Generally, a product variant design business involves several activities to achieve a design task. However, the cross calling among variant design activities make the logic relationships

Table II Relation matrix between basic logic node and its polychromatic set characters

Basic logic nodes	F_1^1	F_2^1	F_3^1	F_4^1	F_5^1	F_6^1	F_7^1	F_8^1	F_9^1
A_1	●		●		●				●
A_2	●			●	●	●		●	
A_3		●	●		●	●		●	
A_4	●			●			●		●
A_5		●	●				●		●
A_6	●			●	●			●	
A_7		●	●		●			●	

Notes: Here, $A_1 \sim A_7$ is sequence node, condition node, selection node, synchronization node, aggregation node, XOR branch node and XOR join node, respectively; F_1^1 and F_2^1 denote that the in-degree of node is equal to 1 or greater than 1; F_3^1 and F_4^1 denote that the out-degree of node is equal to 1 or greater than 1; F_5^1, F_6^1 and F_7^1 denote that the number of node which can be connected synchronously is equal to 1 or greater than 1, otherwise, it is equal to the total number of nodes, which can be connected synchronously. F_8^1 and F_9^1 denote that the probability of connecting with its subsequent node or predecessor node is between 0 and 1 or is equal to 1; and ● is the concerned node very complex. With the help of basic process logic nodes, the logic constraint relationships among variant design activities can be expressed clearly. Therefore, constructing a workflow structure among variant design activities which integrates with basic logic nodes is highly significant for researching product variant design process.

4.1 Procedure of constructing a general logic workflow structure

Based on the polychromatic set characters of basic process logic nodes presented in Table II, a procedure of constructing a general logic workflow structure among variant design activities, integrated with the status information of variant design activity in an integrated process model, is as follows:

- *Step 1*: Constructing adjacency matrix among variant design activities based on the integrated process model of product variant design. This adjacency matrix describes that how many child activities a parent activity has or how many parent activities a child activity has.
- *Step 2*: Calculating the out-degree and in-degree of each activity node based on adjacency matrix and analyzing the connectivity probability among activity nodes.
- *Step 3*: Inferring logic relation among variant design activities by referring to Table II and based on the out-degree and in-degree data of each activity node, as well as its connectivity probability.
- *Step 4*: Inserting logic nodes into the integrated product variant design process model to construct the general logic workflow structure of it in the end.

4.2 Adjacency matrix of variant design activities

The calling relations among activities and their executing sequences in the integrated process model indicate that activities are correlated. The adjacency matrix A of variant design activities is defined as following:

$$A = [a_{ij}]_{n\times n}$$

$$a_{ij} = \begin{cases} 1 & \text{activit y i and activity j connected} \\ 0 & \text{activity i and activity j don't connected} \end{cases}$$

Where n is the number of activities. The element a_{ij} in $A = 0$ or 1. The row in adjacency matrix denotes activities, namely, the elements in the first row of adjacency matrix, and $a_{ij} = 1$ (i denotes row and j denotes column) denotes that activity a_i points to activity a_j in the integrated process model of product variant design. The number of element 1 in each row denotes the out-degree of current activity. Similarly, the column in adjacency matrix denotes activities, namely, the elements in the first column of adjacency matrix. The number of element 1 in each column denotes the in-degree of current activity. Based on these definitions, the adjacency matrix corresponding to the integrated process model of product variant design in Figure 5 is presented in Table III.

4.3 Status characters of activity node and its connectivity probabilistic matrix

Based on adjacency matrix, the out-degree of each activity and its in-degree can be calculated as the following:

$$Activity_i^{\text{out}-\text{degree}} = \sum_{j=1}^{n} a_{ij}$$

$$Activity_i^{in-\text{degree}} = \sum_{i=1}^{n} a_{ij}$$

They are the last column and the last row in Table III. Simultaneously, the connectivity probabilistics among activities are identified based on the assumption that an activity will connect with its next activities with equity possibility if it has more than one executable next activity. For example, activity part type identification and the connectivity probability between two sequence activities is equal to 1. In addition, if current activity is performed, then more than one

Table III Adjacency matrix of variant design activities

Variant design activities	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	$\sum_{j=1}^{n} a_{ij}$
P_1			1														1
P_2			1														1
P_3				1													1
P_4					1	1				1							3
P_5									1								1
P_6							1										1
P_7										1		1					2
P_8									1								1
P_9																1	1
P_{10}											1						1
P_{11}																1	1
P_{12}														1			1
P_{13}														1			1
P_{14}										1					1		2
P_{15}																1	1
P_{16}																	0
$\sum_{i=1}^{n} a_{ij}$	0	0	2	1	1	1	1	0	2	3	1	1	0	2	1	3	

activity will be called simultaneously. The connectivity probability between current activity and each next activity is also equal to 1. If the value is more than 1, previous activity should be called simultaneously to perform current activity. The connectivity probability between current activity and each previous activity is also equal to 1. As a result, the connectivity probability matrix among variant design activities can be constructed as shown in Table IV. Based on these, the polychromatic set characters of variant design activities are also be described in Table V with the definitions of unified color in Table II.

It can be seen from Table III that there are activities whose out-degree is greater than 1, for example, Activity P_4. This means that there exists a selecting operation based on condition judgment in Activity P_4 with the consideration of the connectivity probability among them presented in Table IV. In addition, the out-degree of Activity P_{14} and its in-degree are all greater than 1. This does not conform to the polychromatic set characters of any type of basic process logic nodes. It means that there exist more complex logic relations between Activity P_{14} and the activities around it. All these provide evidences to infer the logic relations among activities reasonably.

4.4 Inferring logic relations among activities

Basic process logic nodes are very important elements in describing the constraint relations among activities. It expresses which activities will be called and their execution sequences. This section, integrated with the definition of basic process logic nodes and their polychromatic set characters, introduces the key principles of inferring the logic relations among activities. Without loss of generality, we take two activities such as Node i and Node $i + 1$ arbitrarily as analysis objects. The inferring processes are presented in Figure 6. Start nodes such as p_1 and p_2 are not included in this inferring process, and Activity p_8 and

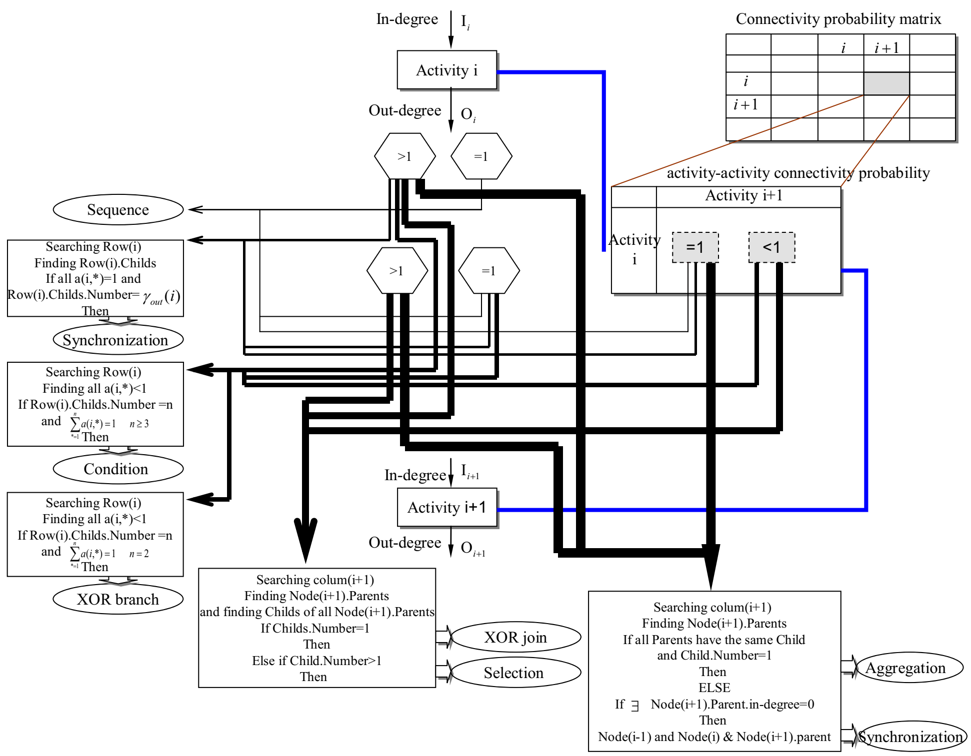
Table IV Connectivity probability matrix among activities

Variant design activities	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}
P_1			1													
P_2			1													
P_3				1												
P_4					1/3	1/3				1/3						
P_5									1							
P_6							1									
P_7										1/2		1/2				
P_8									1							
P_9																1
P_{10}											1					
P_{11}																1
P_{12}														1		
P_{13}														1		
P_{14}										1/2					1/2	
P_{15}																1
P_{16}																

Table V Polychromatic set characters of activities

Variant design activities	F_1^1	F_2^1	F_3^1	F_4^1	F_5^1	F_6^1	F_7^1	F_8^1	F_9^1
P_1			●		●				●
P_2			●		●				●
P_3		●	●		●				●
P_4	●			●		●		●	
P_5	●		●		●				●
P_6	●		●		●				●
P_7	●			●		●		●	
P_8			●		●				●
P_9		●	●		●				●
P_{10}		●	●		●				●
P_{11}	●		●		●				●
P_{12}	●		●		●				●
P_{13}			●		●				●
P_{14}		●		●		●		●	
P_{15}	●		●		●				●
P_{16}		●							●

Figure 6 Principles of inferring logic relations among activities



Activity p_{13} are not start nodes because they are called by the activities which are located at the relative depth position of the activity path. The out-degree of current Activity i and the in-degree of the next Activity $i + 1$, as well as the connectivity probability between them, are important information to identify logic relations. In addition, the pseudo-code about this inferring principle presented in Figure 7 describes the inferring process in detail.

According to pseudo-code descriptions, the basic logic node will be synchronization or aggregation if the next activity can be executed directly without condition judgment in parent–child activity pair. Otherwise, it will be XOR branch (join) or selection. The connectivity probability between synchronization or aggregation node and connected activity is equal to 1, whereas the connectivity probability between XOR branch (join) nodes or selection nodes and its connected activity is in the range from 0 to 1. In addition, the synchronization node or aggregation node can connect with all concerned activities simultaneously, whereas the XOR branch (join) node or selection node can only connect with one or more than one activities (but smaller than the total number of concerned activities) simultaneously. Specially, the activities whose in-degree is 0 and start nodes are absent, such as p_8 and p_{13}, the logic relations between Nodes i and $i-1$ will change because they are always inserted into the activity path which had been constructed as a synchronization relation. Obviously, this increases the difficulty of inferring logic relation. This is the very shortage of traditional process model methods that cannot deal with this issue, whereas the approach in this work can solve it by inferring logic relations and inserting synchronization logic node among them.

4.5 General logic workflow structure of product variant design process

Based on the inferring principle presented in Section 4.4, the logic workflow structure of product variant design process, based on the connectivity probability among activities in Table IV, can be constructed by inserting basic logic node into the integrated product variant design process model (Figure 8).

Based on this general logic workflow structure, an ordered activity set, namely, activity path, corresponding to a variant design business, can be derived from it based on an appropriate algorithm. This general logic workflow structure provides a platform to generate an activity path for a variant design business.

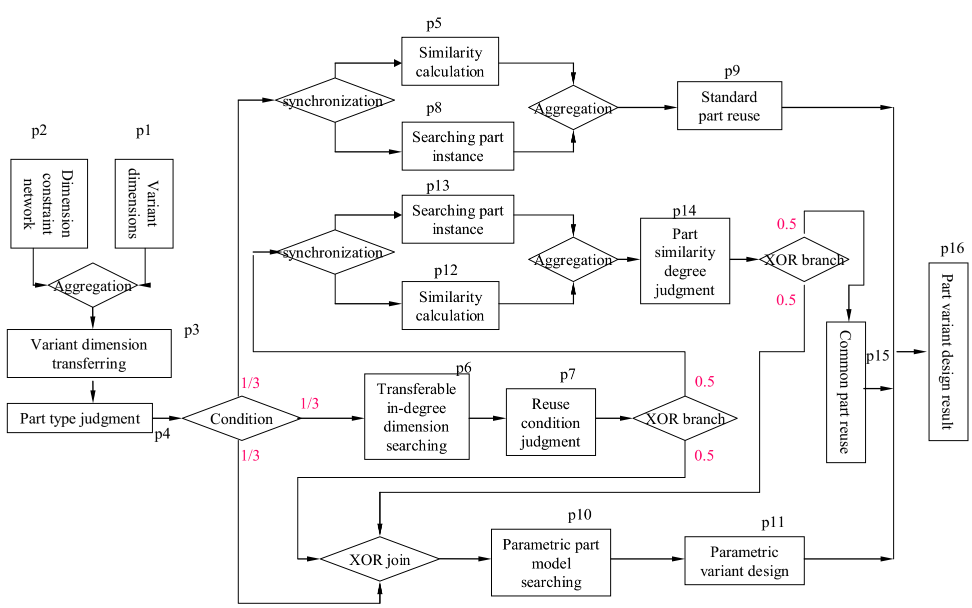
5. System framework based on general logic workflow structure

Based on a general logic workflow structure, a system framework of product variant design process platform based on general logic workflow structure is developed, which consists of three layers, namely, data layer, business layer and application layer, as shown in Figure 9. The first layer stores the model, information and knowledge that support business implementation. The second layer involves the establishment of product variant design process platform that is mainly based on the general logic workflow structure. This platform consists of function modules corresponding to variant design activities and its description based on the polychromatic set. The third layer aims at applying the general logic workflow structure to generate an activity path for variant design business and character its function so as to manage product variant design process effectively.

Figure 7 Pseudo-code descriptions about logic relation inferring

```
Dim ConnectivityProbability between Node i and Node i+1 As a(i,i+1);   Dim ConnectivityProbabilityMatrix As Anxn
Dim In-degree of Node i as γin(i) ; Dim Out-degree of Node i as γout(i) ;   Dim In-degree of Node i+1 as γin(i+1)
Dim Out-degree of Node i+1 as γout(i+1)
LogicRelationInferring
{ IF γout(i)=1 AND γin(i+1)=1 AND a(i,i+1)=1 THEN constructing sequence logic node between Node i and Node i+1
  ELSE
  IF γout(i)>1 AND γin(i+1)=1 AND a(i,i+1)=1 THEN searching all Row(i).Childs
        { IF Row(i).Childs.Numberr > 1 and all a(i, *)=1 THEN
          constructing synchronization logic node between Node i and
          all Row(i).Childs including Node i+1 }
  ELSE
  IF γout(i)>1 AND γin(i+1)=1 AND a(i,i+1)<1 THEN searching all Row(i).Childs
        { IF Row(i).Childs.Number >=3 and all Σ_1^{n(n>=3)} a(i,*)=1 THEN
          constructing condition logic node between Node i and all
          Row(i).Childs including Node i+1
          ELSE
          IF Row(i).Childs.Numberr =2 and all Σ_1^{n(n=2)} a(i,*)=1 THEN
          constructing XOR branch logic node between Node i and all
          Row(i).Childs including Node i+1                              }
  ELSE
  IF γout(i)=1 AND γin(i+1)>1 AND a(i,i+1)<1 THEN searching all Colum(i+1).Parents and calculating  the number of all
        Colum(i+1).Parents.Childs
        { IF Colum(i+1).Parents.Childs.number=1 THEN constructing
          XOR join logic node between Node i+1 and all
          Colum(i+1).Parents including Node i
          ELSE
          IF Colum(i+1).Parents.Childs.number>1 THEN constructing
          constructing Selection logic node between Node i+1 and all
          Colum(i+1).Parents including Node i                           }
  ELSE
  IF γout(i)=1 AND γin(i+1)>1 AND a(i,i+1)=1 THEN searching all Colum(i+1).Parents
        { IF ∃ Colum(i+1).Parents.in-degree=0   THEN constructing synchronization
          logic node between node i-1 and node i & this Colum(i+1).parents
          ELSE
          IF All Colum(i+1).Parents.in-degree ≠  0  THEN constructing aggregation
          logic node between Node i+1 and all Colum(i+1).Parents including Node i}
                                                                                 }
```

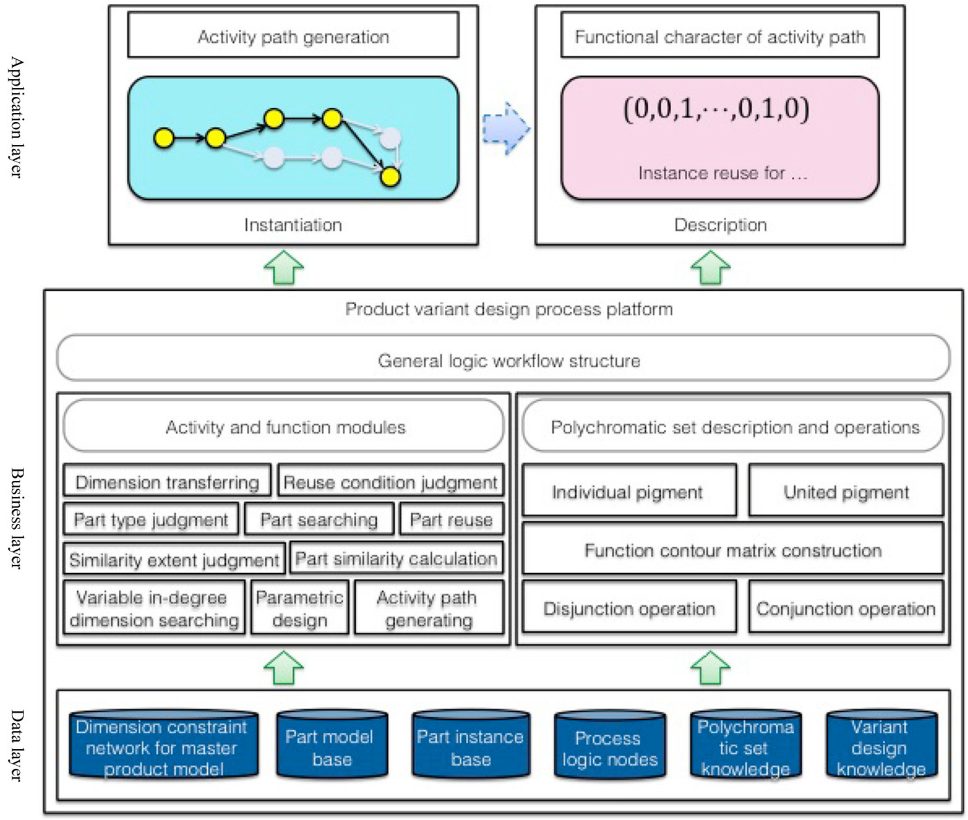
Figure 8 Logic workflow structure of the product variant design process



5.1 Data and model foundation

The data layer includes a dimension constraint network for master product model, part model, part instance, process logic nodes, polychromatic set knowledge and variant design knowledge. The dimension constraint network is the carrier of transferring variant dimension information, and the results from it will be used as evidences to implement part variant design. Part model is a kind of parametric model that will be used to implement variant design if reuse condition cannot be satisfied within a part. Part instance includes standardized part instance and common part instance that are reusable resources. Process logic nodes that are defined in Section 3 are used to describe the calling mechanism among variant design activities. In addition, polychromatic set knowledge base

Figure 9 System framework of a product variant design process platform



contains the descriptions for design activities and process logic nodes. They are used for functional character operation for activity path and logic relation inferring among activities. Variant design knowledge base stores all the thresholds and conditions which are used in generating activity path. All these data and model provide the resources to implement product variant design.

5.2 Businesses in product variant design process platform

This platform contains mainly three modules, activity and function module, polychromatic set operation module and general logic workflow structure. The activity and function module is a set of function programs that correspond to activities defined in Section 2.2 and achieve the goal of activity. The polychromatic set module performs activity description and function description based on the polychromatic theory such as individual pigment and united pigment, as well as the function contour matrix that describes the relation description among them. At the same time, the operations in the polychromatic set such as disjunction and conjunction are designed as function programs to execute logic operation for an activity path. As a result, the function character of an activity path can be inferred. This is highly significant for designer to manage product variant design process from the perspective of function. The general logic workflow structure integrates variant design activities with process logic nodes according to the business workflow and the calling mechanism among activities. This logic structure and these business units form a process platform of supporting product variant design.

5.3 Applications of product variant design process platform

The general logic workflow structure approach is mainly applied in planning a variant design activity path for a family of product which consist of different types of parts and in identifying the function characters of these activity paths based on the polychromatic set. These will provide evidences for designers to plan the business and modules of a product variant design system on the context of mass customization.

6. Practical implementation study

The general logic workflow structure for complex product variant design provides the foundation of generating a variant design activity path. The product variant design process platform is the specific implementation of this idea. The practical implementation contains mainly two aspects, generating activity path and describing its function character. Based on the general logic workflow structure which is constructed in the front of this paper, these two applications are achieved by designing the algorithm of activity path generation and introducing the polychromatic set theory, as well as its operations. The results from these two applications are presented in detail in the following sections.

6.1 Algorithm of generating activity paths

The general logic workflow structure of product variant design, based on the definitions of variant design activities and basic process logic nodes, establishes a master process of variant design for complex product. However, a specific variant design business is always implemented through a serial of activities. Therefore, identifying this ordered activity set, namely, activity path, from the general logic workflow structure is highly significant for identifying the variant design businesses of complex product because the ordered activity sets (namely, activity paths) are the foundations of constructing a function module of product variant design system. In each module, the calling logics among activities are all fixed. Therefore, it is convenient to reuse these modules at other platforms or product variant design systems. The algorithm of generating the activity paths of variant design business is designed in Figure 10.

Based on this algorithm, five activity paths which achieve specific variant design business are generated from the general logic workflow structure in Figure 8. Each activity path is a kind of complete function module for product variant design presented in Table VI. With the help of the control mechanism of basic logic node, some invalid or incomplete activity paths can be avoided. For example, activity path (P_3, P_4, P_6, P_7, P_{12}, P_{14} and P_{15}) in an integrated product variant design process is an invalid path because similarity calculation cannot be performed without part instance. However, the similarity calculation activity integrates with the searching common part instance activity by using the synchronization logic node, and then the similarity calculation can be conducted. As a result, a complete and valid activity path (P_3, P_4, P_6, P_7, P_{12}, P_{13}, P_{14} and P_{15}) will be obtained. This is the very efficacy of basic process logic node in process modeling at the activity level.

6.2 Functional description of activity path

Activity path which consists of several ordered variant design activities can achieve a specific variant design business. Thereupon, identifying the functional character of each activity path normatively is highly significant for designer to get an insight into the task which can be achieved by an activity path. Based on polychromatic set theory, by constructing the functional contour matrix (Xu, 2000a; 2000b; Chaudhry *et al.*, 2000) of variant design activities, each activity can be described as a Boolean vector. It denotes which function can be achieved by this activity. This is explained by the Element 1 in Boolean vector.

According to the goal of product variant design on the context of mass customization, the united pigments for product variant design system, integrated with the resource

Figure 10 Algorithm of generating an activity path

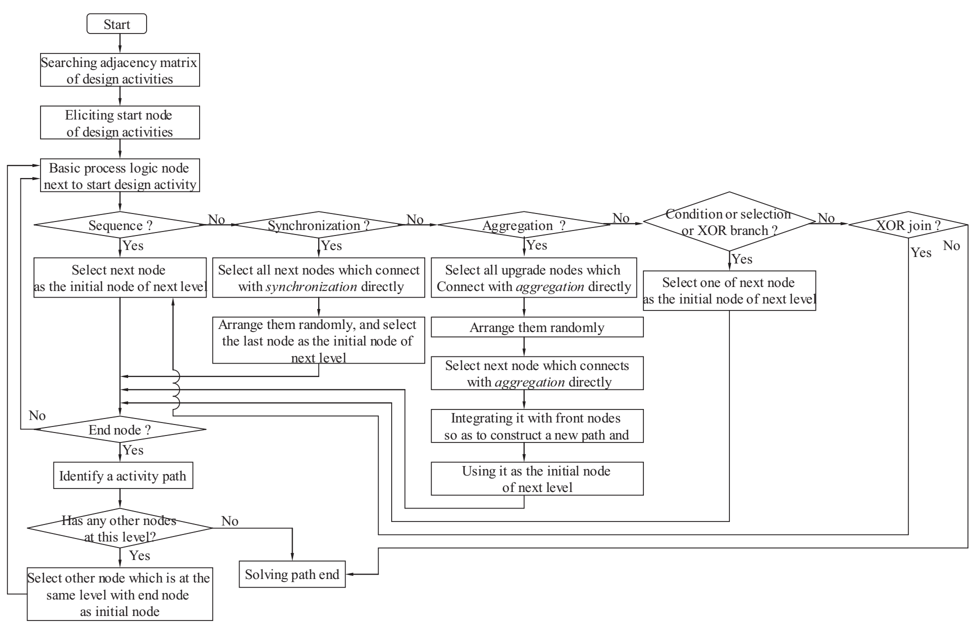


Table VI Activity paths and their functional description

Activity paths and their concerned nodes	Polychromatic set description	Achieving function task
u_1 = (3,4,5,8,9)	$F(u_1)$ = (1,0,0,1,0,0)	Standard part reusing and variant design
u_2 = (3,4,6,7,12,13,14,15)	$F(u_2)$ = (0,1,0,0,1,0)	Common part instance reusing and variant design
u_3 = (3,4,6,7,12,13,14,10,11)	$F(u_3)$ = (0,1,0,0,0,1)	Common part parametric variant design because similarity degree does not satisfy the threshold set in this work
u_4 = (3,4,6,7,10,11)	$F(u_4)$ = (0,1,0,0,0,1)	Common part parametric variant design because reusing conditions are not met
u_5 = (3,4,10,11)	$F(u_5)$ = (0,1,1,0,0,1)	Parametric variant design for customizable part and common part

types of complex product, are defined, which includes six united pigments, F_1, F_2, F_3, F_4, F_5 and F_6. The function contour matrix for product variant design is constructed and presented in Figure 6. On the basis of these, Activity P_3, for example, can be described as vector (1,1,1,1,1,1). It means that Activity P_3, variant dimension transferring, will be used in all function goals of product variant design, and Activity P_5 can be described as vector (1,0,0,1,0,0). It means that Activity similarity calculation (standard) will only be used in the business of standard part variant design. So, other activities can also be described like this. Then, the function character of each activity path can be inferred based on the disjunction and conjunction operations (Li and Xu, 2003) in the polychromatic set theory.

6.2.1 Conjunction of polychromatic set

According to a polychromatic set, if the composition of the entity $A_k(F_j)$ of united color $F_j(A)$ contains more than one element a_{ip}, then:

$$A_k(F_j) = \bigwedge_{p=1}^{m} a_{ip}$$

In this case, the relationship between united color and individual color is called conjunction format, and polychromatic set itself is called polychromatic set in conjunction.

6.2.2 Disjunction of polychromatic set

If the composition of the system entity of all united colors $F_j \in F(A)$ can be represented in terms of the equation:

$$A(F_j) = \bigvee_{p=1}^{m} a_{ip}$$

As a result, activity paths and their elements are presented in Table VI, as well as their function descriptions. The function property of each activity path can be expressed through the number 1 in Boolean vector.

For example, the activity elements of path u_1 include (3,4,5,8,9), and the Boolean vectors of them are P_3 = (1, 1, 1, 1, 1, 1), P_4 = (1, 1, 1, 1, 1, 1), P_5 = (1, 0, 0, 1, 0, 0), P_8 = (1, 0, 0, 1, 0, 0) and P_9 =(1, 0, 0, 1, 0, 0). The function character of $F(u_1)$ can be inferred based on the logic operational rules such as disjunction and conjunction. This calculation process can be descried as the following:

$$F(u_1) = P_3 \wedge P_4 \wedge P_5 \wedge P_8 \wedge P_9 = (1, 0, 0, 1, 0, 0)$$

It can be seen from the Boolean vector (1,0,0,1,0,0) of path u_1 that this path implements the standard part and its instance reusing task. Therefore, this activity path can execute the variant design scheme of the standard part. After these activities such as (3, 4, 5, 8, 9) are called orderly, an appropriate part instance to be reused as a new variant of the standard part will be found out, and the variant design of standard part will be achieved. Similarly, other activity paths and their function descriptions are presented in Table VI. Specifically, the Boolean vector of activity path u_5 is (0,1,1,0,0,1). It means that this activity path can also achieve the parametric variant design of a common part. We think that this exactly exists in practical engineering. For example, a common part can be implemented parametric variant design directly to better meet a customer's requirements promptly. In addition, the common part, through a series of calculation and decision-making, such as similarity calculation activity and similarity degree judgment activity, would still be performed using the parametric variant design, namely, calling functional activity p_{10} and functional activity p_{11}, if all conditions cannot be satisfied. Therefore, the function description result of activity path u_5 can perform the parametric variant design of common part is reasonable. However, this function character of activity path will not be explored or explained (Table VII).

7. Conclusions

Process modeling based on activity presents a new perspective of product variant design research. Instead of focusing on product fulfillment through platform-based techniques, this research analyzes the logic relations among variant design activities so that the calling mechanism among them can be expressed clearly, which is very significant for generating an activity path for a variant design business on the context of mass customization. Such an effort is appropriate because of the activity concurrency and reusability inherent in the product variant design process. Based on the thorough analysis of the design process workflow of different types of parts constituting the complex product, the variant design modes inherent in different types of parts are exploited, and the elements of variant design activities are defined, which are divided into function activity, decision-making activity and resources concerned activity. Furthermore, the trigger structures among these activities are identified. This illustrates the diversified characters of the design activities in the product variant design process, and the product variant design process is complex, especially at the aspect of logic control among variant design activities.

Table VII Function contour matrix [A × F(A)]

Variant design activities	F_1	F_2	F_3	F_4	F_5	F_6
P_3	•	•	•	•	•	•
P_4	•	•	•	•	•	•
P_5	•			•		
P_6		•			•	•
P_7		•			•	•
P_8	•			•		
P_9	•			•		
P_{10}		•	•			•
P_{11}		•	•			•
P_{12}		•			•	•
P_{13}		•			•	•
P_{14}		•			•	•
P_{15}		•			•	

Notes: Here, F_1 is the function contour of the standard part; F_2 is the function contour of the common part; F_3 is the function contour of the customizable part; F_4 is the function contour of the standard part instance reusing; F_5 is the function contour of the common part instance reusing; F_6 is the function contour of the parametric variant design; P_3~P_{15} are the activity nodes in contour; · is the concerned node

To tackle different trigger mechanisms among variant design activities, basic process logic nodes are introduced. Simultaneously, the characters of these logic nodes are presented based on the polychromatic set theory. The variant design activities can be better organized and described based on these basic process logic nodes. Furthermore, the principle of inferring logic relations among variant design activities is proposed based on their in-degree, out-degree and connectivity probability, as well as the polychromatic set descriptions of basic logic nodes. The pseudo-code of this inferring algorithm is also presented. This inferring approach provides a quantity evidence for logic relation analysis. It does not need knowledge or experience of experts. Therefore, it decreases the difficulty of constructing the process variant design process model at the activity level. Then, a general logic workflow structure of product variant design is constructed, which provides a process platform of planning an activity path for variant design business. On the basis of these, the algorithm of generating an activity path is developed for specific variant design business. This general logic workflow structure is advantageous to the traditional integrated process model because it can describe the complex activity trigger mechanism explicitly so as to avoid some invalid or incomplete activity paths. These variant design activity paths are very useful for planning the business units of a product variant design system and further building variant design business module. It may increase the reuse level of variant design activity or variant design business in the end.

On the other hand, variant design activities have diversified functions and attributes. For example, they may be used in different types of parts or variant design businesses. Therefore, it is difficult to describe these functions or attribute characters. This jeopardizes the reuse of variant design activities in product variant design process. To solve such a problem, the polychromatic set theory is introduced to be used in describing these activities and their businesses because of its unified pigments and individual pigments methods. By constructing the functional contour matrix between elements (variant design activity) and unified pigments (business goal), the Boolean vector of each design activity will be obtained. Then, the Boolean vector of activity path corresponding to a variant design business can be inferred based on the conjunction operation and (or) disjunction operation of the polychromatic set. The Element 1 in Boolean vector denotes that the unified pigments will be achieved after these ordered variant design activities are executed. This is highly significant for designer to gain an insight into the function character of variant design activity path and to use it to build sub-modules for product variant design system conveniently.

As might be noticed that the parts constituting complex product are divided into three groups in this work such as standardized part, common part and customizable part. However, in the actual product structure, there may be more types of parts. For example, there are cooperative parts in some products. Therefore, the variant design modes and variant design activities are more complex because they will be determined according to the actual situation of cooperative enterprises. At the same time, the logic relations among activities are also more complex. These are a huge challenge for our methods. Therefore, we need to further improve and explore them in our future work. In addition, the similarity calculation methods among parts in this work and its threshold also affect the result of searching reusable instances which will change activity path and decrease the reuse level of part resources in the end. How to find a robust solution to decrease these influences is the next goal of our research work. Besides, developing product variant design system based on the application results from this work is also an open issue. We will continue to conduct research on it.

References

Chaudhry, S.S., Varano, M. and Xu, L.D. (2000), "Systems research, genetic algorithms and information systems", *Systems Research and Behavioral Science*, Vol. 17 No. 2, pp. 149-162.

Chen, C.Y., Liao, G.Y. and Lin, K.S. (2015), "An attribute-based and object-oriented approach with system implementation for change impact analysis in variant product design", *Computer-Aided Design*, Vol. 62, pp. 203-217.

Du, G., Jiao, R.J. and Chen, M. (2015), "Joint optimization of product family configuration and scaling design by Stackelberg game", *European Journal of Operational Research*, Vol. 232 No. 2, pp. 330-341.

Feng, Y.X., Hao, H., Tan, J.R. and Hagiwara, I. (2010), "Variant design for mechanical parts based on extensible logic theory", *International Journal of Mechanics & Materials in Design*, Vol. 6 No. 2, pp. 123-134.

Forster, J., Fothergill, P. and Arana, I. (1997), *Enabling Intelligent Variant Design Using Constraints*, IEE Colloquium on Intelligent Design Systems, London, pp. 6-8.

Fowler, J.E. (1996), "Variant design for mechanical artifacts: a state-of-the-art survey", *Engineering with Computers*, Vol. 12 No. 1, pp. 1-15.

Germani, M. and Mandorli, F. (2004), "Self-configuring components approach to product variant development", *AI EDAM*, Vol. 18 No. 1, pp. 41-54.

Jiang, P.Y., Shao, X.Y., Qiu, H. and Li, P. (2008), "Interoperability of cross- organizational workflows based on process-view for collaborative product development", *Concurrent Engieerinf Research and Applications*, Vol. 16 No. 1, pp. 73-87.

Khajavirad, A. and Michalek, J.J. (2008), "A decomposed gradient-based approach for generalized platform selection and variant design in product family optimzation", *ASME Journal of Mechanical Design*, Vol. 130 No. 7, pp. 1-8.

Kumar, A.V.K. and Ganesh, L.S. (1998), "Use of petri nets for resource allocation", *IEEE Transaction Engineering Management*, Vol. 45 No. 1, pp. 49-56.

Kuo, T.C., Smith, S., Smith, G.C. and Huang, S.H. (2016), "A predictive product attribute driven eco-design process using depth-first search", *Journal of Cleaner Production*, Vol. 112, pp. 3201-3210.

Li, Z.B. and Xu, L.D. (2003), "Polychromatic sets and its application in simulating complex objects and systems", *Computer & Operations Research*, Vol. 30 No. 6, pp. 851-860.

Liu, R.J., Jiang, D.F. and Shi, L. (2016), "Agent-based simulation of alternative classroom evacuation scenarios", *Frontiers of Architectural Research*, Vol. 5 No. 1, pp. 111-125.

Liu, W., Cao, G.Z., Tan, R.H. and Li, M.L. (2015), "Variant design based on products genes and physical description", *Computer Integrated Manufacturing Systems*, Vol. 21 No. 2, pp. 381-391 (in Chinese).

Liu, X.Z., Gi, G.N., Fu, J.Z., Fan, B.B. and Xu, J. (2012), "A design process model of integrated morphological matrix and conflict resolving principles", *Journal of Zhejiang University: Engineering*, Vol. 46 No. 12, pp. 2243-2251 (in Chinese).

Lo, C.H., Tseng, K.C. and Chu, C.H. (2010), "One-Step QFD based 3D morphological charts for concepts generation of product variant design", *Expert Systems with Applications*, Vol. 37 No. 11, pp. 7351-7363.

Lu, Y.J. and Liu, G.W. (2011), "Research on variant design of parts based on tabular layouts of article characteristics", *Advanced Materials Research*, Vols 181/182, pp. 782-786.

Nie, Y., Yin, G.F., Zhao, X.F., Fang, H. and Yin, Y. (2011), "Rapid locking assembly variant design based on product configuration model and CBR", *Applied Mechanics & Materials*, Vols 48/49, pp. 868-872.

Ning, C., Chen, M. and Zhou, D. (2014), "Hidden Markov model-based statistics pattern analysis for multimode process monitoring: an index-switching scheme", *Industrial & Engineering Chemistry Research*, Vol. 53 No. 27, pp. 11084-11095.

Pine, B.J. (1993), *Mass Customization*, Harvard Business School Press, Boston, MA.

Prebil, I., Zupan, S. and Lučič, P. (1995), "Adaptive and variant design of rotational connections", *Engineering with Computers*, Vol. 11, pp. 83-93.

Qiao, H., Mo, R. and Xiang, Y. (2015), "The generation of adaptive assembly from predicting change propagation", *Assembly Automation*, Vol. 35 No. 3, pp. 269-280.

Raposo, A.B., Magalhaes, L.P. and Ricarte, *et al.* (2000), "Petri nets based coordination mechanisms for multi-workflow environments", *Computer Systems Science & Engineering*, Vol. 15 No. 5, pp. 315-326.

Sambhoos, K., Koc, B. and Nagi, B. (2009), "Extracting assembly mating graphs for assembly variant design", *Journal of Computing and Information Science in Engineering*, Vol. 9 No. 3, pp. 1-9.

Sause, R. and Powell, G.H. (2010), "A design process model for computer integrated structural engineering", *Computing in Civil Engineering & Geographic Information Systems Symposium*, Vol. 6 No. 3, pp. 129-143.

Schuh, G., Rudolf, S. and Riesener, M. (2014), "Similarity-based product configuration", *47th Conference on Manufacturing Systems (CMS 2014)*, *Windsor*, *28-30 April 2014*.

Stef, I.D., Draghici, G. and Draghici, A. (2013), "Product design process model in the digital context", *Procedia Technology*, Vol. 9, pp. 451-462.

Strömman, M., Seilonen, I., Peltola, J. and Koskinen, K. (2011), "Design process model for optimizing design of continuous production processes", available at: www.researchgate.net/publication/221274483_Design_Process_Model_for_Optimizing_Design_of_Continuous_Production_Processes

Tan, R.H., Dong, Y.Q., Zhang, H.G., Yu, F. and Jin, H. (2011), "An incremental innovation design process model based on TRIZ", *Advanced Materials Research*, Vols 418/420, pp. 2174-2181.

Trivedi, R.D., Shah, D.B. and Patel, K.M. (2013), "3D parametric modeling for product variants using case study on inner ring of spherical roller bearing", *Procedia Engineering*, Vol. 51, pp. 709-714.

Tu, Y.L. and Xue, D.Y. (2008), "A framework for generating product production information for mass customization", *International Journal of Advanced Manufacturing Technology*, Vol. 38 No. 11, pp. 1244-1259.

Wang, A.H. (2001), *Variant Design of Complex Assemblies in Agile Manufacturing*, State University of New York at Buffalo, Buffalo.

Wang, A.H., Koc, B. and Nagi, R. (2005a), "Complex assembly variant design in agile manufacturing. Part I: system architecture and assembly modeling methodology", *IIE Transactions on Design and Manufacturing*, Vol. 37 No. 1, pp. 1-5.

Wang, A.H., Koc, B. and Nagi, R. (2005b), "Complex assembly variant design in agile manufacturing. Part II: assembly variant design methodology", *IIE Transactions on Design and Manufacturing*, Vol. 37 No. 1, pp. 17-33.

Wang, K., Chan, T., Kwa, S.T., Ma, Y.F. and Lau, R. (2014), "Meta-modelling for fast analysis of CFD-simulated vapor cloud dispersion processes", *Computer & Chemical Engineering*, Vol. 69, pp. 89-97.

Wang, L., He, Y.R. and Gong, Y.P. (2012), "Clutch product rapid variant design based on tabular layouts of article characteristics", *Applied Mechanics & Materials*, Vols 229/231, pp. 597-601.

Wilkes, J.R. and Leonard, R. (1988), "Variant design as a method of automating the design process", *Computer-Aided Engineering Journal*, Vol. 5 No. 3, pp. 97-102.

Xu, L.D. (2000a), "Systems science systems approach and information systems research", *Systems Research and Behavioral Science*, Vol. 17 No. 2, pp. 103-104.

Xu, L.D. (2000b), "The contribution of systems science to information systems research", *Systems Research and Behavioral Science*, Vol. 17 No. 2, pp. 105-116.

Xu, L.D., Li, Z.B., Li, S.C. and Tang, F. (2007), "A decision support system for product design in concurrent engineering", *Decision Support Systems*, Vol. 42 No. 4, pp. 2029-2042.

Xu, Q.L. and Jiao, J.X. (2009), "Modeling the design process of product variants with timed colored Petri nets", *Journal of Mechanical Design*, Vol. 131 No. 6, pp. 9-18.

Xu, X.S., Li, D. and Sun, M.L. (2011), "Variant design method of assembly consisting of different types of parts", *Advanced Materials Research*, Vols 201/203, pp. 1460-1463.

Xu, X.S., Li, D., Yan, T.H. and Xin, C. (2012a), "Dimension constraint conflicts delayed methods and its applications in variant design", *Computer Integrated Manufacturing Systems*, Vol. 18 No. 2, pp. 254-160 (in Chinese).

Xu, X.S., Yan, T.H., Tao, X.Z., Zhu, T.R. and Li, D. (2012b), "Generating NC program based on template for mass customization product", *Assembly Automation*, Vol. 32 No. 3, pp. 251-261.

Xu, Z.J., Li, Y., Zhang, J., Cheng, H., Jiang, S.S. and Tang, W.B. (2012c), "A dynamic assembly model for assembly sequence planning of complex product based on polychromatic sets theory", *Assembly Automation*, Vol. 32 No. 2, pp. 152-162.

Yang, B.Z. and Li, C.N. (2013), "Variant design method of series products based on skeleton", *Applied Mechanics & Materials*, Vols 300/301, pp. 318-322.

Zhang, P., Baxter, D., Gao, X. and Kalta, M. (2007), *Developing a Design Process Model for the Reuse of Knowledge*, DET2007, Bath, GB, 17-19 September.

Zhao, S.S. and Li, Z.B. (2008), "A new assembly sequence generation of three dimension product based on polychromatic sets", *Information Technology Journal*, Vol. 7 No. 1, pp. 112-118.

Zhu, J., Li, Z.B. and Gao, X.Q. (2006), "Modeling and route acquired algorithm of workflow using polychromatic sets", *Journal of Xi'an Jiaotong University*, Vol. 40 No. 3, pp. 348-352 (in Chinese).

Further reading

Liu, F.Y. and Qi, G.N. (2007), "Dimension parameters transfer method of configuration product and its application", *Chinese Journal of Mechanical Engineering*, Vol. 43 No. 4, pp. 144-151 (in Chinese).

Corresponding author

Xinsheng Xu can be contacted at: lionkingxxs@163.com