Int J Adv Manuf Technol (2007) 33:652–661
DOI 10.1007/s00170-007-1025-9

ORIGINAL ARTICLE

Concurrent design process analysis and optimization for aluminum profile extrusion product development

Hong - Qin Wei

Received: 29 October 2002 / Accepted: 9 March 2003 / Published online: 10 May 2007

Abstract Owing to intricate interdependency and information feedback among tasks, the concurrent design process probably cannot converge to the correct solution when there are wrong integration and improper interaction between activities. Therefore research on concurrent design process optimization is necessary. In the paper, a novel methodology is proposed to analyze and optimize the concurrent engineering process scientifically. Based on design structure matrix and graph theory, coupled task recognition and design task level plotting are performed for the concurrent design process of aluminum profile extrusion product development. Three factors are used to describe the coupling property of activity, namely sensitivity, complexity and affection factors, which provide the basis to analyze development process quantitatively. And an optimism algorithm is presented to define the initial iteration order of coupled task set. Finally a rational and efficient concurrent design process model is constructed, which can make aluminum profile product development faster, with lower costs and higher quality. The methodology proposed is also applicable to other concurrent engineering fields.

Keywords Aluminum profile · Product development · Concurrent design · Process optimization

H. Wei (✉)
Glorious Sun Business and Management School,
Dong Hua University,
1882 West Yan'an Road,
Shanghai 200051, People's Republic of China
e-mail: hqwei@dhu.edu.cn

1 Introduction

Concurrent engineering (CE) has become increasingly important for product development in recent years. CE is a philosophy that suggests the need to consider design issues simultaneously where they were considered sequentially in the past. The sequential design process has been considered inefficient, since this type of design process typically leads to greater development time, greater cost, and lower overall design quality, all of which lower the overall profit generated by the design [1]. But concurrent design is not always prior to sequential method. Owing to intricate information feedback, when there are wrong integration and improper interaction between activities, concurrent design process possibly divergent [2]. Therefore research on concurrent design process is necessary. Process analysis and scheduling of concurrent design aims to replace the confused coupling condition by a few of small design loops with information feedback according to the interdependency degree between activities.

In general, it is considered that managing the design process includes four major steps [3]: (1) model the information and dependency structure of the design process; (2) provide a design plan showing the order of execution for the design tasks; (3) reduce the risk and magnitude of iteration between design tasks; (4) explore opportunities for reducing the project cycle time. Several management tools have been developed to model the interface and dependencies among the decomposed tasks of design process. The project evaluation and review technique (PERT) is a diagraph of a project [4]. In the PERT method, three probabilistic time estimates are given to each task. The critical path method (CPM) is a variation of the PERT method. CPM assumes a time-cost tradeoff rather than probabilistic time used in PERT. Both methods improve the process flow only by crashing the critical activities, but they do not consider iteration and feedback loops that are characteristics of engineering designs, and they ignore the concurrency and overlapping of the design process. To study information management processes the standardized IDEF0 modeling technique is a useful tool. IDEF0 was driven from structural analysis and design technique [5]. The IDEF0 technique supports the needs of modeling the process in a formalized manner to be able to compare and refine the modeled process. However, the IDEF0 methodology is inefficient to support the modeling of concurrent activities and iterations between activities are difficult to analyze with the help of the IDEF0 technique. Direct graph is a general method to describe procedure relations. It is composed with vertexes representing design activities and vectorial lines representing information connection. The direct graph can not only describe system clear and visually, but also analyze the process quantitatively based on mathematic tools of graph theory (GT) [6]. But when concurrent system is complex and involved activity amount is too large, the direct graph will become very confused. A more compact representation of a design process is the design structure matrix (DSM) [7]. It overcomes the size and visual complexity of all graph-based techniques and matrices are amenable to computer manipulation and storage. DSM has been used in some researches for CE implementation [5, 8–10]. In this paper, we also use DSM representation to describe the product design process. On the basis of DSM and GT, a methodology has been proposed to analyze and optimize concurrent product development process.

This paper has been organized in the following manner. In Sect. 2, the interdependencies among activities during design process are analyzed and summarized into three types. An algorithm to recognize the coupled activities of the product development and figure out the order levels of every design activities is introduced. Section 3 proposes three factors to describe the coupling property of activity and a quantitative optimization algorithm is presented for coupled activities to define iteration order. Section 4 analyzes aluminum profile product development process particularly and presents an optimized process model. Finally a summary and proposed future work are given in Sect. 5.

2 Concurrent design process analysis

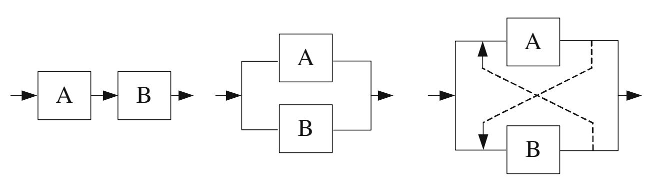
2.1 Relations among design activities and DSM

The product design process is a set of design activities. There are various interrelations among these activities [11, 12]. According to the degree of interdependency, the authors divide the interdependency relationship among activities into three types: sequential relation, parallel relation and coupled relations as shown in Fig. 1.

Sequential relation means that decisions of early design activities affect downstream activities, but no repercussions, generating a sequence of decisions that results in a straightforward process without iteration. Parallel relation means that a design activity is independent from one another or the interdependency degree between them is very low. And there is little information exchange between them. In this situation two or more activities can be executed in parallel. Coupled relationship means that the two activities will affect each other. The interdependency degree between them is very high, and there are information exchanges among coupled activities. Such interdependency needs many iteration loops to set all design information in a consistent way.

By means of DSM (see Fig. 2), an activity set $S=\{a_1, a_2, ..., a_n\}$ can be represented by matrix B with n rows and n columns and, in which $b_{ij}=1(i, j=1, 2,..., n)$ indicates that activity a_i outputs information to a_j while $b_{ij}=0$ indicates that activity a_i doesn't output information to a_j. And B is called DSM of S. It is defined that b_{ii} equals to zero. If one interprets the activity ordering in the DSM as the execution sequence, the elements below the diagonal represent the forward information transfer to later (i.e., downstream) activities, and the elements above the diagonal depict information fed back (or iteration) to earlier (i.e., upstream) activities. As for the three relations, there are the following conclusions:

1. If it is sequential relation between a_i and a_j, then $b_{ij}=1$ and $b_{ji}=0$.
2. If it is parallel relation between a_i and a_j, then $b_{ij}=b_{ji}=0$.
3. If it is coupled relations between a_i and a_j, then $b_{ij}=b_{ji}=1$.



a b c

Fig. 1 Relation types between design activities: (a) sequential relation, (b) parallel relation, (c) coupled relation

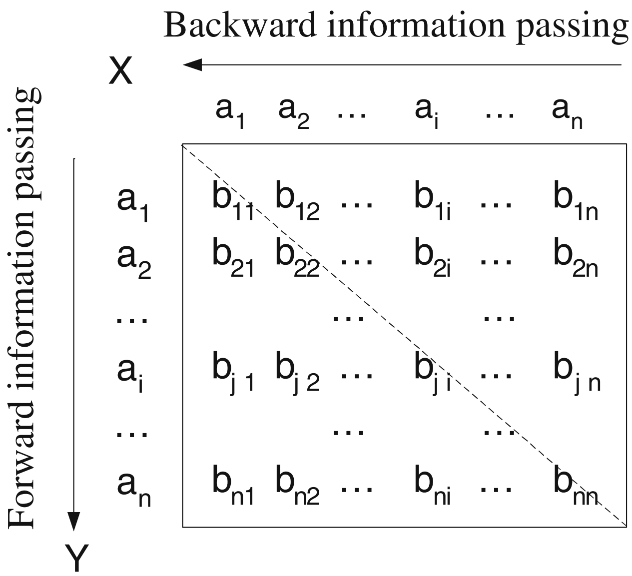


Fig. 2 Design structure matrix

2.2 Coupling activities recognition

In concurrent design, coupled relation is the main relation between activities and the main factor increasing complexity of concurrent design. Frequent information exchanging among activities makes the design and management of process very complicated. Therefore coupled activities should be recognized before re-engineering the design process. Generally designer plans and manages the design tasks by his own domain knowledge and experience qualitatively which is imprecise and unreliable. In this paper, a new algorithm of recognizing coupled activities is introduced as follows.

Defining design activities as nodes of vectorial graph G and activities relation as directed lines, the design process can be described as a directed graph. The directed lines or linkage reflect a dependency or a relationship between the connected activities. And the DSM can be considered as the transpose of the incidence matrix of the corresponding directed graph. The problem of recognizing coupled activities set is translated into the problem of seeking strongly connected component in a directed graph.

Definition 1 Given that G=(V, E) is a directed graph, where V=($\vee_1$, $\vee_2$, ..., $\vee_n$), and A is the incidence matrix of G, P is the accessibility matrix of G Then

$$P = A^{(1)} \vee A^{(1)} \vee A^{(2)} \vee A^{(3)} \vee ... \vee A^{(n)}$$
$$= \bigvee_{j=1}^{n} A^{(i)} (1 \le j \le n, 1 \le i \le n) \quad (1)$$

Table 1 Algorithm of "∨"

Algorithm of "∨"		
∨	0	1
0	0	1
1	1	1

where

$$\underbrace{A^{(i)} = A^{(1)} \wedge A^{(1)} \wedge ... \wedge A^{(1)}}_{i} \quad (2)$$

"∧" is called the Boolean Sum operator, and "∨" is called the Boolean Product operator. And both operations are defined as follows.

Definition 2 (1) The matrix $R = (r_{ij})_{n\times n}$ is a Boolean matrix, if the values of r_{ij}=(i, j=1, 2,..., n) can only be 0 or 1.

(2) The matrix $W = (w_{ij})_{n\times n}$ is a Boolean Sum of R and S, if $w_{ij} = r_{ij} \vee s_{ij}$ where $R = (r_{ij})_{n\times n}$ and $S = (s_{ij})_{n\times n}$ are Boolean matrices.

(3) The matrix $U = (u_{ij})_{n\times n}$ is a Boolean Product of R and S, if $u_{ij} = \vee_{k=1}^{n} (r_{ik} \wedge s_{kj})$, where $R = (r_{ij})_{n\times n}$ and $S = (s_{ij})_{n\times n}$ are Boolean matrices.

(4) The algorithms of "∨" and "∧" are explained in Tables 1 and 2.

Theorem 1 [6] $P = (p_{ij})_{n\times n}$ is accessibility matrix of graph G. P^T is the transposed matrix of P. Operation $P\cap P^T$ is defined as

$$P \cap P^T = \begin{vmatrix} p_{11} & \cdots & p_{1n} \\ \vdots & & \vdots \\ p_{n1} & \cdots & p_{nn} \end{vmatrix} \cap \begin{vmatrix} p_{11} & \cdots & p_{n1} \\ \vdots & & \vdots \\ p_{1n} & \cdots & p_{nn} \end{vmatrix} \quad (3)$$
$$= \begin{vmatrix} p_{11}^2 & p_{12}\cdot p_{21} & \cdots & p_{1n}\cdot p_{n1} \\ p_{21}\cdot p_{12} & p_{22}^2 & \cdots & p_{2n}\cdot p_{n2} \\ \vdots & \vdots & & \vdots \\ p_{n1}\cdot p_{1n} & p_{n2}\cdot p_{2n} & \cdots & p_{nn}^2 \end{vmatrix}$$

Table 2 Algorithm of "∧"

Algorithm of "∧"		
∧	0	1
0	0	0
1	0	1

If it is accessible from node v_i to node v_j, then $p_{ij}=1$. If it is accessible from node v_j to node v_i, then $p_{ji}=1$. Thus the nodes v_i and v_j are accessible from each other, if and only if $p_{ij}\cdot p_{ji}=1$. As to the matrix $P\cap P^T$, If in *i*th row of matrix $P\cap P^T$, elements of column $j_1, j_2, ..., j_k$ is not zero, node v_i, $v_{j1}, v_{j2}, ..., v_{jk}$ is in a strong connected set. And the activities corresponding to these nodes are in a coupled set.

2.3 Design activities level plotting

If every coupled activity set is merged into one activity, and the rows and columns responding to the coupled activity set have been merged into one row and column, the accessibility matrix P becomes a reduced matrix P. Design activities levels can be plotted according to Theorem 2.

Theorem 2 [13] If P′ is the reduced matrix of accessibility matrix P, $P'E_{m-1} = (p_1, p_2,..., p_n)^T$, $m \geq 1$, n-dimension vector $E_0=(1,1,...,1)^T$, $E_m=(e_1, e_2,..., e_n)^T$, where

$$e_i = \left| \begin{array}{l} 0\,(p_i \in \{0,\ 1\}) \\ 1\,(p_i \notin \{0,\ 1\}) \end{array} \right. \tag{4}$$

Then the necessary and sufficient condition of $L_r = \{v_i\}$ is $p_i = 1$, where L_r means that a_i is an element of *m*th level in the Graph G.

3 Concurrent design process optimization

3.1 Design activities reengineering

As the result of design tasks analysis, some deficiencies such as too large coupled set and insufficient concurrent operation can be found. To make whole design process rational, some methods are applied to reengineer tasks. There are mainly two operations proposed, which is shown in Fig. 3.

1. Decoupling operation. If the connection between two interdependent tasks is very weak, the relation can be deleted to make coupling linkage break. In this operation, owing to wrong assumption, it is likely to cause iteration. The weaker the dependency is, the less risk of iteration there will be.
2. Detailing operation. By dividing a task into several smaller activities, upstream activity can provide design information to downstream earlier while downstream activity can also feedback forward timely [14]. Then overall iterative work and cycle time are reduced.

After all activities and relation have been analyzed, coupled activities recognizing and level plotting will be made for the second time. Then iteration order of coupled activities is defined. The complete flowchart is shown in Fig. 4.

3.2 Iteration order optimization for coupled activities

The objective of iteration order optimization is to reduce the iteration work and time caused by traditional empirically based order definition method and make the process converge as soon as possible [15, 16]. In this paper, an optimization algorithm is presented to get scientific initial iteration order. The coupled activities are plotted by the following principles:

1. Activity affected by other activities greatly or sensitive to other tasks should be executed later to make it acquire sufficient information and to reduce loop times;
2. The more complex an activity is, the higher the iteration cost will be. Therefore complex activity should be executed later;
3. Activity affecting others greatly should be placed forward.

In the design process, every activity can be regarded as an "information machining centre", in which the input information is transformed into output information. In this paper, the input and output information of an activity is divided into two groups: local information relating to activities in the same coupled set while global information relating to activities out of the set. As shown in Fig. 5, $\{I_1,I_2, ...,I_n\}$ are the local input information, while $\{I_{n+1}, ...,I_m\}$ global input information, $\{O_1,O_2, ...,O_n\}$ local output information and $\{O_{n+1}, ...,O_m\}$ global output information.

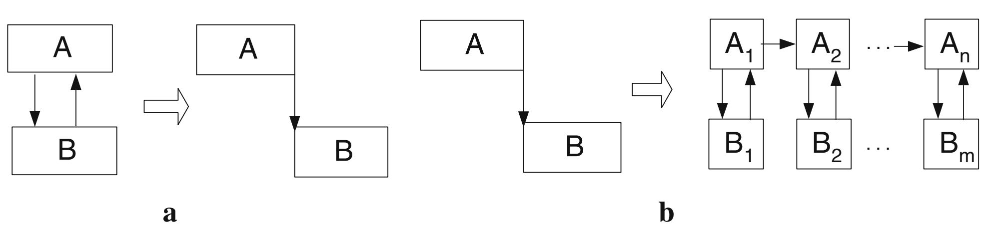


Fig. 3 Design activities reengineering operations: (**a**) decouping, (**b**) detailing

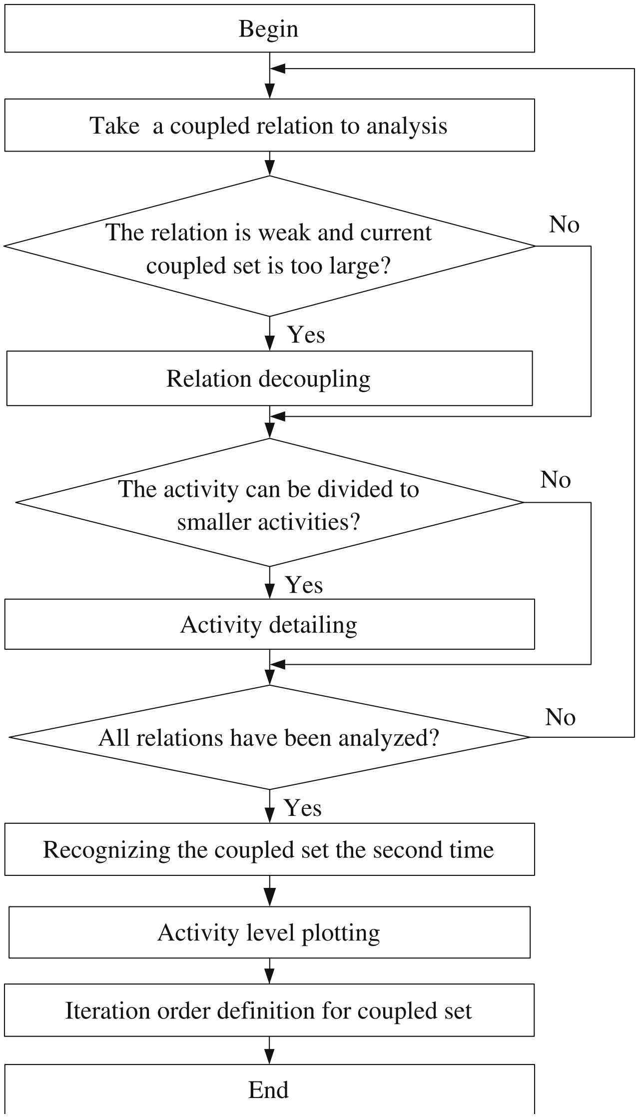


Fig. 4 Design activities reengineering flowchart

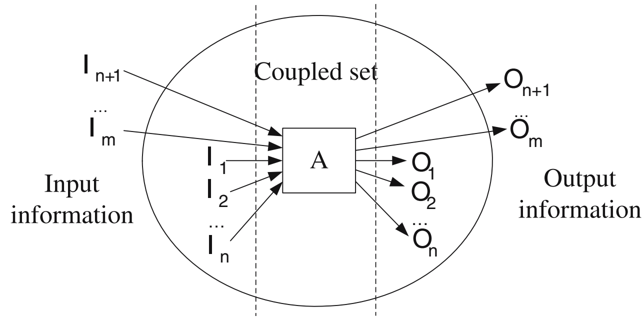


Fig. 5 Information flow of design activity

To analyze design activity quantitatively, the coupling property of an activity can be represented by three factors:

1. Sensitivity factor f_s Activity sensitivity to other activities in the same set is represented by factor f_s. We define f_s as f_s=m/n, in which n refers to total information amount needed by the activity and m refers to local input information.
2. Complexity factor f_c Activity complexity is represented as factor f_c. Factor f_c is mainly decided by required time and cost of this activity. More time and cost a design activity need, more complex it is. We define $f_c \in (0,1)$ and the specific values are decided by domain expert.
3. Affection factor f_a Affection power of an activity to others in the same set is represented as f_a. We define f_a=k, in which k refers to local output information amount. The iteration order is defined as follows. Firstly, input and output information for every activity will be analyzed in detail. The local and global information are recognized separately. Initial iteration order is mainly decided by sum of f_s and f_c. Considering complexity is more important, we define weight values of f_s and f_c as 0.5 and 1 respectively. When computed results of $f_s/2+f_c$ are same, iteration order is decided by affection factor f_a.

4 Case study

Aluminum profile extrusion product development involves product design, manufacturing process design, equipment selection and extrusion die design, etc. Owing to large quantities of geometry parameters and intricate relation among procedures of whole product life cycle, it is very complicated to develop a new aluminum profile product. Up to now to a large extend the development process is empirically based. In this paper a scientific process model based on concurrent engineering philosophy and method proposed above is to provided to improve the development efficiency.

4.1 Design tasks analysis of aluminum profile product development

Design activities and procedures involved in the process of aluminum profile extrusion product development can be classified as five groups and each group can be divided further into several specific activities, as shown in Table 3. There are a total of 19

Table 3 Design activities analysis for aluminum profile extrusion product development

Design activities	Input information	Output information		
G1	Product design and analysis	a_1 Product design	Market requirement and feasible analysis result	Profile geometry, material and dimension tolerance
		a_2 Product feasible analysis	Product information	Product feasibility
G2	Extrusion process design	a_3 Process flow scheduling	Product and enterprise resources information	Extrusion process flow
		a_4 Extrusion method selection	Product information	Extrusion method
		a_5 Extrusion temperature definition	Product information	Extrusion temperature
		a_6 Extrusion speed definition	Product information	Extrusion speed
G3	Extrusion equipment and accessorial tools selection or design	a_7 Press selection	Extrusion force, product area and enterprise resources	Press parameters
		a_8 Tool type selection	Press type and product shape	Type of extrusion ram, container, die plate and gasket
		a_9 Tool dimension definition	Press type, product size, profile positioning and analysis results	Dimension of extrusion ram, container, die plate and gasket
		a_{10} Structure and strength analysis for accessorial tools	Tools dimensions and extrusion force	Structure and strength feasibility of tools
G4	Extrusion die design and analysis	a_{11} Die type selection	Extrusion method and profile geometry	Die type
		a_{12} Profile positioning (for solid die) or porthole design (for porthole die)	Section size of tools and metal flow balance analysis results	Profile position or porthole shape, number and welding chamber geometry
		a_{13} Die hole working dimension	Process parameters, profile geometry and tolerance	Detail die inner dimension
		a_{14} Pre-chamber design	Die inner shape and metal flow balance analysis results	Pre-chamber geometry information
		a_{15} Bearing design	Die inner shape and metal flow balance analysis results	Bearing length and shape
		a_{16} Adjustment structure design	Extrusion metal flow balance analysis	Adjustment structure information
		a_{17} Outer shape design	Die plate and gasket geometry and die strength analysis	Die outer shape and detail dimensions
		a_{18} Material selection	Extrusion force	Die material
		a_{19} CAE analysis	Detail dimension of die	Strength security and metal flow analysis results

activities. To investigate interdependent relationship between activities, the required input information and final output information are analyzed for every activity in detail.

From Table 1, DSM for aluminum profile product development can be gotten as shown in Table 4.

According to Theorem 1, accessibility matrix P and $P \cap P^T$ can be figured out on the basis of DSM as follows:

$$P = \begin{vmatrix}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1\\
&&1&&&&&&&&&&&&&&&&\\
&&&1&&&1&1&1&1&1&1&1&1&1&1&1&1&1\\
&&&&1&1&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&1&1&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&1&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1\\
&&&&&&1&1&1&1&&1&1&1&1&1&1&1&1
\end{vmatrix};$$

Table 4 Design Structure Matrix (DSM) of aluminum profile concurrent product development

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}
a_1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0
a_2	1	0	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0
a_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a_4	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	1	1
a_5	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1
a_6	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	1	1
a_7	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
a_8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
a_9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
a_{10}	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
a_{11}	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
a_{12}	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	0	1
a_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{15}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{16}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{17}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{18}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
a_{19}	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0

$$
P \cap P^{T} = \left| \begin{array}{ccccccccccccccccccc}
1 & 1 & & & & & & & & & & & & & & & & & \\
1 & 1 & & & & & & & & & & & & & & & & & \\
 & & 1 & & & & & & & & & & & & & & & & \\
 & & & 1 & & & & & & & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & & & & & \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & & & & & 1 & & & & & & & & \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & 1 & 1 & 1 & 1 & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{array} \right|
$$

Based on Theorem 2, three coupled activity sets are recognized. They are $C_1=\{a_1,a_2\}$, $C_2=\{a_5,a_6\}$ and $C_3=\{a_7, a_8,a_9,a_{10},a_{12},a_{14},a_{15},a_{16},a_{17},a_{18},a_{19}\}$.

Merging the rows and columns responding to the coupled activity set into one row and column, the reduced matrix of P is obtained. According to Theorem 2 activity levels plotting can be implemented as follows:

$$PE_0 = (6, 1, 3, 2, 2, 1)^T;$$

$$L_1 = \{a_3, a_7\}$$

$$E_1 = (1, 0, 1, 1, 1, 0)^T$$

$$PE_1 = (4, 0, 2, 1, 1, 0)^T$$

$$L_2 = \{a_5, a_6, a_{11}\}$$

$$E_2 = (1, 0, 1, 0, 0, 0)^T$$

$$PE_2 = (2, 0, 1, 0, 0, 0)^T$$

$$L_3 = \{a_4\}$$

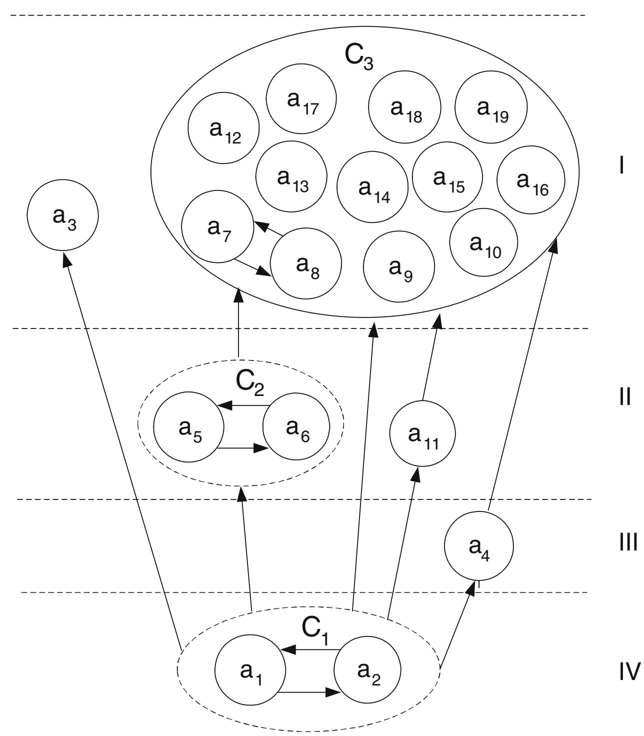


Fig. 6 Initial process model for aluminum profile extrusion product development

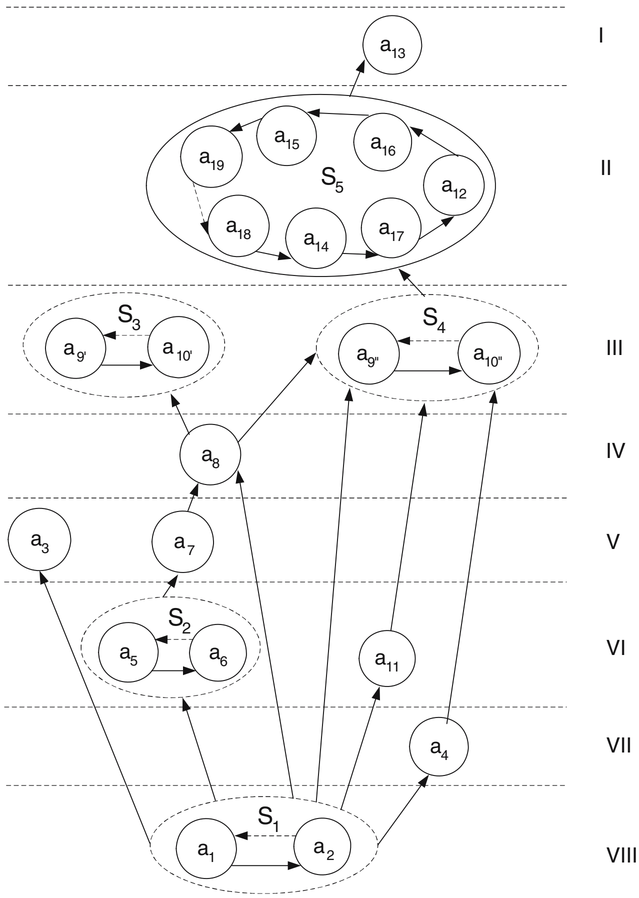


Fig. 7 Optimized process model for aluminum profile extrusion product development

$$E_3 = (1, 0, 0, 0, 0, 0)^T$$

$$PE_3 = (1, 0, 0, 0, 0, 0)^T$$

$$L_4 = \{a_1\}$$

Plotted design process is shown in Fig. 6. It is noted that in set C_3 only one pair of interdependent relations is labeled and the others are omitted.

4.2 Design process optimization

To avoid coupled set too large, which may cause design divergent and cycle time increasing, the third coupled activity set is reengineered particularly. Considering that cavity geometry of extrusion die is decided mainly by profile shape and their positioning and detailed working dimensions affect die strength very little, relation between activity a_{13} and activity a_{19} can be removed. On the other hand, extrusion ram, container and padding relate to inner geometry of die only while die frame relates to the outer shape of die, therefore a_9 (tools dimension definition) can be divided into activity a'_9 (extrusion ram, container and padding dimension definition) and a''_9 (die frame dimension definition) while activity a_{10} (structure and strength analysis for accessorial tools) is divided into a'_{10} (extrusion ram, container and padding analysis) and a''_{10} (die frame analysis). What is more, the interdependent relationship between equipment and tools selection (Group 4) and die design (Group 5) is weak. In practice, equipment and tool selection is usually done before die design. Therefore, activities 7, 8, 9 and 10 are separated from the third set and form a new coupled set. After decoupling and detailing operations, coupled activities and their levels are recognized once more, and the process model is shown in Fig. 7.

4.3 Iteration order definition for coupled activities

There are total four coupled sets in the optimized process model. Coupling properties of all activities have been analyzed to get the rational initial iteration order. The

Table 5 Iteration order definition for coupled activities

Coupled set	S_1		S_2		S_3		S_4										
Sub-activity	a_1	a_2	a_5	a_6	a'_9	a'_{10}	a_7	a_8	a''_9	a''_{10}	a_{12}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}
Sensitivity factor f_s	0.50	1.00	0.50	0.50	0.50	1.00	0.33	0.50	1.00	1.00	0.50	0.66	1.00	1.00	1.00	0.33	1.00
Complexity factor f_c	0.20	0.40	0.10	0.15	0.25	0.20	0.10	0.30	0.20	0.50	0.50	0.30	0.40	0.30	0.15	0.15	0.70
Affection factor f_a	3	2	1	1	3	3	1	4	1	2	2	1	1	3	1	1	2
$f_s/2+f_c$	0.45	0.90	0.35	0.40	0.50	0.70	0.27	0.55	0.70	1.00	0.75	0.63	0.90	0.80	0.65	0.32	1.20
Initial iteration order	[1]	[2]	[1]	[2]	[1]	[2]	[1]	[3]	[6]	[10]	[7]	[4]	[9]	[8]	[5]	[2]	[11]

calculation results of their sensitivity, complexity, affection factors and obtained iteration order are shown in Table 5.

The iteration order is also illustrated in Fig. 7. Based on sufficient analysis of interdependency and information flow, it is believed that the development process provided by Fig. 7 will be more rational and efficient.

5 Conclusions

The objective of concurrent design process modeling and optimization is to improve parallel degree of operations and reduce whole execution time by eliminating iteration and loop times caused by improper operation order. In the paper, a novel quantitative methodology is proposed to analyze and optimize the concurrent engineering process scientifically which can provide rational concurrent operation order and can make the product development more effective and efficient. The design activities in aluminum profile product development are studied particularly and an optimized process model is presented. The model has been applied in a computer-aided aluminum profile product development system prototype successfully. Further research will be focus on the implement of collaboration mechanism among task teams in a distributed development environment.

Acknowledgements We would like to extend our sincere thanks to Professor J. Liu and engineers in Paicheng Aluminum Incorporation, China for their support on domain knowledge. The authors also wish to express their gratitude to the editor, Professor B. Davies and the reviewers of this paper for their valuable suggestion and comments.

References

1. Finger S, Fox MS, Prinz FB, Rinderle JR (1992) Concurrent design. Applied AI 6:257–283
2. Huang GQ, Mak KL (1998) Re-engineering the product development process with ‘design for X’. Proc Inst Mech Eng, Part B 212:259–268
3. Assine A, Falkenburg D, Chelst K (1999) Engineering design management: an information structure approach. Int J Prod Res 37:2957–2975
4. Spinner M (1989) Improving project management skills and techniques. Englewood Cliffs, Prentice Hall, NJ
5. Malmström J, Pikosa P, Malmqvist J (1999) Complementary roles of IDEF0 and DSM for the modeling of information management process. Concurr Eng Res Appl 7(2):95–103
6. Fu Y, Gu X (1997) Discrete maths and its application. Publishing House of Electrical Industry, Beijing, China
7. Steward AD (1981) The design structure system: a method for managing the design of complex systems. IEEE Trans Softw Eng EM-28(3):71–74
8. Steven D (1992) Eppinger, organizing the tasks in complex design projects, NSF Design and Manufacturing System Conference, U.S.A.
9. Kusiak A, Larson TN, Wang J (1994) Reengineering of design and manufacturing process. Comput Ind Eng 26(3):521–536
10. Kusiak A, Park K (1990) Concurrent engineering: decomposition and scheduling of design activities. Int J Prod Res 28 (10):1883–1900
11. Albano LD, Suh NP (1994) Axiomatic design and concurrent engineering. CAD 26(7):499–504
12. Seliger G, Karl H, Weber H (1997) Cooperative design, manufacturing and assembly of complex products. CIRP Ann 46 (1):67–70
13. Xiao R, Fei Q (1997) Application of the improved method in structural modeling to comprehensive management of the Mine Bereaus. System Engineering: Theory and Practice (in Chinese) 17(3):57–62
14. Krishnan V, Eppinger S, Whitney D (1997) A model-based framework to overlap product development activities. Manage Sci 43(4):437–451
15. Smith RP, Eppinger SD (1997) Identifying controlling features of engineering design iteration. Manage Sci 43(3):276–293
16. Krishman V (1996) Managing the simulateous execution of coupled phases in concurrent product development. IEEE Trans Manage 43(2):210–217