DOI: 10.3901/CJME.2013.01.176, available online at www.springerlink.com; www.cjmenet.com; www.cjmenet.com.cn

Multi-segment and Multi-ply Overlapping Process of Multi Coupled Activities Based on Valid Information Evolution

WANG Zhiliang*, WANG Yunxia, and QIU Shenghai

Faculty of Mechanical Engineering, Nanjing Institute of Technology, Nanjing 210013, China

Received September 21, 2011; revised September 20, 2012; accepted September 28, 2012

Abstract: Complex product development will inevitably face the design planning of the multi-coupled activities, and overlapping these activities could potentially reduce product development time, but there is a risk of the additional cost. Although the downstream task information dependence to the upstream task is already considered in the current researches, but the design process overall iteration caused by the information interdependence between activities is hardly discussed; especially the impact on the design process' overall iteration from the valid information accumulation process. Secondly, most studies only focus on the single overlapping process of two activities, rarely take multi-segment and multi-ply overlapping process of multi coupled activities into account; especially the inherent link between product development time and cost which originates from the overlapping process of multi coupled activities. For the purpose of solving the above problems, as to the insufficiency of the accumulated valid information in overlapping process, the function of the valid information evolution (VIE) degree is constructed. Stochastic process theory is used to describe the design information exchange and the valid information accumulation in the overlapping segment, and then the planning models of the single overlapping segment are built. On these bases, by analyzing overlapping processes and overlapping features of multi-coupling activities, multi-segment and multi-ply overlapping planning models are built; by sorting overlapping processes and analyzing the construction of these planning models, two conclusions are obtained: (1) As to multi-segment and multi-ply overlapping of multi coupled activities, the total decrement of the task set development time is the sum of the time decrement caused by basic overlapping segments, and minus the sum of the time increment caused by multiple overlapping segments; (2) the total increment of development cost is the sum of the cost increment caused by all overlapping process. And then, based on overlapping degree analysis of these planning models, by the VIE degree function, the four lemmas theory proofs are represented, and two propositions are finally proved: (1) The multi-ply overlapping of the multi coupled activities will weaken the basic overlapping effect on the development cycle time reduction (2) Overlapping the multi coupled activities will decrease product development cycle, but increase product development cost. And there is trade-off between development time and cost. And so, two methods are given to slacken and eliminate multi-ply overlapping effects. At last, an example about a vehicle upper subsystem design illustrates the application of the proposed models; compared with a sequential execution pattern, the decreasing of development cycle (22%) and the increasing of development cost (3%) show the validity of the method in the example. The proposed research not only lays a theoretical foundation for correctly planning complex product development process, but also provides specific and effective operation methods for overlapping multi coupled activities.

Key words: multi coupled activities, valid information evolution, multi-segment & multi-ply overlapping, development time and cost, trade-off, iteration

1 Introduction

With the increasing personalized market demand and rapid technology development, product functions and implementation technologies are more complex. Such complex products' design involves thousands of tasks or activities, so product development will be faced with two problems.

* Corresponding author. E-mail: wwangzzll@njit.edu.cn
This project is sponsored by Jiangsu Provincial Colleges and Universities Natural Science Foundation of China (Grant No. 08KJD410001), Humanities and Social Sciences Planning Fund of Ministry of Education of China (Grant No. 12YJAZH151), and Humanities and Social Sciences Youth Fund of Ministry of Education of China (Grant No. 12YJCZH209)

First, which execution pattern (sequential, overlapped, or parallel) will be implemented in these tasks development process? Among three execution patterns, overlapping these activities can potentially reduce product development time [1–3], but there is a risk of the additional cost[4–6]. When sequential pattern is replaced by overlapped pattern, as to the upstream activity *A* and the downstream activity *B*, the downstream activity *B* is designed based on initial information provided from the upstream activity *A*. When the upstream activity *A* changes (design change) in subsequent phases (from the beginning at which initial information is provided to the end at which activity *A* is complete), and the upstream information provided again to the downstream activity *B* must change, so the activity *B* must take into account the upstream information change and correspondingly adjust its own design process, this will prolong the development cycle of activity *B*.

Second, there is strong interdependence of design information among activities, and information interdependence leads to design iteration [3, 7–8]. Therefore, the upstream task *A* does not get all of the required information from the downstream activity *B* in advance, or can only obtain part of information. The upstream activity *A* must predict the required information before the design work starts. But when the downstream task *B* is complete, if the design results of activity *B* are inconsistent with prior assumptions, that is, the predicted information is wrong partly or totally, activities *A* and *B* must be redesigned, and the next overall iteration (redesign) of the whole design process (upstream and downstream tasks) is necessary, so the design cycle time will extend and cost increases. We can see, design information interdependence is a key factor when planning product development process.

So far, overlapping strategy of design activities has been researched considerably. FARID, et al[9], used graphical evaluation and review technique (GERT) to confirm that fully paralleling activities were not always appropriate design process management strategy, and there was trade-off between development time and development cost, activity overlapping pattern could reduce the product development cycle[3–4]. SMITH, et al[10], introduced sequential pattern to parallel pattern, and proposed a hybrid model, but this model is not a full sense of the overlapping pattern.

KRISHNAN, et al[6,11], creatively put forward the concepts of upstream evolution and downstream sensitivity, and built a model to plan the overlapped process and reduce product development cycle. LOCH, et al[4], analyzed information communication process between upstream and downstream, and constructed a nonstationary Poisson process to describe design changes, so obtained optimal dynamic communication polity and the optimal overlap. As to a design activity group consisting of an upstream activity and several downstream activities, WANG, et al[12], studied the development process parallelism problem by optimizing the downstream start time. These three models only considered the downstream activities' information dependence to the upstream activities, and did not take the coupling between activities completely into account, but the strong interdependence is a basic feature of the complex products[1]; and at the same time, these models are only the single-phase overlapping models as to two activities (In contrast with the single overlap consisting only of two activities—the single overlap also means the single overlap segment as to two activities in once iteration process, multi-segment and multi-ply overlap consisting of multi coupled activities is the focus of this paper, and multi coupled activities means more than two coupled activities in this paper). As to three general types of information dependencies among the tasks (independent, dependent, and interdependent or coupling) and three different execution patterns (sequential, overlapping, and concurrent)[7], YASSINE, et al[7], planned the design activities execution process; despite most of them are only the single overlap models, we are inspired largely from these conclusions. Based on the results of KRISHNAN and LOCH, et al, CARRASCOSA, et al[13], established a discrete event simulation model to estimate the complete probability of the project containing the coupling activities; although the model takes the multi-segment overlapping of coupled tasks into account, just as the author says, the model does not is applied to the bigger coupled activities set; and at the same time, stochastic model makes the design process' traceability and analysis deteriorate, so it is difficult to effectively manage the development process.

Considering the downstream sensitivity difference to different design changes of the upstream activity, the four factors which determine the downstream sensitivity amount are constructed, and they are transformation processes, lead time, modularity, interaction of components respectively. The research can help practitioners plan the overlapping of design activities effectively[14]. Through analyzing information evolution and design change sensitivity' influence upon overlapping strategy, WANG, et al[15], proposed an overlapping process model considering both activity overlapping and iterations at the same time to capture the uncertain nature and complex interactions of product development processes, and a discrete event simulation algorithm was developed to predict the impacts of process structure on the development lead-time. Although XU, et al[16], considered design iteration and overlap, and presented a development time model, but they didn't analyze in-depth iteration process. As downstream design changes could affect upstream design activities in overlapping stage, XU, et al[17], considered bidirectional information communication in an overall iteration process, and built a development time model to optimize lead-time. But multi-ply overlapping between multi coupled tasks is not taken into account, too.

Most of the above models and methods do not really focus on the development cost and the inherent link between development time and development cost when overlapping activities[18]. However, intense market competition makes enterprises face the dual pressures from development time and development cost. So, based on the prediction incorrect probability function of overlapping phase, ROEMER, et al[18], developed two overlapping models by nonlinear constraint programming model. One model minimizes development cycle time under overlapping costs constraint, and the variation minimizes development cost under overlapping cycle time constraint. These models emphasize the trade-off between time and cost, and can determine the appropriate overlapping strategy[19]. CHAKRAVARTH[3] defined the risk and mapping functions between design cycle and build cycle, and presented the single overlap model, and then extended to the multiple overlap model and gain optimal overlap. These models consider the link between development cycle time and cost, and are multiple overlap models, but they do not contain information feedback and resulting overall design iterations. The previous research considers information interdependence, and present development time and cost models, but these models are based on the single overlap [8].

Although overlapping activities can reduce the product development cycle time, but the complexity of overlapping problem calls more intensive and more in-depth researches, and strong coupling of complex products also add more difficulties, especially considering the trade-offs between development time and cost, overlapping problem of multi coupled activities still remains an open question. The following questions are needed to be explored: (1) As to overlapping problem of multi-coupled activities, how does overlapping degree between design activities influence development cycle time and cost, and whether there is trade-off between development cycle time and design cost or not; (2) How to effectively plan overlapping process of multi coupled tasks to gain this appropriate trade-off and optimal overlap. Solutions to the problem are expected to help designers plan and manage complex systems' design process, and support enterprises schedule product development time and development costs effectively at the same time.

2 Single Overlap Process Planning

2.1 Nominal information evolution (NIE) and valid information evolution (VIE)

As is mentioned above, under overlapping pattern, a downstream task must start based on the initial information of the upstream activities; if the upstream design changes occur at later phase, the downstream activity duration will be extended. Different levels of upstream information changes will result in downstream activity duration extend correspondingly. If the upstream activity can accumulate more and more accurate information to the downstream activity before overlapping (naturally, the upstream activity just accumulates less information in the following overlapping phase), the probability that the upstream information changes occur will be smaller, so the impact on the downstream activity from the upstream design changes will become also smaller. On the other hand, if the downstream activity can have a greater ability to accommodate the upstream information changes, i.e., the downstream activity possesses higher robustness to the upstream information changes, this impact on the downstream will be more weakened because of this robustness. Therefore, it is necessary to analyze the upstream information evolution process and the situation of impact on the downstream activity from information changes[6].

In the next section, we will construct nominal information evolution (NIE) and valid information evolution (VIE) to present the information accumulation progress, and at the same time the link between information evolution degree function and activities rework probability, then explore the upstream design changes' impact on the downstream by a non-stationary Poisson process

As product development is not only a process of information processing, but also an upstream information accumulation process for the downstream activities, from this point of view, we can use the information amount that the upstream accumulates for the downstream to measure the evolution degree of the upstream activity.

Definition 1: Let $I(t)$ denotes the information amount that the upstream accumulates for the downstream at time t in design process, I_D denotes the whole information amount that the upstream accumulates for the downstream after completion of the upstream activity, the information evolution degree $h(t)$ of the upstream is defined as: $h(t)=I(t)/I_D$.

From the above analysis, in the definition of information evolution degree, we only consider the upstream information accumulation progress for its downstream activity, and did not refer to the information interdependence between the upstream and the downstream. While this information interdependence exists, the concept of this information evolution degree remains to be further expanded.

Information interdependence between tasks means that any activity's design must rely on information that other coupled activities provide. When the upstream activity's design starts, designers must forecast the downstream activity' out information as the input information of the upstream activity (multi-disciplinary team work will help to transfer more downstream output information to upstream activity; In fact, on this basis, the upstream activity predict downstream out information, but this is a static exchange of information, and can not replace dynamic exchange of information in the design process[20]) In this case, the information that the upstream activity provides to the downstream in the overlapping phase is based on this forecast. When there is biased forecast, the information that the upstream activity provides to the downstream have naturally false and irrational components. So, we classify and label the information that the upstream activity provide to the downstream by nominal information and valid information[8]; while nominal information refers to all of the information provided to the downstream activity by the upstream, valid information refers to real information contained in nominal information and only valid information is the information that downstream activity really requires. And at the same time, we derive nominal information evolution degree and valid information evolution degree.

Definition 2: Let $NI(t)$ denotes the nominal information amount that the upstream accumulates for the downstream at time t in design process, NI_D denotes the whole nominal information amount that the upstream accumulates for the downstream after completion of upstream activity, $EI(t)$ denotes the valid information amount that the upstream accumulates for the downstream at time t in design process, EI_D denotes the whole valid information amount that the upstream accumulates for the downstream (after receiving EI_D information, the downstream activity can be completed once without iteration design, this looks like there is no interdependence between them), degree of nominal information evolution (denoted by NIE degree) is defined as $g(t) = NI(t)/NI_D$, and degree of valid information evolution (denoted by VIE degree)is defined as $h(t) = EI(t)/EI_D$.

NIE degree describes the upstream information accumulation progress for the downstream activity, VIE degree actually reflects the real information accumulation situation. Obviously, while there is no information interdependence between tasks, because all the information accumulated in the upstream activity are consistent with the fact, NIE degree and VIE degree are the same, i.e., information evolution degree defined in definition 1. Thus, after the completion of upstream activity, NIE degree and VIE degree reach maximum 1. On the other hand, while information interdependence exists, after the completion of the upstream activity, the upstream activity has accumulated all the necessary information for downstream activity and NIE degree reaches maximum 1. However, as nominal information contains incorrect components, VIE degree does not reach maximum 1, so design iterations necessitate, and the upstream activity gradually provides more real information to downstream activity in subsequent design iterations process until the upstream activity accumulates all the valid information after completion of iteration process.

We can see, as to the whole design process consisting of the n times overall iteration (for the convenience of presentation, the first design process of activities is referred as the first iteration), although NIE degree reaches maximum 1 at each iterative process, each iteration process accumulates only part of the real valid information, the next iteration continues to accumulate the valid information based on the accumulated real information. When the n times iterative process is complete and all the valid information has been gradually accumulated, VIE degree reaches the maximum 1. They are shown in Fig. 1.

Further, let $g_i(t_i)$ is NIE degree function for the ith overall iteration at t_i (t_i is time variable at the ith time iteration), $h_i(t_i)$ is VIE degree function, D_i is the design duration. While $g_i(0)$, $h_i(0)$ indicates NIE degree and VIE degree at the beginning of the ith iteration respectively, $g_i(D_i)$, $h_i(D_i)$ indicates NIE degree and VIE degree at the end of the ith iteration respectively. Based on the above analysis, we can assume $g_1(0) = h_1(0) = 0$ (without loss of generality); at the same time, $g_i(D_i) = 1$, $h_n(D_n) = 1$, $g_i(0) = h_i(0) = h_{i-1}(D_{i-1})$ ($h_0(0) = 0$) and $h_i(D_i) - h_{i-1}(D_{i-1})$ indicates the accumulated real amount of information at the ith solution process ($i = 1, 2, \cdots, n$).

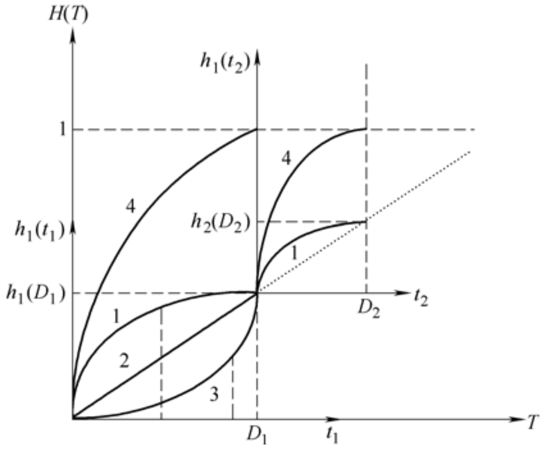

Fig. 1. Coupled activities information accumulation process

1. Fast VIE degree function; 2. Even VIE degree function; 3. Slow VIE degree function; 4. NIE degree function

Because different design process have different information evolution progress, and thus there is fast evolution progress, even evolution progress and slow evolution progress. But for a certain design process, the information evolution degree will get the concrete value to a different time point. Therefore, assuming that the information accumulation progress can be represented by the exponential function $(t/D)^m$ in each solution process without loss of generality, where m is the speed ratio of information accumulation. If $m > 1$, $(t/D)^m$ corresponds to slow evolution progress; if $m = 1$, $(t/D)^m$ corresponds to even evolution progress; if $m < 1$, $(t/D)^m$ corresponds to fast evolution progress (shown in Fig. 1).

We must notice that any valid information progress can be described by combination of multi exponential functions when valid information accumulation progress can satisfy consistency, i.e., the accumulation amount of valid information is increasing along design process. So, the valid information evolution functions that we define represent the general information accumulation progress.

Thus, VIE degree function at the ith iteration solution is

$$h_i(t_i) = h_{i-1}(D_{i-1}) + [h_i(D_i) - h_{i-1}(D_{i-1})] \times \left(\frac{t_i}{D_i}\right)^m. \quad (1)$$

If let $H(T)$ represents the real information accumulation progress function at the entire iterative process (where T is time variable in the entire iterative process), obviously, it consists of each iteration's VIE degree function $h_i(t_i)$ family together, viz.

$$H(T) = \left\{ h_i(t_i) \middle| i = 1, 2, \cdots, n, t_i \in [D_{i-1}, D_i] \right\}, T \in \left[0, \sum_{j=0}^{n} D_j\right]. \quad (2)$$

While $T \in \left[\sum_{j=0}^{i-1} D_j, \sum_{j=0}^{i} D_j\right]$, $H(T)$ can be represented concretely by $h_i(t_i)$. While $T=0$, $H(0)=0$. If $T=T_D=\sum_{j=0}^{n} D_j$, $H(T_D)=1$. When information interdependence does not exist between the tasks, $n=1$, $H(T)=h_1(t_1)$, $h_1(D_1)=1$.

2.2 VIE degree functions solution

Assuming the information coupling intensity between two tasks A and B, viz. the rework possibility matrix is shown in Table 1. The task execution order is from A to B. As the activity A rework possibility after completion of activity B is a_{12}, that is, the possibility that activity A needs to re-design is a_{12}, there is the possibility $1-a_{12}$ that the activity A does not re-design. On the other hand, the activity A design process is the information accumulation process for the activity B, so the activity A accumulates only $\beta_{ab1}=1-a_{12}$ percent valid information for activity B after completion of activity A, and a_{12} percent valid information still needs to accumulate at the subsequent iteration process. Activity B design work starts after B gets β_{ab1} percent valid information activity from activity A, and when B is completed, the valid information (percentage) that activity B accumulates for A is $\beta_{ba1}=1-(1-\beta_{ab1})\times a_{21}=1-a_{12}\times a_{21}$.

Table 1. Rework probability matrix

Activity	A	B
A	0	a_{12}
B	a_{21}	0

So, two types of the valid information (percentage) that the activity A accumulates for the activity B and the activity B accumulates for the activity A in the ith iteration are respectively:

$$\beta_{abi}=1-(1-\beta_{ba,i-1})\times a_{12}, \beta_{bai}=1-(1-\beta_{abi})\times a_{21}, \quad (3)$$

where integer $i \geqslant 1$, $\beta_{ba0}=0$.

Comparing β_{abi} and $h_i(D_i)$, as all they indicate valid information accumulation degree, therefore, $\beta_{abi}=h_i(D_i)$. So, as to coupled activities A and B, Eq. (1) can be rewritten as

$$h_{abi}(t_{abi})=\beta_{ab,i-1}+(\beta_{abi}-\beta_{ab,i-1})\times\left(\frac{t_{abi}}{D_{ai}}\right)^m, \quad (4)$$

where $h_{abi}(t_{abi})$ denotes the VIE degree at t_{abi}.

NIE degree and VIE degree really describe the information accumulation progress, the concept presentment and corresponding function construction lay a theoretical foundation for planning overlapping process of coupled activities.

2.3 Overlap process planning method

2.3.1 Overlap process math description

As the information interdependence between the coupled tasks, the designers need to exchange design information at certain manner; and at the same time, designers must also review the completed work; so, they can determine whether the completed work is redesigned, and pass the review results to the coupled tasks. Therefore, product development process planning not only requires making sure a clear exchange way of information, but also considering design review setup. We define the information communication way and design review as follows.

(1) There is an information transfer from the upstream to the downstream when overlapping starts, the initial information that the upstream activity accumulates is provided to the downstream activity.

(2) There is a design work process review after the completion of the upstream, and so designers can judge whether the initial information is correct;

(3) In an iterative solving process, information exchange is one-way, i.e., process review results do not affect the upstream activity completed.

(4) There is an iteration review (different to process review) to all of the activities in the task set and then designers can obtain the information that the upstream and the downstream activities need at the next iteration.

The reasons to make these settings are as follows.

(1) Any information exchange and design review will take time consumption. In particular, too frequent design review will consume too much time in unnecessary assessment, which will result in the extension of the design cycle; and too small design review will not be able to detect the design error, thus increase the re-design workload, the design process will naturally extend[4].

(2) Input and output information interdependence between the coupled tasks; so, whether in an iterative process, or between the iterations, any task changes will lead others design changes. At overlapping phase, the downstream activity design is just beginning, so information that the downstream can provide is not only less in quantity, but has greater uncertainty, while such information is passed to the upstream design task, greater design risks will appear in design process.

(3) In order to avoid excessive overlapping which will lead the downstream activity duration to prolong too much (later chapters will argue that the multi-ply overlapping process of multi coupled activities will weaken the overlapping effect on the development cycle time reduction), the downstream activity starts just when the upstream activity has been accumulated most of the information for the downstream design activity; this will greatly reduces the probability of information transfer from

the downstream to the upstream at overlapping phase. On the other hand, when the upstream design activity passes information to the downstream activity several times, it is possible that the previous modification do not be completed, and subsequent modification request has been reached. This phenomenon will result in repetitious redesign of the downstream activity.

So, as to the overlapped execution of multi coupled tasks, the information exchange route is shown in Fig. 2.

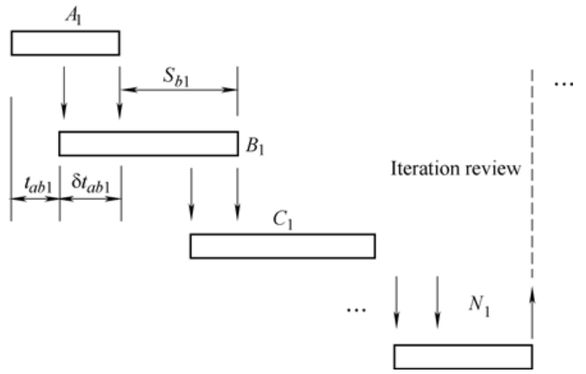

Fig. 2. Information exchange route

In the overlapping phase, if the design changes of the upstream activity do not occur, i.e., the upstream activity's information outputted to the downstream does not change, the downstream activity will be executed normally and its duration remains unchanged. If the upstream design changes occur, the upstream information provided to the downstream activity also naturally changes, the downstream activity will certainly need to rework partly or totally and the downstream design duration is extended accordingly. If a design change is regarded as an event, we can use the nonstationary Poisson Process to describe the number of design change events [5].

According to Poisson process, let X be the number of design change events in period $(t, t+\delta t)$, X is a random variable, $X=\{0,1,2,\cdots,k,\cdots\}$. Thus, at period $(t, t+\delta t)$ the probability that design changes appear just k-times follows the Poisson distribution with the parameters of λ:

$$P(X=k)=\frac{(\lambda\cdot\delta t)^k}{k!}\exp(-\lambda\cdot\delta t),\ k=0,1,\cdots$$

If $k=0$; $P(X=0)=\exp(-\lambda\cdot\delta t)$, it represents the probability that any design change does not also occur at period $(t, t+\delta t)$.

The $\lambda\cdot\delta t$ is the parameter of Poisson distribution and means the average number of events per unit time, when the rate λ is a constant value, Poisson process is stationary; If the rate λ is time-dependent intensity function, $\{\lambda(t), t\geqslant 0\}$, the Poisson process transforms to a non-homogeneous Poisson process:

$$P(X=k)=\frac{w^k}{k!}\exp(-w),\ k=0,1,\cdots$$

where the Poisson distribution parameter is $w=\int_t^{t+\delta t}\lambda(u)\mathrm{d}u$. And thus, the probability that the upstream design change does not occur equals to $P(X=0)=\exp(-w)$.

As to the coupled activities, when the Poisson process is used to describe the number of the upstream activity design changes in the design process consisting of the n-times overall iteration, the probability that the upstream activity does not change in the overlap period $(t_{ai}, t_{ai}+\delta t_{ai})$ of the ith iteration process is $P(X=0)$; Accordingly, the probability that the upstream design changes exist in the overlap period is $1-P(X=0)$.

Poisson Process describes the possibility of design changes at the design evolution of the upstream activity, and different types of upstream activities follow different Poisson distributions. VIE degree really describes valid information accumulation progress, and different development processes have different information evolution degree functions. But design changes result from the immaturity of the design process and the insufficiency of the accumulated information, the parameter of Poisson Process must associate with VIE degree function.

Let $\lambda(t)=H'(T)/H(T)$ (this definition is consistent with the rate λ meaning in Poisson distribution), the parameter of the non-homogeneous Poisson process is

$$w=\int_T^{T+\delta T}\lambda(u)\mathrm{d}u=ln\frac{H(T_D)}{H(T)}=$$
$$ln(H(T_D))-ln(H(T))=-ln(H(T)),$$

where $T+\delta T=T_D$. So,

$$P(X=0)=\exp(-w)=H(T). \tag{5}$$

This indicates that the probability that the upstream design changes do not occur equals to VIE degree at the beginning of the overlapping period $(t_{ai}, t_{ai}+\delta t_{ai})$ in the i-th iteration. In fact, only valid information can eliminate uncertainty of development process. $H(T)=\{h_i(t_i)\,|\,i=1,2,\cdots,n, t_i\in[0,D_i]\}$ represents exactly valid information accumulation progress in the overall iteration solving process consisting of n times iterations

2.3.2 Determining process parameters of the downstream activity

First, the descriptions of each variable will be given. t_{abi} is the beginning time of overlapping in the ith iteration process, δt_{abi} is the time period length of overlapping stage, p_{abi} is the probability that the upstream design changes do not occur, $p_{abi}=P(X=0)\,|\,_{t_{abi}}$; according to Eq. (5), p_{abi} equals VIE degree at t_{abi}, and it can be calculated by Eq. (4). S_{bi} is the design time that the downstream activity still requires after completion of the upstream activity at the ith iteration, $E(S_{bi})$ is its expectations. $E(B_i)$ is the downstream duration expectation at the ith iteration. D_{ai} and D_{bi} are the nominal durations of activity A and B at the ith iteration when all requisite information is available, respectively.

If the design changes of upstream activity A do not occur in the ith iteration overlapping, the design cycle time D_{bi} of the downstream activity B remains unchanged, that is, after completion of activity A, the design time that activity B still requires is $S_{bi}^1 = D_{bi} - \delta t_{abi}$. If the upstream design changes appear, the downstream still required design cycle S_{bi}^2 naturally extends, $S_{bi}^2 > D_{bi} - \delta t_{abi}$. As the upstream design changes is diverse, the downstream design cycle have different extension, and S_{bi}^2 is a stochastic variable, Let $E(S_{bi}^2)$ is its expectation correspondingly. According to the above, as the probability that the upstream design changes do not occur is $p_{abi} = P(X=0)|_{t_{abi}}$, the probability that design changes appear is $1-p_{abi}$, so the expectation that the downstream still requires design cycle is [7]

$$E(S_{bi}) = p_{abi} \times S_{bi}^1 + (1-p_{abi}) \times E(S_{bi}^2) = p_{abi} \times (D_{bi} - \delta t_{abi}) + (1-p_{abi}) \times E(S_{bi}^2). \quad (6)$$

The downstream design duration expectation is

$$E(B_i) = \delta t_{abi} + E(S_{bi}). \quad (7)$$

2.4 Single overlap models

As to the single overlap, the upstream activity A can generate various types of design changes in the overlapping phase, assuming that these changes are equal probability events, the design time S_{b1}^2 that the downstream activity B needs still after completion of activity A at the first overlapping phase is

$$S_{b1}^2 = D_{b1} - \delta t_{ab1} + \alpha \times \delta t_{ab1},$$

where, the parameter α follows uniform distribution in the interval [0, 1]. So,

$$E(S_{b1}^2) = D_{b1} - 0.5 \times \delta t_{ab1}.$$

From Eqs. (6) and (7), we can obtain

$$E(S_{b1}) = p_{ab1} \times (D_{b1} - \delta t_{ab1}) + (1-p_{ab1}) \times E(S_{bi}^2) = D_{b1} - 0.5 \times \delta t_{ab1} - p_{ab1} \times 0.5 \times \delta t_{ab1}, \quad (8)$$

$$E(B_1) = \delta t_{ab1} + E(S_{b1}) = D_{b1} + 0.5\delta t_{ab1} - p_{ab1} \times 0.5\delta t_{ab1}. \quad (9)$$

Similarly, at the ith iteration process,

$$E(S_{bi}) = D_{bi} - 0.5 \times \delta t_{abi} - p_{abi} \times 0.5 \times \delta t_{abi}, \quad (10)$$

$$E(B_i) = \delta t_{abi} + E(S_{bi}) = D_{bi} + 0.5 \times \delta t_{abi} - p_{abi} \times 0.5 \times \delta t_{abi}. \quad (11)$$

As the iteration process is a learning process, the design duration will decrease when the number of iterations increases. Usually, their relationship can be described as exponent function [21]:

$$D(i) = D(1) \times \exp(-r(i-1)), \quad (12)$$

where r is the decay rate constant parameters.

3 Multi Coupled Activities Overlapping Model and Analysis

As complex products often contain a large number of design activities and they are interdependent, we will emphatically discuss overlapping situation of multi coupled activities in the next section.

3.1 Multi-segment and multi-ply overlapping model

While the single overlap only consists of two activities and has only an overlap segment, multi coupled activities (It means more than two activities in this paper) can forms multi overlap segments in once iteration process, and more than two activities are likely to overlap each other at the same time. So, multi-coupled activities' overlapping should have the following two characteristics.

(1) Multi-segment overlap, viz. there will be more than one single overlap segment in once overall iteration process.

(2) Multi-ply overlap, viz. there is an overlapping scenario that more than two activities overlap each other at some overlapping phase.

As to these two characteristics, in order to illustrate the overlapping planning process and methods, we will start from the simplest multi-overlapping, shown in Fig. 3.

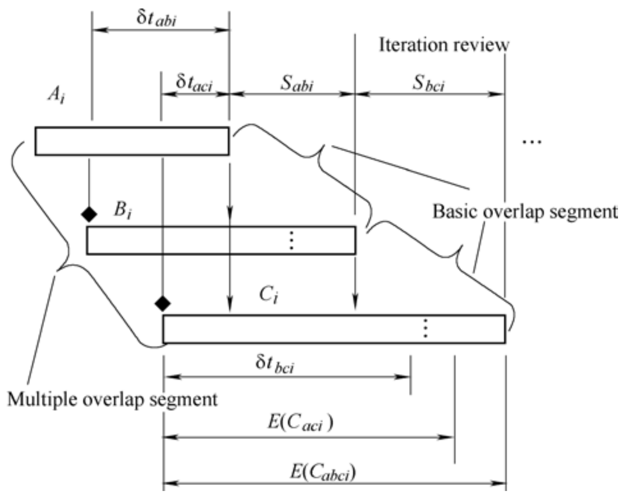

Fig. 3. Multi-segment & multi-ply overlapping process

We can see from Fig. 3 that three activities constitute three overlap segments (processes): A–B, A–C, and B–C.

And overlap segments *A–B* and *A–C* again overlap each other, so the above two characteristics have get satisfied.

According to the aforementioned, at the *i*th overall iterative design process, by Eqs. (10) and (11), from the overlap segment *A–B*, we can obtain the expectation of the design time that the downstream activity *B* still requires after completion of upstream activity *A*, and the expectation of the downstream whole design duration at the *i*th iteration process, they are respectively:

$$E(S_{abi}) = D_{bi} - 0.5\delta t_{abi} - p_{abi} \times 0.5\delta t_{abi},$$
$$E(B_{abi}) = \delta t_{abi} + E(S_{abi}) = D_{bi} + 0.5\delta t_{abi} - p_{abi} \times 0.5\delta t_{abi}.$$

From the overlap segment *A–C*, similarly, we can obtain:

$$E(S_{aci}) = D_{ci} - 0.5\delta t_{aci} - p_{aci} \times 0.5\delta t_{aci},$$
$$E(C_{aci}) = \delta t_{aci} + E(S_{aci}) = D_{ci} + 0.5\delta t_{aci} - p_{aci} \times 0.5\delta t_{aci}.$$

In order to ensure the continuity of design activities and avoid excessive overlap between activities, the downstream activity *C* should not be completed when the upstream activity *B* does not provide the final information yet, viz. $E(S_{abi}) < E(S_{aci})$.

On this basis, we can get from the overlap segment *B–C*:

$$E(S_{bci}) = E(C_{aci}) - 0.5\delta t_{bci} - p_{bci} \times 0.5\delta t_{bci},$$
$$E(C_{abci}) = \delta t_{bci} + E(S_{bci}) = E(C_{aci}) + 0.5 \times \delta t_{bci} - p_{bci} \times 0.5 \times \delta t_{bci}.$$

We can see, when the overlapping process *B–C* is planned, we first consider the design changes' impact on activity *C* from the activity *A* , and on this basis, then from the activity *B*. $E(C_{abci})$ is the activity *C* duration expectation when taking into account the impacts from activities *A* and *B*.

Therefore, the development time and development cost of the whole task set at the *i*th overall iteration are respectively as follows:

$$T_i = D_{ai} + D_{bi} + D_{ci} - (1 + p_{abi}) \times 0.5\delta t_{abi} + (1 - p_{aci}) \times 0.5\delta t_{aci} - (1 + p_{bci}) \times 0.5\delta t_{bci}, \quad (13)$$

$$C_i = D_{ai} + D_{bi} + D_{ci} + (1 - p_{abi}) \times 0.5\delta t_{abi} + (1 - p_{aci}) \times 0.5\delta t_{aci} + (1 - p_{bci}) \times 0.5\delta t_{bci}, \quad (14)$$

where the development cost is measured by time unit, it means time cost.

The whole iterative development time and development costs of the task set are respectively as follows:

$$T_s = \sum_{i=1}^{n} T_i, C_s = \sum_{i=1}^{n} C_i. \quad (15)$$

3.2 Models further analysis

3.2.1 Models construction analysis

In order to more clearly explain multi-segment & multi-ply overlap and our solving ideas, we will sort all overlap segments of multi coupled activities. We regard a single overlap as an ordinal overlap while this overlap occurs according to order $A \to B \to C \to \cdots$ (such as the overlap *A–B*, the overlap *B–C* in this case), and the other type as non ordinal overlap (such as the overlap *A–C* in this case). In essence, a non ordinal overlap means that there is a multi-ply overlap scenario in once overall iteration process, viz. on the basic of ordinal overlap segment, there are more than one overlap segments at the same time. So, we regard ordinal overlap segment as basic overlap segment, and excess of basic overlap segment as multiple overlap segment. Therefore, the overlap segment *A–B* and the overlap segment *B–C* are all basic overlap segments, but the overlap segment *A–C* is multiple overlap segment in this case. They are also shown in Fig. 3.

From Eq. (13), we can see, as $(1 + p_{abi}) \times 0.5\delta t_{abi}$, $(1 + p_{bci}) \times 0.5\delta t_{bci}$, respectively represents the design time decrement caused by the basic overlap processes *A–B* and *B–C*, and $(1 - p_{aci}) \times 0.5\delta t_{aci}$ represents the design time increment caused by the multiple overlap process *A–C*, thus as to the whole task set, the total decrement of development time at the *i*th iteration process is the sum of the time decrement caused by basic overlap segments, and minus the incremental design time caused by multiple overlap segments.

We can see from Eq. (14), as $(1 + p_{abi}) \times 0.5\delta t_{abi}$, $(1 - p_{aci}) \times 0.5\delta t_{aci}$, $(1 + p_{bci}) \times 0.5\delta t_{bci}$, respectively represents the incremental design time costs caused by the overlapping processes *A–B*, *A–C* and *B–C*; thus, as to the coupled tasks set, the total increment of development time costs is the sum of the incremental costs caused by overlapping at the *i*th iteration process.

As the three coupled tasks overlapping model has covered a variety of multi-segment and multi-ply overlapping, we can conclude:

Conclusion 1: As to multi-segment and multi-ply overlapping of multi coupled activities, the total decrement of the task set development time is the sum of the time decrement caused by basic overlap segments, and minus the sum of the time increment caused by multiple overlap segments; the total increment of development cost is the sum of the cost increment caused by all overlap process.

3.2.2 Overlap degree analysis

As starting time t_{abi} of the *i*th overlap represents overlap degree of this overlap process, we can see from Eqs. (13)–(15), the relation of the product development time and overlap degree of each overlap process is just the relation of $T_s = \sum_{i=1}^{n} T_i$ and starting time t_{abi} of this overlap process, the relation of the product development cost and overlap degree is then the relation of $C_s = \sum_{i=1}^{n} C_i$ and starting time t_{abi}.

As $\partial T_s / \partial t_{abi} > 0$, $\partial T_s / \partial t_{bci} > 0$, (for easier readability of the text, proofs are shown in the appendix 1), while starting time t_{abi} of the basic overlap process decreases, i.e., the overlap degree of basic overlap segment increase, product development cycle will decrease (regarded as LEMMA 1).

As $\partial T_s / \partial t_{aci} < 0$, (proof is shown in the appendix 2), while starting time t_{aci} of the multiple overlap process decreases, i.e., the overlap degree of multiple overlap segment increase, product development cycle will increase (regarded as LEMMA 2). So, we can conclude as follows.

Proposition 1. The multi-ply overlap of the multi coupled activities will weaken the basic overlap effect on the development cycle time reduction.

Despite so, we still can prove, overlapping the multi coupled activities will decrease product development cycle (regarded as LEMMA 3)(Proof is shown in the appendix 3).

On the other hand, as $\partial C_s / \partial t_{abi} < 0$, $\partial C_s / \partial t_{bci} < 0$, $\partial C_s / \partial t_{aci} < 0$ (proofs are also shown in the appendix 3), while overlap degree of any type of overlap increases, product development cost must increase (regarded as LEMMA 4).

According to the aforementioned, we can prove and obtain the following.

Proposition 2: Overlapping the multi coupled activities will decrease product development cycle, but increase product development cost. And there is trade-off between development time and cost.

So, blindly increasing the activities overlap degree is not always a good decision.

If there are not multiple overlap segments, and multi coupled activities only have some basic overlap segments, shown by solid line box in Fig. 4, from Eqs. (13) and (14), as to the task set consisting of *m* activities, the development cycle time and development costs are respectively as follows:

$$T = \sum_{i=1}^{n} T_i = \sum_{i=1}^{n} \left(\sum_{k}^{m} D_{ki} - \sum_{k}^{m-1} [(1 + p_{k,k+1,i}) \times 0.5\delta t_{k,k+1,i}] \right), \tag{16}$$

$$C = \sum_{i=1}^{n} C_i = \sum_{i=1}^{n} \left(\sum_{k}^{m} D_{ki} + \sum_{k}^{m-1} [(1 - p_{k,k+1,i}) \times 0.5\delta t_{k,k+1,i}] \right). \tag{17}$$

So, we can obtain the following.

Conclusion 2: As to multi coupled activities overlap process only consisting of basic overlap segments, the total decrement of the task set development time is the sum of the time decrement caused by all overlap segments; the total increment of development cost is the sum of the cost increment caused by all overlap processes.

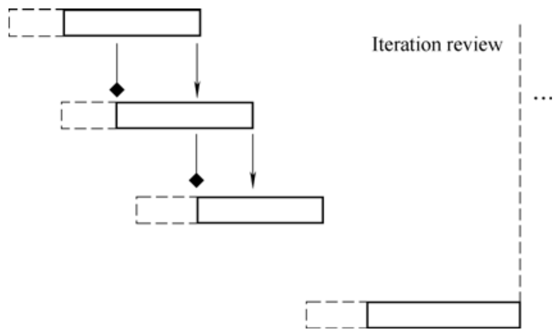

Fig. 4. Only basic overlap processes of multi coupled activities

3.3 Methods of slackening and eliminating multiple overlap

As the multi-ply overlap process of multi coupled activities will weaken the basic overlap effect on the development cycle time reduction, this limits the excessive overlapping of the design activities. In order to further shorten the development cycle, we present two methods to slacken and eliminate the multi-ply overlap effects.

(1) Stage decomposition method. As each activity design process can be divided into two stages to both the setup stage and the core stage [22], assuming that only necessary preparatory work is executed in the preparation stage, when only the downstream activity enters the core stage, it starts to accept the upstream initial information. In this case, we can let the preparation phase have the multiple overlap, and the core stage only have the single overlap, this manner can effectively avoid negative effects of the multiple overlap, as shown in Fig. 4 (The dotted line box indicates the preparation phase; solid line box indicates the core phase). At the same time, this method can also shorten product development cycle very much.

(2) Execution sequence optimization method. As all activities have different development cycles, when the design activity with a shorter development cycle starts firstly, the subsequent development activities can also early start and so improve the overlap degree, the development cycle of the overall task set can be much more shortened. On the other hand, when the design activity with a smaller rework probability starts firstly, the number of the overall iteration can be reduced[23]. Therefore, considering comprehensively all activities development cycle and the rework probability, we propose two sequence optimization algorithms of the coupled activities to determine the preferred execution sequence of the coupled task set[24–25]. As the method increases the overlap degree and effectively decreases the number of the overall iteration, optimal development cycle could be obtained.

4 An Illustrative Example

Stonecity Electronic Technology Company plans to provide a kind of vehicle device for a collaboration company. The vehicle device is a typical complex product, and its overall design can be grouped into 9 activities, namely: an operator panel, a grab device, a cooling system, a support body, a transmission agent, a manipulator, a motor and reducer, a hydraulic system, a connecting rod. Among them, the operator panel, the grab device, the cooling system, the support body (denoted by A, B, C, D, respectively) constitutes a vehicle upper subsystem, and this subsystem is designed by an integrated development team. According to the assessment of the experts group, this subsystem's coupling strength matrix on information is shown in Table 2, and the nominal development durations of the design tasks are respectively: {50, 45, 25, 30}.

Table 2. Rework probability matrix

Activity	A	B	C	D
A	0	0.3	0.05	0.09
B	0.38	0	0.05	0.09
C	0.3	0.3	0	0.35
D	0.22	0.22	0.05	0

The downstream work starts when the upstream nominal information accumulation reaches 80%. At the first overall iteration process, as to activity A, as $\beta_{ab1} = 1 - 0.3 = 0.70$, $p_{ab1} = \beta_{ab1} \times 80\% = 0.56$, from Eq. (4), $\beta_{ab1} \times (t_{ab1} / D_{a1})^{0.5} = 0.56$, $t_{ab1} = 32$, so $\delta t_{ab1} = 50 - 32 = 18$. As to B, $E(S_{ab1}) = D_{b1} - 0.5\times\delta t_{ab1}\ p_{ab1}\times 0.5\delta t_{ab1} = 45 - 0.5\times(1+0.56)\times 18 = 30.96$, $E(B_{ab1}) = D_{b1} + 0.5\delta t_{ab1} - p_{ab1}\times 0.5\delta t_{ab1} = 45 + 0.5\times(1-0.56)\times 18 = 48.96$.

Then similar steps can be used to B–C and C–D overlapping processes. The first overall iteration data are shown in Fig. 5. The first overall iteration time: 125.05 (time), and iteration time costs: 155.62 (time units).

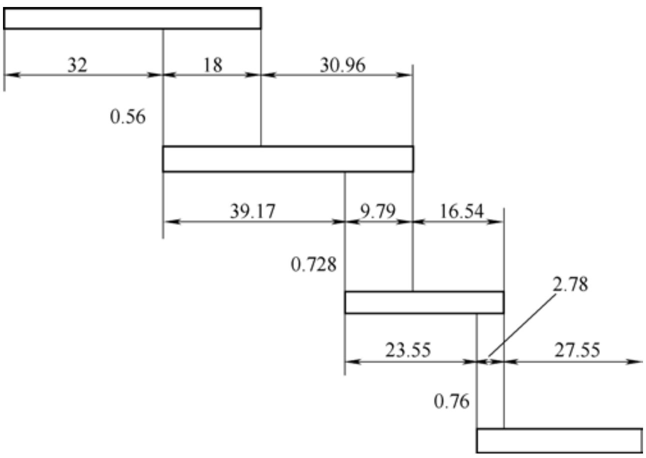

Fig. 5. First overlapping process data

Because of the interdependence between activities, the design process requires the second whole iteration process to accumulate more valid information. It is assumed that the decay rate of the design durations is 0.3, i.e., $r = 0.3$, and so the nominal development durations of the design tasks in the second iteration are respectively: {37.04, 33.33, 18.52, 22.22}. As to A, $\beta_{ab1} = 0.70$, $\beta_{ab2} = 0.965\,8$, $p_{ab1} = \beta_{ab1} + (\beta_{ab2} - \beta_{ab1})\times 80\% = 0.912\,6$, from Eq. (4), $t_{ab2} = 23.705\,6$, $\delta t_{ab2} = 13.334\,4$, other data can be obtained similarly. The second whole iteration data are shown in Fig. 6. The second overall iteration time: 79.84 (time), and iteration time costs: 112.15 (time units).

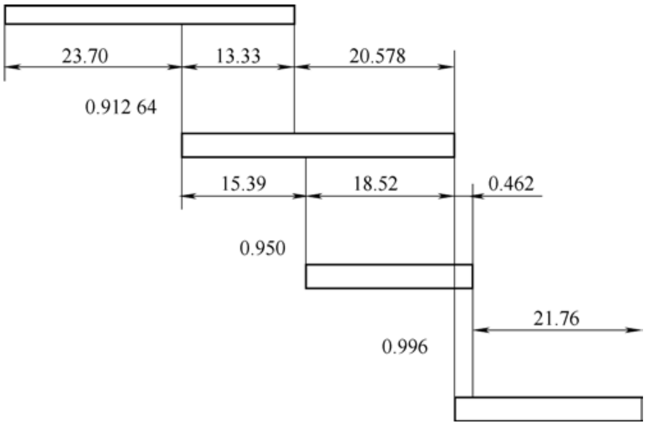

Fig. 6. The second overlapping process data

After two iteration processes, the activities have accumulated the considerable amount of valid information, at the same time the subsystem requires still iterations within the whole system, so we stop the third iteration of the subsystem. The whole iteration time of the subsystem: 125.05+79.84=204.89 (time), and iteration time costs: 267.77 (time units).

On the other hand, if the vehicle upper subsystem is designed in terms of a sequential execution pattern, the first overall iteration time(and time cost): 150 (time unit), the second overall iteration time(and time cost): 111.11 (time unit), and so the whole iteration time(and time cost) of the subsystem: 261.11 (time unit). We can see, the overlapping execution pattern lets the whole iteration time of the subsystem decrease 22%, and the whole iteration time cost increase 3% slightly.

5 Conclusions

Complex product development will inevitably face the design planning of the multi-coupled activities, and overlapped execution of the multi-coupled activities is an important way to reduce the product development time. As the interdependence between activities leads to the insufficiency of the accumulated valid information in overlap process and results overall iteration process, we propose two concepts which are nominal information and valid information, then construct the function of the valid information evolution degree to describe valid information accumulation progress in all the overall iteration processes consisting of the n times iterations, and give VIE degree function solution from rework probability matrix.

By constructing a non-homogeneous Poisson process, we relate the upstream VIE degree function to the probability that the upstream design changes appear, and then to the downstream duration variation. So, we represent the impact on the downstream duration variation from the upstream VIE and design changes, and present development time and cost models of the single overlap.

On this basis, by analyzing overlapping characteristics and overlapping processes of multi coupled activities, multi-segment & multi-ply overlapping planning models of multi coupled activities are built. It is concluded from these models construction analysis: (1) As to multi-segment & multi-ply overlapping of multi coupled activities, the total decrement of the task set' development time is the sum of the time decrement caused by basic overlap segments, and minus the sum of the design time increment caused by multiple overlap segments; (2) the total increment of development cost is the sum of the cost increment caused by all overlap process. Based on the overlap degree analysis of these overlapping planning models, through the four lemmas' concrete proofs, we have proved and obtained two propositions. (1) The multi-ply overlap of the multi coupled activities will weaken the basic overlap effect on the development cycle time reduction. (2) Overlapping the multi coupled activities will decrease product development cycle, but increase product development cost; and there is trade-off between development time and cost. So, blindly increasing the activities' overlap degree is not always a good decision.

At the same time, in order to further shorten the development cycle, two methods are presented to slacken and eliminate multi-ply overlap effects. At last, an example illustrates that using multi-segment and multi-ply overlapping models of multi coupled activities can be helpful to alleviate the dual pressures of product development time and cost.

Future research will introduce multi-information communication within multi coupled activities in once iteration process, and expand further the in-depth study; therefore, more insightful opinions are possibly obtained.

References

[1] UNGER D W, EPPINGER S D. Planning design iterations [EB/OL]. Appears in Innovation in Manufacturing System and Technology. Issue Date: Jan-2002. http://dspace.mit.edu//retrive/4089/IMST016.pdf.

[2] KRISHNAN V, ULRICH K T. Product development decisions: A review of the literature[J]. *Management Science*, 2001, 47(1): 1–21.

[3] CHAKRAVARTH A K. Overlapping design and build cycles in product development[J]. *European Journal of Operational Research*, 2001, 134: 392–424.

[4] HA A Y, PORTEUS E L. Optimal timing of reviews in concurrent design for manufacturability[J]. *Management Science*, 1995, 41(9): 1 431–1 447.

[5] Loch C H, TERWIESCH C. Communication and uncertainty in concurrent engineering[J]. *Management Science*, 1998, 44(8): 1 032–1 048.

[6] KRISHNAN V, EPPINGER S D, WHITNEY D E. A model-based framework to overlap product development activities[J]. *Management Science*, 1997, 43(4): 437–451.

[7] YASSINE A A, CHELST K R, FALKENBURG D R. A decision analytic framework for evaluating concurrent engineering[J]. *IEEE Transaction on Engineering Management*, 1999, 46(2): 144–147.

[8] WANG Zhiliang, ZHANG Youliang. Research on overlapping of coupled activities[J]. *Computer Integrated Manufacturing Systems*, 2006, 6(12): 947–954. (in Chinese)

[9] FARID A, JOHNSOM E, WILL P. Is concurrent engineering always a sensible proposition?[J]. *IEEE Transactions on Engineering Management*, 1995, 42(2): 166–170.

[10] SMITH R P, EPPINGER S D. Deciding between sequential and parallel tasks in engineering design" concurrent engineering[J]. *Research and Applications*, 1998, 6(1): 15–25.

[11] KRISHNAN V, EPPINGER S D. WHITNEY, D E. Accelerating product development by the exchange of preliminary product design information[J]. *Journal of Mechanical Design*, 1995, 117(12): 491–498.

[12] WANG Zheng, YAN Hongsen. Concurrency optimization in product development process[J]. *Computer Integrated Manufacturing Systems*, 2002, 11(8): 852–857. (in Chinese)

[13] CARRASCOSA M, EPPINGER S D, WHINEY D E. Using the design structure matrix to estimate product development time[C]//*Proceeding of DETC'98, 1998 ASME Design Engineering Technical Conferences*, September 13–16, Atlanta, Georgia, USA, 1998: 1–10.

[14] BLACUD N A, BOGUS S M, DIEKMANN J E, et al. Sensitivity of construction activities under design uncertainty[J]. *Journal of Construction Engineering and Management*, 2009, 135(3): 199–206.

[15] WANG Juite, LIN Yung-I. An overlapping process model to assess schedule risk for new product development[J]. *Computers and Industrial Engineering*, 2009, 57(2): 460–474.

[16] XU Duo, YAN Hongsen. Time model and optimization method in concurrent product development process[J]. *Chinese Journal of Mechanical Engineering*, 2006, 42(1): 23–29, 34. (inChinese)

[17] XU Ci-jun, LI Ai-ping, LIU Xue-mei. Time model in overlapped development process of coupled activities[J]. *Computer Integrated Manufacturing Systems*, 2009, 10(15): 1 914–1 920. (in Chinese).

[18] ROEMER T A, AHMADI R. Concurrent crashing and overlapping in product development[J]. *Operations Research*, 2004, 52(4): 606–622

[19] ROEMER T A, AHMADI R, WANG R H. Time-cost trade-offs in overlapped product development[J]. *Operations Research*, 2000, 48(6): 858–865.

[20] KRISHNAN V, EPPINGER S D, WHITNEY D E. Iterative overlapping: accelerating product development by preliminary information exchange[C]//*Proceeding of ASME Design Theory and Methodology Conference*, New York, NY (USA), 1993: 223–231.

[21] ZANGWILL W I, KANTOR P B. Toward a theory of continuous improvement and the learning curve[J]. *Management Science*, 1998, 44(7): 910–920.

[22] HE Yuchen, WANG Xiankui, LIU Chengying, et al. Two application modes for concurrency of concurrent engineering[J]. *Computer Integrated Manufacturing Systems*, 2002, 7(8): 538–541. (in Chinese)

[23] SMITH R P, EPPINGER S D. A predictive model of sequential iteration in engineering design[J]. *Management Science*, 1997, 43(8): 1 104–1 120.

[24] WANG Zhiliang, WANG Yunxia, LU Yun. Primary exploration of optimization theory and method for ordering coupled tasks[J]. *China Mechanical Engineering*, 2011, 22(12): 1 444–1 449. (in Chinese)

[25] WANG Zhiliang, WANG Yunxia, MA Yinzhong. Research on coupled tasks' sequence optimization based on qualitative and quantitative partial order[J]. *Applied Mechanics and Materials*, 2012, 121–126: 4 601–4 605.

Biographical notes

WANG Zhiliang, born in 1965, is currently an associate professor at *Nanjing Institute of Technology, China*. He received his PhD degree from *Nanjing University of Science and Technology, China*,

in 2005. His main research interests include decision-making, manufacturing Automation, industrial engineering.
Tel: +86-25-86118256; E-mail: wwangzzll@njit.edu.cn

WANG Yunxia, born in 1976, is currently an associate professor at *Nanjing Institute of Technology, China*. She received her PhD degree from *Southeast University, China*, in 2005. His main research interests include advanced manufacturing, numerical control, industrial engineering.
Tel: +86-25-86118256; E-mail: wang-yunxial@njit.edu.cn

QIU Shenghai, born in 1967, is currently an associate professor at *Nanjing Institute of Technology, China*. He is currently a PhD candidate at *Nanjing University of Aeronautics and Astronautics, China*. His research interests include advanced manufacturing, management information system, and industrial engineering.
Tel: +86-25-86118256; E-mail: qiush2000@njit.edu.cn

Appendix 1

LEMMA 1. Increasing overlap degree of basic overlap process will lead product development cycle decrease, viz.

$$\frac{\partial T_{\mathrm{s}}}{\partial\, t_{abi}} > 0,\ \frac{\partial T_{\mathrm{s}}}{\partial\, t_{bci}} > 0.$$

Proof. As $p_{abi} = P(X=0)|_{t_{abi}}$, from Eqs. (4) and (5), we can obtain

$$p_{abi} = \beta_{ab,i-1} + (\beta_{abi} - \beta_{ab,i-1}) \times \left(\frac{t_{abi}}{D_{ai}}\right)^m = \beta_{ab,i-1} + h_{abi} \times \left(\frac{t_{abi}}{D_{ai}}\right)^m,$$

$$h_{abi} = \beta_{abi} - \beta_{ab,i-1},$$

$$\frac{\partial T_{\mathrm{s}}}{\partial t_{abi}} = -[(1+p_{abi}) \times \delta t_{abi}]'\Big|_{t_{abi}} = -\left[(1+\beta_{ab,i-1} + h_{ab,i} \times \left(\frac{t_{abi}}{D_{ai}}\right)^m \times (D_{ai} - t_{abi})\right]'\Bigg|_{t_{abi}} = -\frac{0.5}{D_{ai}^m} \times [h_{ab,i} D_{ai} \times m \times t_{ab,i}^{m-1} - h_{ab,i}(m+1)t_{ab,i}^m - (1+\beta_{ab,i-1})D_{ai}^m].$$

Further,

$$\frac{\partial^2 T_s}{\partial t_{abi}^2} = -\frac{0.5}{D_{ai}^m}[h_{ab,i} D_{ai} \times m(m-1) \times t_{ab,i}^{m-2} - h_{ab,i} \times (m+1)m \times t_{ab,i}^{m-1}].$$

Let $\partial^2 T_{\mathrm{s}} / \partial t_{abi}^2 = 0$, we can obtain the extreme points of $\partial T_{\mathrm{s}} / \partial t_{abi}$, $t_{abi} = 0$, and $t_{abi} = D_{ai} \times (m-1)/(m+1)$.

If $0 < m < 1$, the extreme points of $\partial T_{\mathrm{s}} / \partial t_{abi}$ is at endpoint, they are $t_{abi} = 0$ and $t_{abi} = D_{ai}$,

$$\left.\frac{\partial T_{\mathrm{s}}}{\partial t_{abi}}\right|_{t=0} = 0.5(1+\beta_{ab,i-1}) > 0,$$

$$\left.\frac{\partial T_{\mathrm{s}}}{\partial t_{abi}}\right|_{t=D} = 0.5(1+\beta_{ab,i}) > 0.$$

So, while $0 < m < 1$, $\partial T_{\mathrm{s}} / \partial t_{abi} > 0$.

On the other hand, if $m \geqslant 1$, let

$$t_{abi} = \frac{m-1}{m+1} \times D_{ai} \times \lambda,$$

then

$$\frac{\partial^2 T_{\mathrm{s}}}{\partial t_{abi}^2} = -\frac{1}{2D_{ai}} \times h_{ab,i} \times \left(\frac{m-1}{m+1} \times \lambda\right)^{m-2} \times m(m-1)(1-\lambda).$$

While $\lambda > 1$, $\partial^2 T_{\mathrm{s}} / \partial t_{abi}^2 > 0$; $\lambda < 1$, $\partial^2 T_{\mathrm{s}} / \partial t_{abi}^2 < 0$. So $t_{abi} = D_{ai} \times (m-1)/(m+1)$ is the minimum point, and

$$\left.\frac{\partial T_s}{\partial t_{abi}}\right|_{t=\frac{m-1}{m+1}\times D} = -\frac{1}{2}\left[h_{abi} \times \left(\frac{m-1}{m+1}\right)^{m-1} - (1+\beta_{ab,i-1})\right].$$

As maximum of $[(m-1)/(m+1)]^{m-1}$ is 1, while $m \geqslant 1$,

$$\left.\frac{\partial T_{\mathrm{s}}}{\partial t_{abi}}\right|_{t=\frac{m-1}{m+1}\times D} \geqslant -\frac{1}{2} \times (h_{abi} - (1+\beta_{ab,i-1})) > 0.$$

According to the above derivation, while the speed ratio m is any value, $\partial T_{\mathrm{s}} / \partial t_{abi} > 0$.

Similarly, we can prove that $\partial T_{\mathrm{s}} / \partial t_{bci} > 0$.

Appendix 2

LEMMA 2. Increasing overlap degree of the multiple overlap process will lead product development cycle increase, viz. $\partial T_{\mathrm{s}} / \partial t_{aci} < 0$.

Proof. Let $h_{aci} = \beta_{aci} - \beta_{ac,i-1}$,

$$\frac{\partial T_{\mathrm{s}}}{\partial t_{aci}} = [(1-p_{aci}) \times \delta t_{aci}]'\Big|_{t_{aci}} = \left[(1-\beta_{ac,i-1} - h_{ac,i} \times \left(\frac{t_{aci}}{D_{ai}}\right)^m \times (D_{ai} - t_{aci})\right]'\Bigg|_{t_{aci}} = \frac{0.5}{D_{ai}^m} \times [-h_{ac,i} D_{ai} \times m \times t_{ac,i}^{m-1} + h_{ac,i}(m+1)t_{ac,i}^m - (1+\beta_{ac,i-1})D_{ai}^m],$$

$$\frac{\partial^2 T_{\mathrm{s}}}{\partial t_{aci}^2} = -\frac{0.5}{D_{ai}^m}[-h_{ac,i} D_{ai} \times m(m-1) \times t_{ac,i}^{m-2} + h_{ac,i}(m+1)m \times t_{ac,i}^{m-1}].$$

Let $\partial^2 T_{\mathrm{s}} / \partial t_{aci}^2 = 0$, we can obtain the extreme points of

$\partial T_s / \partial t_{aci}$, $t_{aci} = 0$, and $t_{aci} = D_{ai} \times (m-1)/(m+1)$.

If $0 < m < 1$, the extreme points of $\partial T_s / \partial t_{aci}$ is at the endpoint, they are $t_{aci} = 0$ and $t_{aci} = D_{ai}$,

$$\left.\frac{\partial T_s}{\partial t_{aci}}\right|_{t=0} = -0.5(1-\beta_{ac,i-1}) < 0,$$

$$\left.\frac{\partial T_s}{\partial t_{aci}}\right|_{t=D} = 0.5(-1+\beta_{ac,i}) < 0.$$

So, while $0 < m < 1$, $\partial T_s / \partial t_{aci} < 0$.

On the other hand, if $m \geqslant 1$, let

$$t_{aci} = \frac{m-1}{m+1} \times D_{ai} \times \lambda,$$

then

$$\frac{\partial^2 T_s}{\partial t_{aci}^2} = -\frac{1}{2D_{ai}} \times h_{ac,i} \times \left(\frac{m-1}{m+1} \times \lambda\right)^{m-2} \times m(m-1) \times (-1+\lambda).$$

While $\lambda > 1$, $\partial^2 T_s / \partial t_{aci}^2 > 0$; $\lambda < 1$, $\partial^2 T_s / \partial t_{aci}^2 < 0$. So $t_{aci} = D_{ai} \times (m-1)/(m+1)$ is the minimum point, and

$$\left.\frac{\partial T_s}{\partial t_{aci}}\right|_{t=\frac{m-1}{m+1}\times D} = -\frac{1}{2}\left[h_{aci} \times \left(\frac{m-1}{m+1}\right)^{m-1} + (1-\beta_{ac,i-1})\right].$$

As minimum of $[(m-1)/(m+1)]^{m-1}$ approach to 0,

$$\left.\frac{\partial T_s}{\partial t_{aci}}\right|_{t=\frac{m-1}{m+1}\times D} \leqslant -\frac{1}{2}[h_{aci} \times 0 + (1-\beta_{ab,i-1})] < 0.$$

Moreover, at the endpoint, $t_{aci} = 0$ and $t_{aci} = D_{ai}$, $\partial T_s / \partial t_{aci} < 0$. We can gain, while $m \geqslant 1$, $\partial T_s / \partial t_{aci} < 0$.

According to the above derivation, while the speed ratio m is any value, $\partial T_s / \partial t_{aci} < 0$.

So, we have proved the following.

Proposition 1. The multi-ply overlap of the multi coupled activities will weaken the basic overlap effect on the development cycle time reduction.

Appendix 3

LEMMA 3. Overlapping the multi coupled activities will decrease product development cycle.

Proof. From Eq. (13), $(1+p_{abi}) \times 0.5\delta t_{abi}$ represents the design time decrement caused by basic overlap process *A–B* and $(1-p_{aci}) \times 0.5\delta t_{aci}$ represents the design time increment caused by multiple overlap process *A–C*.

While the minimum of $(1+p_{abi}) \times 0.5\delta t_{abi}$ is $0.5\delta t_{abi}$, the maximum of $(1-p_{aci}) \times 0.5\delta t_{aci}$ is $0.5\delta t_{aci}$, but according to overlapping sequence, $\delta t_{abi} \geqslant \delta t_{aci}$. So, from Eq. (13), $-(1+p_{abi}) \times 0.5\delta t_{abi} + (1-p_{aci}) \times 0.5\delta t_{aci}$ is less than zero. Therefore, overlapping the multi coupled activities will decrease product development cycle.

LEMMA 4. Increasing overlap degree of any type of overlap will lead product development cost increase.

$$\frac{\partial C_s}{\partial t_{abi}} < 0, \ \frac{\partial C_s}{\partial t_{bci}} < 0, \ \frac{\partial C_s}{\partial t_{aci}} < 0.$$

Proof. Form Eq. (14),

$$\frac{\partial C_s}{\partial t_{aci}} = [(1-p_{aci}) \times \delta t_{aci}]'\Big|_{t_{aci}}.$$

So, similar as Appendix 2, we can prove that $\partial C_s / \partial t_{aci} < 0$.

Similarly, we can also prove $\partial C_s / \partial t_{abi} < 0$, $\partial C_s / \partial t_{bci} < 0$.

According to LEMMA 3 and LEMMA 4, we have proved the following.

Proposition 2: Overlapping the multi coupled activities will decrease product development cycle, but increase product development cost. And there is trade-off between development time and cost.