Estimation of Project Completion Time and Factors Analysis for Concurrent Engineering Project Management: A Simulation Approach

Enzhen Huang, Shi-Jie (gary) Chen

CE CONCURRENT ENGINEERING: Research and Applications

Estimation of Project Completion Time and Factors Analysis for Concurrent Engineering Project Management: A Simulation Approach

Enzhen Huang and Shi-Jie (Gary) Chen*

Department of Mechanical and Industrial Engineering, Montana State University
Bozeman, MT 59717, USA

Abstract: In concurrent engineering projects, tasks are usually interdependent among each other that require much iteration before completion where the critical path method/program evaluation and review technique (CPM/PERT) may not be applicable to help estimate the project duration. In addition, carrying out a large-scaled project in a dynamic environment has to deal with various factors at the same time. When estimating the project completion time, previous research often focused on one subject of interest and assumed the other factors causing little effect on the overall project duration. The objective of this article is to develop a research framework to help estimate the project completion time and analyze the major factors that affect the estimation for complex concurrent engineering projects. The framework consists of three major components: (1) data collection, where the needed data for simulation are prepared including project task structure, task relations, and quantified team member characteristics; (2) simulation, where tasks are dynamically assigned to the appropriate members/engineers according to each member's knowledge level of the task, teamwork capability, work schedule availability, and learning curve improvement; and (3) data analysis, where significant factors to the project completion time are studied by the ANOVA analysis based on the simulation results. The effectiveness of our framework and the simulation model is demonstrated by an illustrative example.

Key Words: project management, concurrent engineering, design structure matrix, simulation.

1. Introduction

Carrying out a large-scaled project in a dynamic environment has to deal with various factors at the same time. Project managers always find it difficult to estimate the project duration because of the unexpected variances and the limit of resources. Traditional CPM and PERT methods have been widely used in simulations to estimate the project duration. However, the validity of CPM/PERT has been questioned in the highly concurrent project environment [1], which for example, includes the cases in new product design, software development and supply chain interactions, etc. In concurrent engineering projects, tasks are usually interdependent among each other, which require a group of people with different engineering backgrounds to work together. The interdependent task group often results in task rework or iteration where CPM/PERT may not be applicable to help estimate the project duration. One of the popular tools that is able to analyze the rework of tasks in projects is the Design Structure Matrix (DSM). DSM was first introduced by Steward [2] to analyze the engineering design process. It is a square matrix with n rows and columns, and m non-zero elements, where n is the number of nodes, tasks or system elements and m is the number of edges or links of dependencies in the network of the system. If there is an edge from node i to node j, the value of element ij is a unity or a marked sign in the matrix, otherwise the value of the element is zero or empty. When each nonzero element in DSM is replaced by a numerical value (ranging from 0 to 1) to indicate the strength of task interaction, it is called a numerical DSM. In recent years, a DSM has been used as a management aid as well as an engineering tool to guide the organizational structure of design projects [3–14].

In complex project management, it is desirable to make the best use of available human resources to improve the efficiency of project execution. Complex projects often contain interdependent task groups that require a group of members with different characteristics and engineering background to work together in a team [15]. Chen [6] proposed a methodology for task-member assignment for concurrent engineering project management. The research suggested that in order to improve the team efficiency, each member's functional knowledge, their teamwork capability and working relationship should be understood and incorporated in the task-member assignment model. Moreover, project managers also have to consider the availability of each member's schedule in practice because a member may be involved

*Author to whom correspondence should be addressed.
E-mail: gchen@ie.montana.edu
Figure 3 appears in color online: http://cer.sagepub.com

1063-293X/06/04 0329–13 $10.00/0 DOI: 10.1177/1063293X06072482

in working for more than one project at the same time. Therefore, the right members will be assigned to the right task at the right time for complex concurrent engineering projects.

During the progress of tasks, members can improve their knowledge level in the area of the task that they have been working on. Wrights [16] first discovered this fact in manufacturing assembly lines and defined it as a 'learning curve'. Based on observations, Wrights developed a logarithmic function to calculate the learning curve, which was later called Wrights' Law. The function works well for simple repetitive tasks and operational type of tasks. For the knowledge-based tasks, Hanakawa et al. [17] studied the learning curve in software development and developed a mathematical model of the learning curve. The model integrates the specifications of tasks, member's knowledge, and other member's characteristics to determine a member's productivity on a task. A member can improve his/her productivity to finish a task because of the learning curve improvement; therefore the duration of the entire project is reduced.

When estimating the project completion time in project management, previous research often focused on one subject of interest and assumed the other factors causing little effects on the overall project duration. Our research in this article with the development of a research framework considers several major variations in the lead-time estimation for complex project management in a concurrent engineering environment. With the proposed simulation model in the framework, not only is the project duration estimated, solutions of task-member assignments, which are useful for managers, will also be suggested. The simulation model studies five major sources that contribute to lead-time variations: (1) task rework probability; (2) task rework impact value; (3) availability of member schedules; (4) learning curve efficiency; and (5) task-member assignment options. With the result of simulation, an analysis of variance (ANOVA) is performed to analyze the data and identify the significant effect of each source of variance.

2. The Research Framework

The research framework of this article contains the following three major components:

(1) *Data collection*: This component aims to prepare all the needed data for the simulation model. Some data are directly collected by interviewing the project managers/experts (i.e., member schedules and characteristics). The other data can be derived from mathematical models (i.e., task clustering).
(2) *Simulation*: This is the core driver in the framework. The simulation model dynamically assigns tasks to the appropriate members/engineers subject to the resource constraints and the project task structure. The project completion time is estimated as a result of the simulation.
(3) *Data analysis*: The last component is to analyze the simulation results and to identify the important factors that affect the project completion time. The simulation experiment is performed in multiple runs with different factor levels. ANOVA is used to test the significance of the factors and their interactions. Conclusions and recommendations for estimating the completion time of complex projects will be given based on the outcomes from the analysis of ANOVA.

2.1 Task Clustering

The purpose of task clustering is to decompose the large interdependent task groups identified by DSM into smaller and manageable sizes. A preliminary step for clustering is to apply Steward's partitioning algorithm to reveal three basic task types in projects (i.e., independent, dependent, and interdependent tasks) [2]. Although a partitioning algorithm can help identify the interdependencies among tasks, the size of interdependent task groups is often large in a complex project. Research has concluded that the effectiveness of communication depends on the number of communication links among the related tasks or system elements. Therefore an effective and efficient communication will be difficult to achieve as the size of the interdependent task group increases [18–22]. Chen and Lin [4,5] commented that the large interdependent task groups usually make it difficult for task coordination and team organization and thus delay the project completion. The authors developed a model to decompose the large interdependent task group into smaller and manageable subgroups based on numerical DSM and clustering technique. This decomposition model contains the following three steps:

(1) *Symmetrical task interaction matrix*: DSM is a matrix that offers the information of task dependency for their 'from–to' descriptions. By assuming that the input and output connections carry the same weight, the amount of interaction is calculated by averaging each pair of symmetrical elements in a numerical DSM, because the interaction of any two tasks contains both information input and output connections. This symmetrical task interaction matrix is expressed mathematically (for each pair of row *i* and column *j*) as: $\mathrm{SymDSM}_{\mathrm{i,j}} = (\mathrm{NumericalDSM}_{i,j} + \mathrm{NumericalDSM}_{j,i})/2$.
(2) *Decomposition of a large interdependent group*: A large interdependent group is decomposed into

smaller subgroups using a clustering technique. The key is to calculate the distance measures for the matrix. The quantified interaction strengths in the symmetrical matrix, $\text{SymDSM}_{i,j}$, are used to calculate the distance measures using Squared Euclidean Distance, which is able to handle both binary and numerical measures and is appropriate for numerical DSM. When clustering the elements, any two elements with the lowest distance measure are first grouped together before those elements with higher distance measures. Using a robust approach, the average-linkage method, clusters are formed by evaluating the interactions between all elements rather than only each pair of elements (i.e., the case with the single-linkage method). This method is robust to outliers, hence small changes of coupling values in the matrix do not affect the clustering results.

(3) *Clustering performance evaluation by numerical interaction density (NDd):* For an $n \times n$ matrix, there are $n-1$ possible clustering results. To select the best solution from all possible clustering results, a performance measure is needed to evaluate the clustering performance from each result and determine the final groups. Chen and Lin developed a performance measure, NDd, to help select the best clustering result [5]. NDd, measuring the numerical interaction strengths outside the block diagonal of the clustered matrix, is formulated as NDd = Ne/Outer-Cells. Ne is the total coupling strengths outside the block diagonal of the clustered matrix. Outer-Cells is the total number of cells outside the block diagonal of the clustered matrix. The best task clustering is the one with the lowest NDd value.

2.2 Task-member Assignment

The task-member assignment model in our framework is based on Chen's research [6]. In order to ensure a successful team composition, it is important to carefully choose the team members with desirable qualifications. Chen and Lin developed a team member model for three important team member characteristics with quantitative representations (i.e., multifunctional knowledge rating, teamwork capability rating, and working relationship rating) [23]. The first is to represent the multifunctional knowledge of team members. A member who does not work in a certain functional department may still have a certain level of knowledge about that department. This will increase the flexibility when a key functional member is needed during the team organization. Second, to build a successful team, a teamwork capability of team members is needed by taking their experience, communication skills, and flexibility in job assignment into account. Third, since team members work closely, their collegiality directly affects team performance regardless of their knowledge and teamwork capability. Thus working relationship between team members should not be ignored. For effective human resource management, more and more companies maintain their employees' performance, knowledge, skills, experience, interests, and relevant personal characteristics in a computerized data bank, called a skill inventory or a knowledge bank. This knowledge bank can serve as a useful data source to help managers assess a member's multifunctional knowledge, teamwork capability, and working relationship with others.

After task portioning [2] and task clustering [4,5], the entire project task structure in DSM is coordinated into two types: single-task and multiple-task. The focus of task-member assignments will be on these two types following the task sequence identified in DSM from top to bottom. The assumptions are: (1) each task belongs to one functional department and needs one member only; and (2) each member can be assigned to multiple tasks. With quantitative measures of team member characteristics and task workloads, the goal of assigning the right team members to the right tasks will be carried out by the task-member assignment model [6].

For the need of assigning multiple tasks to one member, a workload factor (W) to each task indicating the percentage of a member's capacity required to complete the task is introduced. In an n-task project, the workload W_i of each task X_i is defined as: $W_i = c_i/w$ $(i = 1, 2, \ldots, n)$ where $w \in \{1, 2, 3, \ldots\}$ and $c_i \in \{1, 2, \ldots, w\}$. When $w = 2$, for example, the possible task workload of each W_i in the project is either 50% (1/2) or 100% (2/2). A task with 50% workload means this task requires 50% of a member's capacity. Therefore, once a member is assigned to a 50% workload task, the remaining 50% of his/her capacity is still available for the other assignments. In addition, a task with a 100% workload needs 100% of a member's capacity to finish the task. So, any member being assigned to such a 100% workload task cannot be available for any other task and has to devote his/her full time to complete the task. Based on the definition of task workload factor (W), each workload W_i of task X_i can only be any one of $\{1/w, 2/w, 3/w, \ldots, w/w\}$ where w is a fixed integer for the entire project. This is to avoid the conflict that may happen during the task-member assignments. For example, if w is given by two different integers such as 2 and 3, the possible task workload of each W_i in the project can be any one of $\{1/2, 2/2, 1/3, 2/3, 3/3\}$. After being assigned to a 50% workload task (1/2), a new selected member will have 50% capacity left. If we assign the same member to another task with 33.3% workload (1/3), although this member still has 16.7% capacity (1/6) left, he/she is no longer available for any other task assignments. The 16.7% capacity of this member will be wasteful to the company. In terms of task workload, members can be assigned to multiple tasks at the same time and will devote a percentage of their time to each assigned task X_i with respect to the task workload W_i.

For the single-task assignment, no teamwork capability and working relationship between team members are considered since only one member is needed. A member with the highest knowledge rating to that task's functional requirement and with the lowest teamwork capability will be selected, since single tasks do not require cooperation and team effort between members. For a project with n tasks in a company that has p departments, the procedure for a single-task assignment is:

(1) Determine a functional department D_k the current task X_i belongs to and the workload W_i of this task, where $k \in \{1, 2, \ldots, p\}$ and $i \in \{1, 2, \ldots, n\}$.
(2) Select all members whose capacity is equal or higher than the current task's workload W_i and whose knowledge rating for the functional department D_k is above a chosen threshold (R_s^*) determined by the project managers and put them in candidate group C_1.
(3) Of all members in C_1, choose the one with the lowest teamwork capability rating. If there is a tie with more than one member chosen, go to Step 4. Otherwise, go to Step 6.
(4) Put those tied members in group C_2.
(5) Of all members in C_2, choose the member with the highest knowledge rating. If tie, choose randomly.
(6) The single-task assignment for task X_i is complete.
(7) This procedure is repeated for any single-task identified in the DSM matrix.

For the multiple-task assignment, in order to increase team performance, it has to consider teamwork capability and working relationship between team members in addition to their knowledge ratings. After choosing the appropriate candidate members with respect to the task workload (W) and the knowledge rating, a mathematical model is formulated. The objective of the mathematical model is to optimize the task-member assignments in terms of teamwork capability and working relationship ratings. For a project with n tasks in a company that has p departments, the procedure for multiple-task assignments is:

(1) Determine which functional department D_k that each task X_i belongs to and the workload W_i of each task, where $k \in \{1, 2, \ldots, p\}$ and $i \in \{1, 2, \ldots, n\}$.
(2) M functional departments are determined in Step 1, namely $\{\mathrm{D}_1, \mathrm{D}_2, \ldots, \mathrm{D}_m\}$ corresponding to m tasks in a multiple-task group.
(3) Let $j = 1$.
(4) Select all members whose capacity is equal to or higher than the current task's workload W_i and whose knowledge rating for the functional department D_j is above a chosen threshold (R_m^*) determined by the project managers and put these members in the candidate group C_j.
(5) If $j < m$, let $j = j + 1$ and go to Step 4, otherwise, go to Step 6.
(6) To select the right team member from each corresponding candidate group $\{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_m\}$, we formulate a mathematical model as follows:

$$\begin{aligned} &Max. \quad \sum_{a=1}^{m} \sum_{b=a+1}^{m} \sum_{\forall i} \sum_{\forall j} w_{ai,bj} x_{ai} x_{bj} \\ &ST \quad \sum_{\forall i} x_{ai} = 1 \quad (a = 1, 2, \ldots n) \\ &\forall x_{ai}, x_{bj} = 0 \text{ or } 1 \end{aligned} \tag{1}$$

where

$$w_{ai,bj} = \alpha(T_{ai} + T_{bj}) + (1 - \alpha) R_{ai,bj} \quad (0 \le \alpha \le 1)$$

is the descriptor of teamwork capability and working relationship of the i-th candidate member in group a and j-th candidate member in group b

T_{ai} = teamwork capability rating of the i-th candidate member in group a
T_{bj} = teamwork capability rating of the j-th candidate member in group b
$R_{ai,bj}$ = working relationship rating between the i-th candidate member in group a and j-th candidate member in group b
x_{ai} = the i-th candidate member in group a
x_{bj} = the j-th candidate member in group b
m = number of candidate groups

(7) This procedure is repeated for any multiple-task group identified in the DSM matrix.

Each candidate member x_{ai} in the a-th group is either to be selected as a team member ($x_{ai} = 1$) or not to be selected ($x_{ai} = 0$). First, a descriptor $w_{ai,bj}$ is used for representing teamwork capability and working relationship between a candidate member x_{ai} from the a-th group and a candidate member x_{bj} from the b-th group. To decide the relative contribution of teamwork capability rating and working relationship rating to $w_{ai,bj}$, a weight α is used. For instance, if $\alpha = 0.5$, total teamwork capability rating of the two members, T_{ai} and T_{bj}, and working relationship $R_{ai,bj}$ will contribute equally to $w_{ai,bj}$. The project managers are responsible for determining the value of α. The choice of α depends on the relative importance of the two factors (teamwork capability and working relationship) to the decision makers. The objective function will assign team members recursively by comparing all possible team compositions for every pair of members from different candidate groups. Finally, the best team is formed with the maximum objective function value.

2.3 Learning Curve Improvement

In order to cope with the real situation, each member's learning curve improvement over the performing stage of the project is considered because: (1) members can often gain experience and knowledge in the area of the task that they are working on; (2) the improvement of the experience and knowledge will enhance each member's ability to complete the assigned task in a shorter time; (3) it is expected that the longer the duration of a project is, the higher effect the learning curve improvement has on the total project time; and (4) how much a member will be able to learn by performing a task is different from person to person. Therefore it is necessary to consider each member's learning efficiency in the simulation model. Hanakawa's learning curve equation is used in the simulation [17]:

$$L_{ij}(\theta) = \begin{cases} K_{ij}e^{-E_{ij}(\theta - b_{ij})} & (b_{ij} \leq \theta) \\ 0 & (b_{ij} > \theta) \end{cases} \tag{2}$$

where:

$L_{ij}(\theta) =$ the quantity of gain to knowledge of member i executing task j, which has a knowledge level θ.

$b_{ij} =$ member i's knowledge level about task j.

$K_{ij} =$ the maximum quantity of gain to knowledge of member i by executing task j.

$E_{ij} =$ member i's efficiency of gain to knowledge by executing task j.

$\theta =$ the required knowledge level to execute task j.

3. The Simulation Model

The simulation model for estimating the project completion time is shown in Figure 1. The solid arrows indicate the flows of simulation and the dashed lines represent the required information or data inputs in the simulation. The entire project lifecycle is divided into multiple time units (e.g., in days). The time index is a counter to update and record the project duration. In each time unit, the first thing is to examine whether there are any tasks that require rework. When rework is needed, the task is usually not restarted from scratch. Only a portion of the adjusted task will be reworked, which varies from task to task depending on the values of 'rework probability' and 'rework impact'. Browning and Eppinger proposed 'rework probability' and 'rework impact' in their DSM study to analyze and control the task rework in simulation [24,25]. The rework probability is the probability for a specific task to make adjustment or rework when it receives the feedback information from another related task. The rework impact value describes the amount of work in the task to be reworked.

Task clustering and task-member assignment are described in sections 2.1 and 2.2, respectively. The clustered DSM shows a well-organized project task structure and prepares the interdependent task groups with manageable sizes. Three important team member characteristics (i.e., multifunctional knowledge, teamwork capability, and working relationship) are quantified. Using the task-member assignment model, assigning the right tasks to the right members is achieved.

In practice, there are also some other important constraints to consider in the task-member assignment such as member schedule, knowledge level threshold, and task-member assignment options. First, it is often seen that a member is involved in multiple project tasks at the same time. Before starting each project, project managers may want to consider each task's workload and the availability in each member's work schedule, so that the members will not be overloaded by the assigned tasks and each task may not wait too long for an available member. Second, in a complex project, project managers usually do not assign an inexperienced member to a task, which is totally new to this member. A threshold of knowledge level can be used to cut the unqualified members out of consideration and only the members with the qualified knowledge level are to be kept in the candidate pool. Third, it is also common practice for project managers to prefer to let a task be handled and completed by the same member. However, if this member's schedule is tight, project managers may consider allowing more than one member to take turns and be in charge of the task. Such kind of member rotations will require a transition time, which is the time needed for a member to be familiar with the task progress in order to know what has been done so far if he/she is assigned to continue a partially completed task left by a previous member.

Task progression is simulated by a stochastic process. The estimated duration of each task is based on the triangular distribution. The pessimistic, average and optimistic times to complete a single task are obtained by interviewing the project managers and the functional experts. As the task is proceeding, the member's learning curve for the task he/she is working on continues to improve. Using the learning curve model described in section 2.3, the simulation records the member's learning improvement over each time unit. When each task is finished, the simulation model checks whether the task is inter-related to the other tasks that will require rework. If so, the rework adjustment will be noted down. At the end of each time unit, the simulation

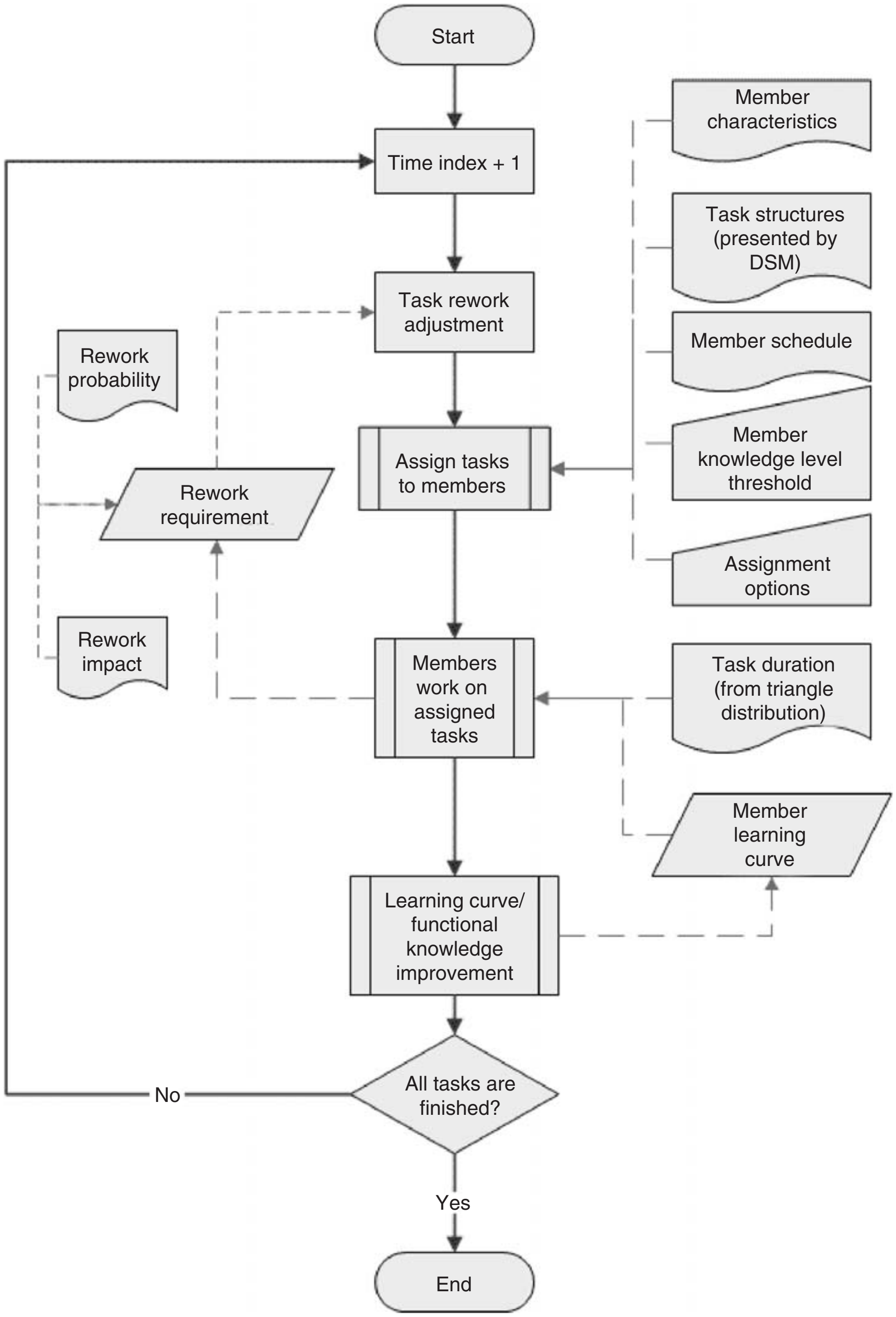
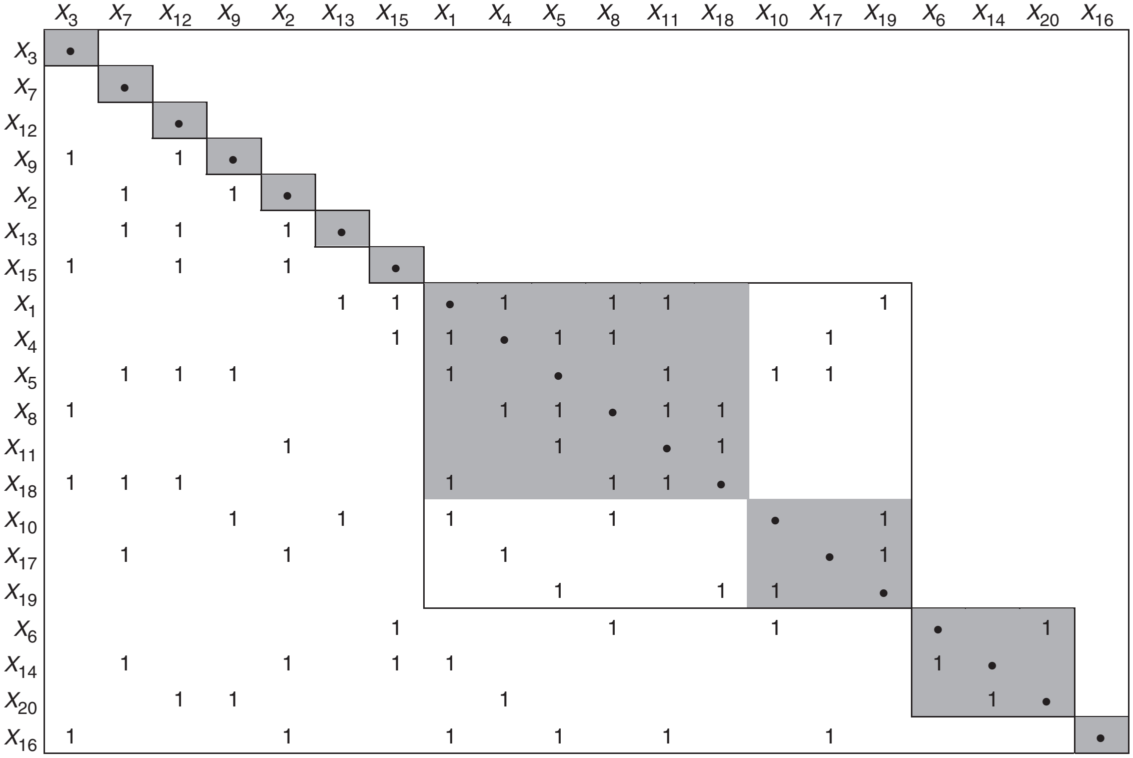


Figure 1. The simulation model.

checks whether all the tasks are finished without further rework. If so, the project is completed and the simulation run is ended, otherwise the simulation goes to the next time period in the loop and continues to run.

4. Analysis of Results

The objective of this research is not only to estimate the project completion time, but also to analyze the major factors that affect the estimation. The following five major sources that contribute to lead-time variations are studied: (1) task rework probability; (2) task rework impact value; (3) availability of member schedules; (4) learning curve efficiency; and (5) task-member assignment options. Factors 1 and 2 represent the coupling strengths or the extent of interdependencies among the tasks. The higher the rework probabilities and rework impacts are, the stronger the tasks are interdependent to each other. Factor 3 is about the



	X_{3}	X_{7}	X_{12}	X_{9}	X_{2}	X_{13}	X_{15}	X_{1}	X_{4}	X_{5}	X_{8}	X_{11}	X_{18}	X_{10}	X_{17}	X_{19}	X_{6}	X_{14}	X_{20}	X_{16}
X_{3}	•																			
X_{7}		•																		
X_{12}			•																	
X_{9}	1		1	•																
X_{2}		1		1	•															
X_{13}		1	1		1	•														
X_{15}	1		1		1		•													
X_{1}						1	1	•	1		1	1				1				
X_{4}							1	1	•	1	1				1					
X_{5}		1	1	1				1		•		1		1	1					
X_{8}	1								1	1	•	1	1							
X_{11}					1					1		•	1							
X_{18}	1	1	1					1			1	1	•							
X_{10}				1		1		1			1			•		1				
X_{17}		1			1				1						•	1				
X_{19}										1			1	1		•				
X_{6}							1				1			1			•		1	
X_{14}		1			1		1	1									1	•		
X_{20}			1	1					1									1	•	
X_{16}	1				1			1		1		1			1					•

Figure 2. The coordinated 20-task DSM.

resource constraint as indicated by the member's working hours. Factor 4 studies the member's learning characteristics and factor 5 is a management decision between two assignment options: 'completing a task by the same member' versus 'completing a task by rotating different available members'. An experiment is designed based on these five factors and an analysis of variance (ANOVA) is used to identify the significance of the factors. Finally, conclusions and recommendations will be drawn from the ANOVA analysis to help the managers with their decision-making.

5. An Illustrative Example

To demonstrate the effectiveness of the research framework, a project with 20 tasks and a company with 30 members in seven functional departments are used. The project is an engineering design project with 20 tasks: $X_{1=}$operating structure design, $X_2=$vessel design, $X_3=$plant layout/general arrangement, $X_4=$shipping design, $X_5=$structure lifting design, $X_6=$pressure drop analysis, $X_7=$process engineering, $X_8=$structural documentation, $X_9=$size valves, $X_{10}=$wind load design, $X_{11}=$seismic design, $X_{12}=$piping design, $X_{13}=$process and instrumentation diagram, $X_{14}=$equipment support, $X_{15}=$pipe flexibility analysis, $X_{16}=$design documentation, $X_{17}=$foundation load design, $X_{18}=$insulation structural design, $X_{19}=$structural bill of materials, and $X_{20}=$assembly design. The 30 members from each of the seven functional departments are: $D_1=\{M_1, M_2, M_3, M_4, M_5\}$; $D_2=\{M_6, M_7, M_8, M_9\}$; $D_3=\{M_{10}, M_{11}, M_{12}, M_{13}, M_{14}\}$; $D_4=\{M_{15}, M_{16}, M_{17}, M_{18}\}$; $D_5=\{M_{19}, M_{20}, M_{21}, M_{22}\}$; $D_6=\{M_{23}, M_{24}, M_{25}, M_{26}, M_{27}\}$; $D_7=\{M_{28}, M_{29}, M_{30}\}$.

According to Steward's partitioning algorithm, Figure 2 shows the partitioned DSM for the 20-task engineering design project from which not only the sequence of 20 tasks is identified, the entire structure of the design process is also revealed. Tasks X_3 and X_7 are independent tasks so that they can be performed concurrently. Tasks $\{X_{12}, X_9, X_2, X_{13}, X_{15}\}$ are a set of dependent tasks, so they can be carried out sequentially. The two interdependent task groups found in the matrix are $\{X_1, X_4, X_5, X_8, X_{10}, X_{11}, X_{17}, X_{18}, X_{19}\}$ and $\{X_6, X_{14}, X_{20}\}$. Due to various functional requirements among the interrelated tasks, multifunctional teams are needed for these two interdependent groups. However, team performance is usually degraded when team size is large. The large interdependent task group (i.e., the 9-task interdependent group), therefore, has to be decomposed into smaller and manageable sizes. According to the decomposition model developed by Chen and Lin [4,5], the 9-task interdependent group $\{X_1, X_4, X_5, X_8, X_{10}, X_{11}, X_{17}, X_{18}, X_{19}\}$ in Figure 2 is clustered into two smaller groups: $\{X_1, X_4, X_5, X_8, X_{11}, X_{18}\}$ and $\{X_{10}, X_{17}, X_{19}\}$, which are shown by the shaded blocks. The task structure shown in Figure 2 lays a sound foundation for team organization where each team is limited in a manageable size so team members can work closely and more efficiently.

The task-member assignment model shown in section 2.2 is implemented in our simulation model to help form the best team composition. As each task is assigned to a right member, the member's learning curve improvement with the task can be calculated by Equation (2) in section 2.3. This learning curve equation shows that if a member's knowledge level to the task is above the required level, he/she will not gain any knowledge improvement. The learning curve efficiency E_{ij}, which is a number between 0 and 1, indicates a member i's efficiency in gaining knowledge by executing task j. The higher the E_{ij} value is, the slower the member i is able to improve knowledge by executing task j. In this example, the required knowledge level θ for every task is set to 0.9. The maximum quantity of knowledge gained by member i by executing task j is assumed as the difference between the required knowledge level θ and the member's current knowledge level b_{ij}, that is $K_{ij} = \theta - b_{ij}$.

5.1 Simulation

Before the simulation runs, the decision between two task-member assignment options: 'completing a task by the same member' versus 'completing a task by rotating different available members', has to be made by the managers. Then according to the simulation model shown in Figure 1, the simulation starts to run. No rework is expected in the first iteration, so the simulation skips the rework adjustment and goes to the next step, the task-member assignment. Since a member may work on several project tasks at the same time, it is necessary to consider the task workload and the work schedule of each member. A task with a workload value '1' means that the task requires 100% of a member's time and effort to work on it. If a member's schedule is not 100% available, he/she cannot be assigned to this task. After tasks are assigned, members will perform the assigned tasks that each consumes a number of times, which is estimated using triangle distribution (i.e., optimistic, expected, and pessimistic completion times). Task workload, task duration, and each member's work schedule can be estimated by interviewing managers/experts from the corresponding functional areas. The following shows one possible outcome of task-member assignments by simulation: $X_1 \to M_1$, $X_2 \to M_{11}$, $X_3 \to M_2$, $X_4 \to M_7$, $X_5 \to M_{10}$, $X_6 \to M_{18}$, $X_7 \to M_6$, $X_8 \to M_{16}$, $X_9 \to M_{11}$, $X_{10} \to M_{13}$, $X_{11} \to M_{22}$, $X_{12} \to M_2$, $X_{13} \to M_{17}$, $X_{14} \to M_{20}$, $X_{15} \to M_{28}$, $X_{16} \to M_7$, $X_{17} \to M_{19}$, $X_{18} \to M_{26}$, $X_{19} \to M_{25}$, and $X_{20} \to M_{26}$. It should be noted that members M_2, M_7, and M_{26} are assigned to more than one task in this case. This is due to the fact that the workloads of the tasks assigned to them do not use up all the available hours from their work schedules.

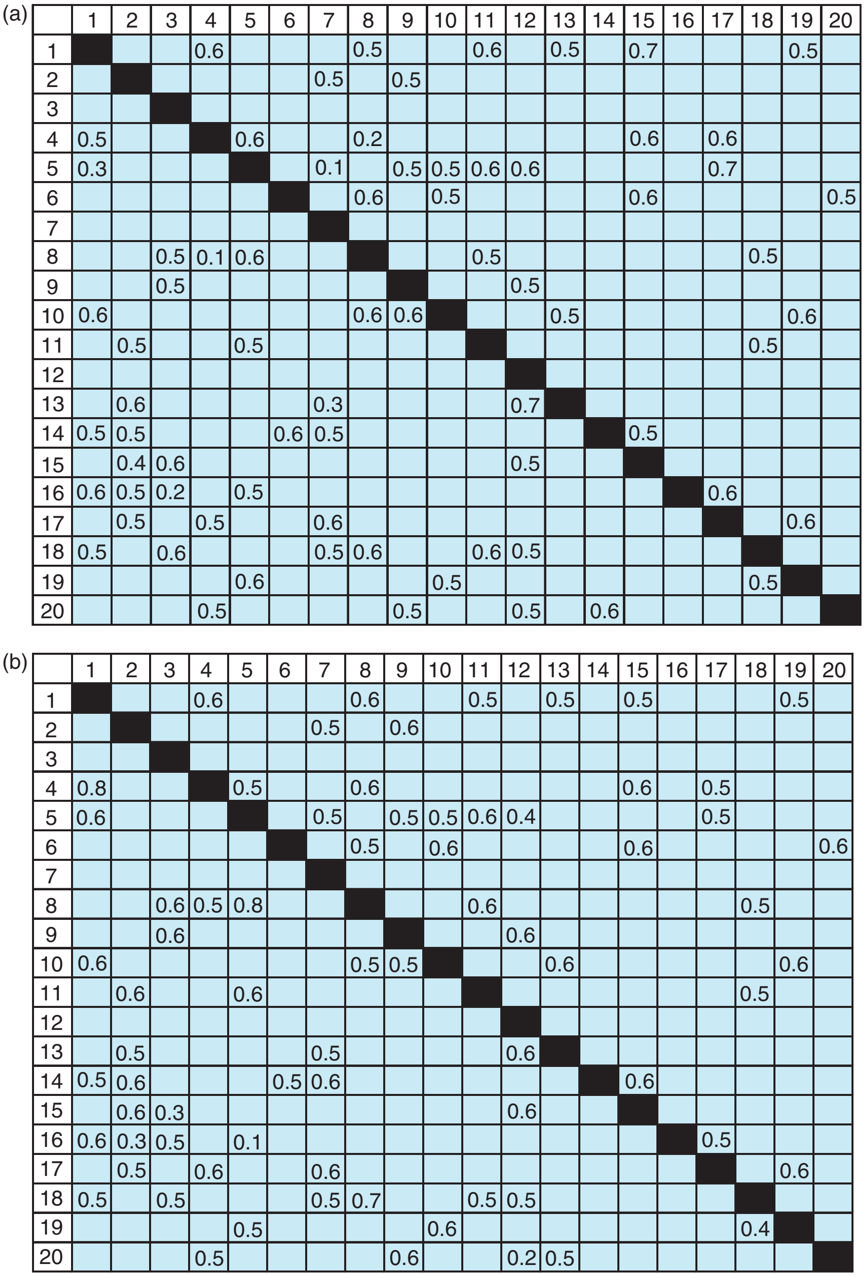
For each task, the functional knowledge level of the member in charge will also influence the progress of the task. The estimated duration of each task in this study is based on the assumption that the member in charge is an expert in the corresponding functional area. Therefore, if the member selected for the task is not knowledgeable about the area, the task progression will not be as good as the estimation.

When a task is completed, simulation will check to see whether this task is related to the other tasks or not. If so, rework (or iteration) between this task and its related tasks will be required in the simulation. Figure 3 shows DSMs of rework probability and rework impact used in the 20-task example, which determine the probability and the number of iterations required in the simulation. The values of rework probability and rework impact should be carefully estimated by the experienced managers/experts. A rework probability 0.5 at the location of tasks (i, j) means that there is a 50% probability for task i to rework after task j is completed. The amount (or percentage) of rework for a given task is determined by its value in the rework impact DSM. For example, after task X_{13} is completed, the simulation identifies that tasks X_1 and X_{10} are dependent on task X_{13} and knows that the two tasks both have a rework probability 0.5 with task X_{13} as indicated in the rework probability DSM. For task X_{10}, the simulation generates a random number between 0 and 1. If the number is not greater than 0.5, rework is scheduled for task X_{10} and the simulation records its impact value, which is 0.6 indicated in the rework impact DSM. At the beginning of next iteration, the remaining work of task X_{10} will be set as 60%. A similar procedure will be taken for task X_1 to determine the need of rework and record the amount of rework in the next round, if necessary.

As tasks continue to proceed in the simulation, the learning curve improvement will adjust the knowledge level of each member who has performed the assigned task for a period of time. Such learning curve improvement will be effective in the next iteration. At the end of each iteration, if all the tasks are completed and no further rework is needed, the simulation stops. Each iteration represents one unit of time (e.g., a day in our example). The final value of time index is the estimated project completion time.

5.2 Analysis of Results

To analyze the factors that have an effect on the project completion time, an experiment using ANOVA is carried out. Five factors will be tested and they are: (1) rework probability; (2) rework impact; (3) availability



(a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	■			0.6				0.5			0.6		0.5		0.7				0.5	
2		■					0.5		0.5											
3			■																	
4	0.5			■	0.6			0.2							0.6		0.6			
5	0.3				■		0.1		0.5	0.5	0.6	0.6					0.7			
6						■		0.6		0.5					0.6					0.5
7							■													
8			0.5	0.1	0.6			■			0.5							0.5		
9			0.5						■			0.5								
10	0.6							0.6	0.6	■			0.5						0.6	
11		0.5			0.5						■							0.5		
12												■								
13		0.6					0.3					0.7	■							
14	0.5	0.5				0.6	0.5							■	0.5					
15		0.4	0.6									0.5			■					
16	0.6	0.5	0.2		0.5											■	0.6			
17		0.5		0.5			0.6										■		0.6	
18	0.5		0.6				0.5	0.6			0.6	0.5						■		
19					0.6					0.5								0.5	■	
20				0.5					0.5			0.5		0.6						■

(b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	■			0.6				0.6			0.5		0.5		0.5				0.5	
2		■					0.5		0.6											
3			■																	
4	0.8			■	0.5			0.6							0.6		0.5			
5	0.6				■		0.5		0.5	0.5	0.6	0.4					0.5			
6						■		0.5		0.6					0.6					0.6
7							■													
8			0.6	0.5	0.8			■			0.6							0.5		
9			0.6						■			0.6								
10	0.6							0.5	0.5	■			0.6						0.6	
11		0.6			0.6						■							0.5		
12												■								
13		0.5					0.5					0.6	■							
14	0.5	0.6				0.5	0.6							■	0.6					
15		0.6	0.3									0.6			■					
16	0.6	0.3	0.5		0.1											■	0.5			
17		0.5		0.6			0.6										■		0.6	
18	0.5		0.5				0.5	0.7			0.5	0.5						■		
19					0.5					0.6								0.4	■	
20				0.5					0.6			0.2	0.5							■

Figure 3. Rework probability DSM (a) and rework impact DSM (b).

of member schedules; (4) learning curve efficiency; and (5) task-member assignment options. The experimental design in Table 1 shows that factors A and B both have three levels while factors C, D, and E have two levels each, which result totally in 72 treatment combinations. Each treatment combination will be given 50 replications in the experiment.

From Table 2 (P-value = 0.05), all the factors have significant effects on the project completion time except factor D; and all the interactions are significant except for the interactions of A*D, B*D, and D*E. The following conclusions are made according to the ANOVA table:

(1) Factor C (availability of member schedules) has the strongest effect than the other factors. This can be seen from the highest F-ratio of factor C (F= 1702.385) as compared with the others. Therefore, even though the time performance in a complex project can be influenced by different factors, the most important factor is the availability of human resources (i.e., engineers/members) in this case.

Table 1. Treatment levels in the simulation experiment.

Factors	Levels	Detail
(A) Rework probability	High	80% are randomly generated between 0.5 and 0.6. The other 20% are between 0 and 1
	Medium	80% are randomly generated between 0.3 and 0.4. The other 20% are between 0 and 1
	Low	80% are randomly generated between 0.1 and 0.2 The other 20% are between 0 and 1
(B) Rework impact	High	80% are randomly generated between 0.5 and 0.6. The other 20% are between 0 and 1
	Medium	80% are randomly generated between 0.3 and 0.4. The other 20% are between 0 and 1
	Low	80% are randomly generated between 0.1 and 0.2 The other 20% are between 0 and 1
(C) Availability of member schedules	High	The average availability of member schedules is 80%
	Medium	The average availability of member schedules is 60%
(D) Learning curve efficiency	High efficiency	$E_{ij} = 0.02$
	Low efficiency	$E_{ij} = 0.1$
(E) Task-member assignment options	Option 1	Completing a task by the same member
	Option 2	Completing a task by rotating different available members

Table 2. ANOVA table.

Source	Type III sum of squares	*df*	Mean square	*F*	Significance
A	286435.941	2	143217.970	454.216	0.000
B	442268.077	2	221134.039	701.326	0.000
C	536776.022	1	536776.022	1702.385	0.000
D	434.723	1	434.723	1.379	0.240
E	161242.402	1	161242.402	511.380	0.000
A*B	56216.353	4	14054.088	44.573	0.000
A*C	9026.885	2	4513.443	14.314	0.000
A*D	176.405	2	88.202	0.280	0.756
A*E	2888.795	2	1444.398	4.581	0.010
B*C	18368.762	2	9184.381	29.128	0.000
B*D	989.562	2	494.781	1.569	0.208
B*E	2407.835	2	1203.918	3.818	0.022
C*D	1582.714	1	1582.714	5.020	0.025
C*E	162046.503	1	162046.503	513.930	0.000
D*E	1.563	1	1.563	0.005	0.944
Error	1126596.453	3573	315.308		
Total	2807458.993	3599			

(2) Factors A (rework probability) and B (rework impact), which represent the complexity level of the project task structure and task relations, also significantly impact the project completion time. Rework probability determines the iteration between related tasks and rework impact contributes the number of iterations. Project managers should pay more attention to clarifying the entire project task structure and simplifying the task relations, so that the chance and number of iterations are reduced.

(3) In factor E, the first task-member assignment option ('completing a task by the same member') requires much more time than the second option ('completing a task by rotating different available members'). The main reason is that the transition time (i.e., 0.5 day), which is the additional time needed for a member to take over a partially completed task left by a previous member, chosen in this study is low. If the transition time is long for most tasks in the project, the outcomes may be different.

(4) Although this study shows factor D (learning curve efficiency) is not significant, we do not suggest that project managers ignore the learning curve factor when estimating the completion time for any projects. The average completion time estimated by simulation in the 20-task project example is

51.2 days, which is not generally considered as a long-term project. We expect the learning curve may play a bigger role in complex engineering projects that usually takes months, even several years to complete, because members will have a better chance to develop their efficiency and improve the task performance.

(5) The interaction of C*E shows a stronger effect ($F = 513.930$) than the other interactions. In general, project managers would prefer a task to be completed by the same member (option 1 in factor E) in order to make their job easier for human resource allocations. And of course, a better task performance will be expected because having a member stay with the same task will keep this member learning to improve with the task. Therefore, when the level of factor C (availability of member schedules) is 'High', the likelihood that a project manager will choose the task-member assignment option 1 is high. However, as human resources become limited and each member's schedule turns out to be tight (i.e., the level of factor C is 'Medium' or 'Low'), project managers may allow a task to be completed by rotating different available members (option 2 in factor E), so that the task will be able to continue without much delay.

(6) Even though factor D (learning curve efficiency) is not significant in this example, it is interesting that the interaction of C*D shows a significant effect. The reason is that if members have more available hours, it is more likely that each task will be handled and completed by the same member. In such cases, each member will have a better opportunity to continue improving his/her task performance and thus reduce the overall completion time of the assigned task.

6. Conclusions

In this article, we developed a research framework with a simulation model to help estimate the project completion time and analyze the major factors that affect the estimation for concurrent engineering project management. According to task clustering using DSM, the complexity of project task structure is clearly understood. The task-member assignment model employed in the simulation facilitates the goal of assigning the right members to the right tasks at the right time in terms of each member's knowledge, teamwork capability and their working relationships. Rework probability and rework impact represented by DSM control task iterations are often occurred in concurrent engineering projects. Each member's knowledge improvement in the simulation is modeled by a learning curve. The work schedule of members and the workload of tasks are also incorporated in the simulation in order to cope with the dynamic environment of the project. According to the simulation results, the major factors that significantly affect the project completion time are identified using ANOVA. Therefore project managers can focus more on those significant factors to reduce the project completion time.

The major contributions of this research are: (1) the DSM method, which reveals the entire project task structure and task relations, overcomes the limitation of traditional PERT/CPM method that cannot handle task rework/iteration; (2) the simulation model is not only able to help estimate the project completion time, but also offers managers the solution of task-member assignments; and (3) the simulation experiment and the ANOVA analysis give project managers an insight into those factors having significant effects on the project completion time, thus the problems that delay the project can be solved more efficiently and effectively.

The limitations of this research and future extensions are summarized as follows:

(1) In this article, we treat the rework probability and rework impact as constants. It is expected that the rework probability and rework impact of a task will likely be decreased (or nonconstants) over iterations of rework. In some other cases, project managers may limit the number of iterations allowed for rework due to some resource-demanding tasks. One future research extension is to search for a mathematical function that is able to describe the nonconstant forms of rework probability and rework impact including the rates of decrease over iterations. An additional constraint to limit the number of iterations will also be added to the simulation in the future research.

(2) The nature and characteristics of projects vary because different projects have different goals and are carried out in different contexts. A member's learning capability is heavily influenced by how the project is planned and organized for execution. Most learning curve models are specifically designed to fit a certain type of project or task. It is difficult to determine a learning curve model just right for different individuals due to each one's knowledge base, experience, and learning motivation, etc. The simulation model in this article focuses on concurrent engineering projects, which are decomposed into inter-related tasks. These tasks, not the same as simple repetitive jobs (i.e., the assembly line tasks), can be carried out simultaneously, require inputs from the other tasks, or need rework. Our simulation does not limit users to a specific learning curve model. The reason we use Hanakawa's learning curve model is that the

software development project is a good example of concurrent engineering. Software projects are often divided into tasks by different functional modules, which carry the same characteristics of concurrent engineering tasks. It is recommended that the managers should choose the most appropriate learning curve model(s) based on different task types.

(3) Every task in this study shares the same (or constant) transition time, which is set as a low value. We expect that the outcomes may not be the same if different tasks require different (or non-constant) transition times, which are long overall. Another future research will be examining different modes of transition times (i.e., short vs medium vs long, and constant vs non-constant, etc.) and their impacts on the project completion time.

(4) The simulation model in this research handles the project tasks in a concurrent engineering fashion. In practice, some tasks may not be allowed to start until their predecessors are completed (e.g., a predecessor task may provide an essential part or tool for its successor task). This future research will aim at building a mixed model that can accommodate both task structures of sequential and concurrent engineering in the simulation.

References

1. Van Dorp, J.R. and Duffy, M.R. (1999). Statistical Dependence in Risk Analysis for Project Networks Using Monte Carlo Methods, *International Journal of Production Economics*, **58**: 17–29.
2. Steward, D.V. (1981). The Design Structure System: A Method for Managing the Design of Complex Systems, *IEEE Transactions on Engineering Management*, **28**(1): 71–74.
3. Ahmadi, R., Roemer, T. and Wang, R. (2001). Structuring Product Development Processes, *European Journal of Operational Research*, **130**: 539–558.
4. Chen, S.J. and Lin, L. (2002). A Project Task Coordination Model for Team Organization in Concurrent Engineering, *Concurrent Engineering: Research and Applications*, **10**(3): 187–203.
5. Chen, S.J. and Lin, L. (2003). Decomposition of Interdependent Task Group for Concurrent Engineering, *Computers and Industrial Engineering*, **44**(3): 435–459.
6. Chen, S.J. (2005). An Integrated Methodological Framework for Project Tasks Coordination and Team Organization in Concurrent Engineering, *Concurrent Engineering: Research and Applications*, **13**(3): 185–197.
7. Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A. (1994). A Model-based Method for Organizing Tasks in Product Development, *Research in Engineering Design*, **6**(1): 1–13.
8. Eppinger, S.D. (2001). Innovation at the Speed of Information, *Harvard Business Review*, **79**: 149–158.
9. Hu, J., Liu, J. and Prasad, B. (2003). A Constraint-driven Execution Plan for Maximizing Concurrency in Product Development, *Concurrent Engineering: Research and Applications*, **11**(4): 301–312.
10. Kusiak, A. and Wang, J. (1993). Efficient Organizing of Design Activities, *International Journal of Production Research*, **31**(4): 753–769.
11. Lee, S.G., Ong, K.L. and Khoo, L.P. (2004). Control and Monitoring of Concurrent Design Tasks in a Dynamic Environment, *Concurrent Engineering: Research and Applications*, **12**(1): 59–66.
12. Michael, D., Zahra, I. and Jacques, R. (2005). Managing and Organizing Concurrent Processes According to the CMM Levels, *Concurrent Engineering: Research and Applications*, **13**(3): 241–251.
13. Smith, R.P. (1997). The Historical Roots of Concurrent Engineering Fundamentals, *IEEE Transactions on Engineering Management*, **44**(1): 67–78.
14. Yassine, A. and Braha, D. (2003). Complex Concurrent Engineering and the Design Structure Matrix Method, *Concurrent Engineering: Research and Applications*, **11**(3): 165–176.
15. Prasad, B. (1998). Decentralized Cooperation: A Distributed Approach to Team Design in a Concurrent Engineering Organization, *Team Performance Management*, **4**(4): 138–165.
16. Wrights, T.P. (1936). Factors Affecting the Cost of Planes, *Journal of the Aeronautical Science*, **3**(4): 122–128.
17. Hanakawa, N., Morisaki, S. and Matsumoto, K. (1998). A Learning Curve Based Simulation Model for Software Development, *Proceedings – International Conference on Software Engineering*, pp. 350–359.
18. Carmel, E. (1994). Time-to-Completion in Software Package Startups, *Proceedings of the IEEE International Conference on System Sciences*, Vol. 4, pp. 498–507.
19. Chung, W.Y. and Guinan, P.J. (1994). Effects of Participative Management, *Proceedings of the ACM SIGCPR Conference*, pp. 252–260.
20. Clark, K.B. and Wheelwright, S.C. (1992). Organizing and Leading 'Heavyweight' Development Teams, *California Management Review*, **34**(3): 9–28.
21. Johnson, D. and Johnson, F. (1991). *Joining Together: Group Theory and Group Skills*, Englewood Cliffs, NJ: Prentice Hall.
22. Lanigan, M.J. (1994). Task Estimating: Completion Time Versus Team Size, *Engineering Management Journal*, **4**(5): 212–218.
23. Chen, S.J. and Lin, L. (2004). Modeling Team Member Characteristics for the Formation of a Multifunctional Team in Concurrent Engineering, *IEEE Transactions on Engineering Management*, **51**(2): 111–124.
24. Browning, T.R. (1998). Modeling and Analyzing Cost, Schedule, and Performance in Complex System Product Development, PhD Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA.
25. Browning, T.R. and Eppinger, S.D. (1998). A Model for Development Project Cost and Schedule Planning, Working Paper no. 4050, M.I.T. Sloan School of Management, Cambridge, MA.

Enzhen Huang

Enzhen Huang has an MS from Industrial Engineering at the Montana State University, Bozeman. His research interests include software project management, supply chain management and artificial intelligence. He has published an article in the proceedings of PICMET conference.

Shi-Jie (Gary) Chen

Shi-Jie (Gary) Chen is an Assistant Professor of Mechanical and Industrial Engineering at the Montana State University, Bozeman. He completed his PhD in Industrial Engineering (1999) from the State University of New York at Buffalo. His research interests include concurrent engineering and management, project management, health care systems, computer simulation, and CAD/CAM/CIM. He has published articles in Computer Integrated Manufacturing Systems, International Journal of Production Research, Computers in Industry, Concurrent Engineering: Research and Applications, Computers and Industrial Engineering, IEEE Transactions on Engineering Management, and the proceedings of IERC, EDA, IE-TAP, PICMET conferences.