A SIMULATION RESEARCH FRAMEWORK FOR CONCURRENT ENGINEERING PROJECT MANAGEMENT

by

Enzhen Huang

A thesis Submitted in Partial Fulfillment
of the Requirements for the Degree

of

Master of Science

in

Industrial and Management Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

April, 2005

APPROVAL

of a thesis submitted by

Enzhen Huang

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Dr. Shi-Jie Chen

Approved for the Department of Mechanical and Industrial Engineering

Dr. R. Jay Conant

Approved for the College of Graduate Studies

Dr. Bruce R. McLeod

STATEMENT OF PERMISSION TO USE

Enzhen Huang

April 2005

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

ABSTRACT

In concurrent engineering, project tasks usually are interdependent among each other. Iterations, which are required for the interdependent tasks, make traditional PERT/CPM not applicable for the estimation of the project completion time. In addition, carrying out a large scaled project in a dynamic environment has to deal with various factors throughout the entire project life cycle. When estimating the project completion time, previous research often focused on one subject of interests and assumed the other factors causing little effects on the overall project duration. The objective of this thesis is to develop a simulation 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 to 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. According to the findings from the ANOVA, heuristic rules are developed to improve the performance of task-member assignments. The effectiveness of the research framework, the simulation model and the heuristics is demonstrated by an illustrative example.

CHAPTER 1

INTRODUCTION

Project management is the process of managing, allocating, and timing resources to achieve a given goal in an efficient and expedient manner (Badriu and Pulat, 1994). Project management has been receiving more and more attention during the last few decades and become a separate domain. Organizations and companies have seen the effectiveness and benefits that can be derived from using project management tools and methodologies, especially in a dynamic time-intense and resource-intense environments.

Because of the application of new technologies and the keen competition in the market, traditional method of product design and development has shown the shortcomings of low efficiency and slow-reacting to the market. Therefore, concurrent engineering (CE) emerges with its promise to shorten the product development process, to increase product quality, to decrease product cost and to improve services (Yang and Jiang, 1999). Concurrent engineering is the early involvement of a cross-functional team to simultaneously plan product, process, and manufacturing activities (Hartley, 1992; Susman and Dean, 1992). The concurrent engineering methods are widely used in the projects where tasks are performed simultaneously and task reworks are often occurred. It has been proved that concurrent engineering project management has higher efficiency than the traditional project management methods in the projects such as new product design, software development and supply chain interactions, etc.

Now that more and more organizations are using projects management in concurrent engineering environment, it is noted that traditional project management tools, i.e., Critical Path Method (CPM) and Program Evaluation Review Technique (PERT), which have been widely used in estimating the project duration, is not suitable for projects in a concurrent engineering project (Duffy and van Dorp, 1998).

In a highly concurrent engineering project, tasks are usually interdependent among each other, which require a group of people with different engineering background to work together. The interdependent task group often results in task rework or iteration. Levitt et al. (1999) pointed out that CPM/PERT scheduling tools do not represent the "coordination and rework overhead" of executing flexible and interdependent activities in parallel. Thus, concurrent engineering can lead to severe cost overruns and schedule delays relative to the inherent optimism of CPM/PERT schedules for such projects. A new approach for estimating project completion time in concurrent engineering project is needed to overcome the limits in traditional method.

Making best use of available human resources to improve the efficiency of project execution is also an important aspect in project management. Nevertheless, 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. Chen and Lin (2004) 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 involve 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 (1936) first discovered this fact in manufacturing assembly line and defined it as "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. (1998) studied learning curve in the software development context and developed a mathematical model of 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 interests and assumed the other factors causing little effects on the overall project duration. This thesis research 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 the project duration is estimated, solutions of task-member assignments, which are useful for managers, will also be suggested. The simulation model studies the following 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 the simulation model, an analysis of variance (ANOVA) is performed to analyze the data and identify the significant effect of each source of variance.

The objectives of this thesis are as follows:

1) To establish a research framework that integrates various processes (i.e., task clustering, task-member assignment, learning curve improvement, task rework control) for the study including data collection, simulation and data analysis.

2) To develop a simulation model that helps estimate project completion time and dynamically perform task-member assignment.

3) To identify the factors that have significant effects on project completion time by a simulation experiment and the ANOVA analysis.

4) According to the major factors identified by the ANOVA analysis, develop a heuristic rule to improve the task-member assignment performance

CHAPTER 2

LITERATURE REVIEW

Project Management

The Definition of Project Management

A project is a series of activities and tasks to achieve a specific goal. Typically, projects are executed after signing a contract with a customer; or they are internally initiated with the intention to introduce a new product to the marketplace. [014]

Badriu and Pulat (1994) list five required characteristics of project:

1) An identified scope and a goal
2) A desired completion time
3) Availability of resources
4) A defined performance measure
5) A measurement scale of review of work

Based on the project characteristics, the authors define project management as "*the process of managing, allocating, and timing resources to achieve a given goal in an efficient and expedient manner*".

Kerzner (2001) further involves systems approach to the definition of project management with emphasis on personnel assignment:

Project management is the planning, organizing, directing and controlling of company resources for a relatively short-term objective that has been established to complete specific goals and objectives. Furthermore, project management utilizes the systems approach to management by having functional personnel (the vertical hierarchy) assigned to a specific project (the horizon hierarchy).

The Project Management Institute (PMI) defines the project management of knowledge (PMBOK, 1987) with eight major functional areas: scope, quality, time, cost, risk, human resources, contract/procurement, and communications.

System Approaches and Integrated Project Management

Although most research literatures suggest a system approach to the management of projects, practitioners and researchers tend to focus on a specific topic in a project. Pinto and Covin J. K. Pinto and J. G. Covin (1989) found that very little systematic research of project management has distinguished between the project type and the strategic and operational problems of various projects. Shenhar (1998) also concluded that research literature on the project management has been quite slow in its conceptual development and still suffers from a scanty theoretical basis, after studying of 26 case projects. The author believes that one of the main impediments in the study of projects has been the absence of constructs and the little distinction that has been made between the project type and its managerial and organizational style. He calls for a more project-specific contingency approach to project management in organizations.

The fierce competition in industry and the development of information technology is making project more and more complex so that systematic and integrated approaches are needed for modern project management. However, there are only limit researchers who started to use systems theory and integrate two or more aspects in project management in order to enhance the effectiveness of project. Kibler (1992) notices the interrelationship between various components of a properly integrated cost and schedule system. Through the integrated cost/schedule analysis, project progress trends can be analyzed for each area of work. The level of performance, as well as the efficiency of the work itself, is then measured. Riggs et al. (1994) proposed a computerized method for integrating technical, cost, and schedule risk within an analytical methodology. Technical, cost, and schedule preference are represented as the project manager's utility function. The expected value of the utility function is calculated for various alternatives, in the context of a decision tree. Jergeas and Revay (1999) reported an overall 30% of cost reduction in construction projects by using an integrated approach which combines four key concepts and disciplines involved: strategic alliances, value engineering, risk management, and constructability. A simulation-based model NETCOR (NETworks under CORrelated uncertainty) developed by Wang and Demsetz (2000) takes external factors of a construction project such as weather, labor, and site conditions into consideration. The variability cased by the factors is evaluated in completion time estimation. Sensitivity study of the factors can provide management with a better sense of what to control on each path, particularly on large projects. Ballard and Howell (2003) tried to apply lean production theory to project management by viewing projects as temporary production systems. The author developed a “Lean Project Delivery System (LPDS)”, which is structured to achieve “lean” project goal while maximizing value and minimizing waste. Lean project management differs from traditional project management not only in the goals it pursues, but also in the structure of its phases, the relationship between phases and the participants in each phase.

The research above notices the dynamics factors and captures one or more aspects in a project. However, there is a lack of quantitative study to identify the important factors in project management. Furthermore, a more expendable and flexible model is necessary because every project is different. These researches also overlook the interactions among the tasks in a project. Thus not much effort is effort is spent on the rework study.

Some recent research is focus on providing useful tools or methods for integrated project management. Moselhi et al. (2004) presents a web-based tool that supports project time and cost control in an integrated manner. The system utilizes object-oriented modeling to represent the process of project delivery and map the process of project control. Eighteen key indicators are considered to represent the resources utilized in each control object and serve as sensors to highlight problematic areas associated with unfavorable performance. Anbari (2004) proposes the earned `value project management method which integrates three critical elements of project management (i.e., scope management, cost management and time management). The method requires the periodic monitoring of actual expenditures and physical scope accomplishments, and allows calculation of cost and schedule variances, along with performance indices. It allows forecasting of project cost and schedule at completion and highlights the possible need for corrective action.

Estimating Project Completion Time

Estimating project completion time is one of the most important aspects in project management. It is an essential and preliminary step for project scheduling, resource allocation and risk assessment. Solomon and Carter (1995) categorized the existing methodologies into five categories:

(1) Analytical approach: the actual task cost and completion time distribution can be derived from accurate data by applying analytical approach. This approach is applicable when the task network is small and the probability density functions (PDF) and the cumulative density functions (CDF) of the task distributions are integrable.

(2) Numerical Integration: this approach is used when the actual task cost and completion time distribution cannot be derived from the actual data. Computer packages are used to provide approximations in such cases.

(3) Moments method: this approach is particularly useful when task time distributions are in empirical form or when the analytical method cannot be sued because the mathematics becomes unmanageable.

(4) PERT/CPM: PERT/CPM analysis is a very popular approach. However, the limit of PERT/CPM is discussed by researchers (Duffy and van Dorp, 1998, Levitt et al. 1999).

(5) Simulation: the most flexible approach for estimation of project completion time because it can use any kind of input distribution and it can handle complex project with modern computer speed.

Simulation approach has the advantages of flexibility and is able to act in most project scopes and/or environment. As early as 1980s, Ahuja (1985) used simulation to estimate project completion time and analyze the variations. From the information that is collected normally for a progress update of the tactical plan and by simulating the project environment, the combined impact of the uncertainty variables is predicted for each progress period. By incorporating the combined impact in the duration estimates of each activity duration distributions, the probability of achieving the original project completion time and of completing the project at any other time is computed. Bandopadhyay and Sundararajan (1987) allowed more than one critical path in a PERT network because of the random activity durations. The authors' simulation model utilized the concept of criticality indexing to define the critical path. Putcha and Rao (1991) used the probabilistic network analysis for project completion time for various networks. Based on CPM approach, a Monte Carlo Simulation is conducted for two different networks. Uniform and normal distribution are used as the random variables which are the activity durations and the activity precedence relationships. Such simulation models and other similar approaches (Markevicius and Rouphail, 1986; Ramani 1986; Orczyk and Hancher, 1987; Riggs, 1989; Finley and Fisher, 1994) are based on a CPM/PERT network analysis. In order to represent the variation of task times in a network, stochastic processes are employed to quantify task completion time distributions. Bella et al. (1995) concluded that Erlang distribution of activity times is able to provide an accurate estimate of a project completion time distribution for a large range of practical situation. Schmidt (2000) presents a new technique for computing the exact overall duration of a project. The task durations use a probability density function (p.d.f.) which combines piecewise polynomial segments and Dirac delta functions, defined over a finite interval. A semi-analytical procedure is proposed to compute the cumulative distribution function (c.d.f.) directly by integrating a linear transformation of the p.d.f, of the task durations. Abdelkader (2004) further discussed the advantages of using Weibull distribution to describe task time to evaluate project completion time.

Other methods to estimate project completion time are not dependent to traditional CPM/PERT network analysis. These simulation models focus on the individual task variability so that they cope with the uncertainties in reality. A simple non-linear mathematical programming model is used by Meherez and David (1999) to solve the problem of determining the timing of risky R&D tasks (activities) and non-routine tasks. This model ignores issues related to the flow of information gathered at each time point as a function of the amount of resources invested to this time point. In a spacecraft project, Kandathil (2003) noticed that many activities bear uncertainty on its progress and completion, chance of rework and going back to previous stages of the project. The author used the stochastic ability of Markoff chain analysis in his proposed model and the results are of Markoff model shows significant improvement over the prevailing models. Again, the model is lack of flexibility and expandability.

The review of previous works in estimating project completion time shows that PERT/CRM are still widely used in spite of its limitations. Although researchers try to incorporate probabilistic methods into PERT/CRM to solve the stochastic problems in reality, the complexity of problem will be very difficult to deal with if the size of the network is large. In addition literature review didn't provide any case that solved the task rework problem using PERT/CRM approach. Some other research models have very good result on a specific project. However, these models are not flexible or expandable to adapt other project quickly.

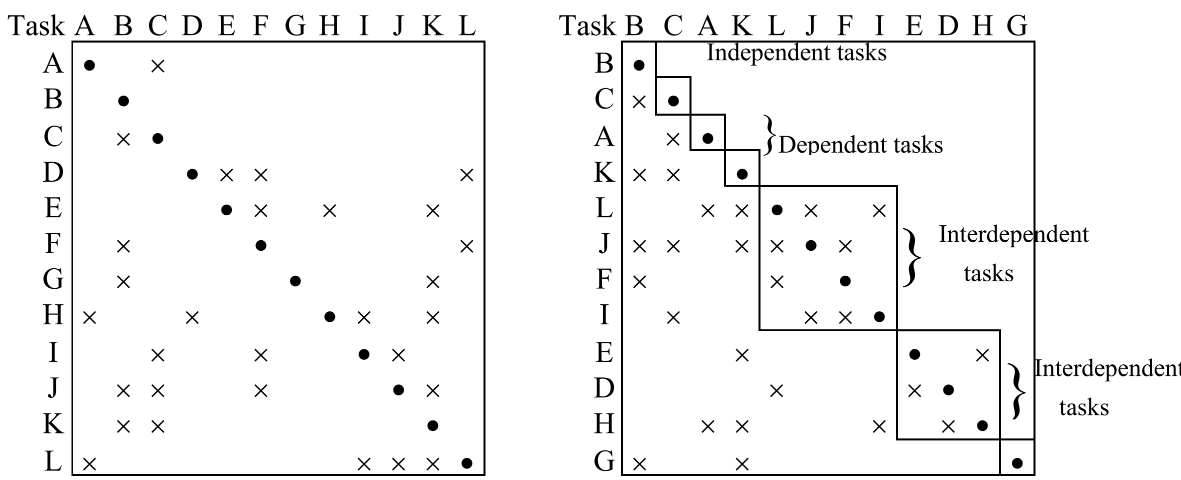
Concurrent Engineering and Multi-functional Team

The fast changing globalization and market competition requires companies to focus on efficiency. They will have to adopt an organizational design that is efficient in acquiring and processing additional information but also one that is capable of processing rich information. Wheelwright and Clark (1992) describe such an organizational design which includes an early involvement of constituents who belong to a cross-functional team that works on different phases of product development concurrently. The 'integrated problem solving' approach, which is later called concurrent engineering, links the upstream and downstream groups involved in product development in time and in the pattern of communication. In a review of concurrent engineering by Koufteros et al. (1999), the authors conclude three basic elements of concurrent engineering: early involvement of participants, the team approach, and the simultaneous work on different phases of product development. Among these three elements, researchers conclude that central component of the implementation of concurrent engineering is the deployment of multi-functional teams (MFTs) (Clark and Fujimoto, 1991; Chen and Lin, 2002).

A multi-functional team consists of engineer members who represent many aspects of the product life cycle, such as marketing, sales, R&D, design, manufacturing, purchasing, testing, quality control, and service. These members cross traditional departmental boundaries to share their ideas while negotiating conflicts during a product's design phase (O'Grady and Young, 1991). When managed effectively, a multifunctional design team will offer improved communication, strong identification and commitment to the assigned tasks, and a focus on cross-functional problem solving (Clark and Wheelwright, 1992). Members in a successful multifunctional team must speak a common language despite of different engineering knowledge body. It is also recommended that the members should be located near each other to encourage ongoing communications. It helps to provide workspaces for everyone connected with the team, even for people in ancillary disciplines who spend only part of their time on the project (Turino, 1991). However, as projects become more complex, so do the process of design and the project teams. This will inevitably degrade team performance when the team size becomes too large to manage. Both the team members and managers need a useful tool to clearly represent the project and provide a "common language" in team communication. The effectiveness of communication also depends on the number of communication links among team members. Therefore, an effective and efficient communication will be very difficult as the team size increases. Research indicates that the preferable team size would be between two and six (Johnson and Johnson, 1991; Clark and Wheelwright, 1992; Carmel, 1994; Chung and Guinan, 1994; Lanigan, 1994)

Design Structure Matrix (DSM), Task Clustering and Task-member Assignment

Design structure matrix (DSM) is a useful tool that is suitable for engineers to analyze a complex system by providing a clear view of the system as well as the interdependencies between its system elements. DSM is first introduced by Steward (1981) to analyze the process of engineering design. Figure 1 shows an example of DSM that 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. In DSM, information links among tasks are clearly revealed by the systematic mapping. For example, in Figure 1(a), the non-zero elements in row D represent that task D will receive information input from task E, task F and task L before task D is started. Likewise, the non-zero elements in column A represent that task A will provide information output to task H and task L after task A is completed. According to Steward's partitioning algorithm, the task sequence on the row and on the column can be rearranged and then turn the original DSM into a well-organized manner in which three basic task types (independent, dependent, and interdependent tasks) are clearly revealed as shown in Figure 1(b). When each non-zero element in the binary DSM is replaced by a numerical value (i.e., ranging from 0 to 1) to indicate the strength of task interaction, the DSM is called numerical DSM (See Figure 2).



(a) Original binary matrix (unpartitioned)

(b) Reordered binary matrix (partitioned)

Figure 1. A Binary Design Structure Matrix

There are various research in the past regarding the systems decomposition and architecture for product design and development. Alexander (1964) described the design process by decomposing designs into minimally coupled groups.

Task	B	C	A	K	L	J	F	I	E	D	H	G
B	•											
C	.5	•										
A		.2	•									
K	.6	.9		•								
L			.2	.6	•	.3		.5				
J	.4	.5		.8	.1	•	.7					
F	.5				.2		•					
I		.8				.5	.8	•				
E				.4					•		.6	
D					.4				.4	•		
H			.3	.5				.2		.1	•	
G	.3			.5								•

Figure 2. A Numerical Design Structure Matrix

Base on the DSM technique, Steward (1981) and Eppinger et al. (1990) analyzed parameter-level interactions to create design parameter groupings that must be solved iteratively. Kusiak and Wang (1993) focused on the decomposition of DSM to structure tasks and parameters in the detailed design stage. Ulrich and Eppinger (1995) developed a method for product architecture, but interactions are considered only after the architecture is chosen. They defined several types of product architecture in terms of how functional elements are mapped onto physical components and related the strategic importance of architecture choice to firm performance. Smith and Eppinger (1997a, b) described decomposition as a fundamental approach to handling complexity in engineering design. Kusiak and Wang (1993) used binary interactions to develop physical design layouts. Lovejoy (1992) related design decomposition to the organizational structure of complex product design processes. The author noted that an approach to solving complex problems lies in controlling the interactions between the elements. The author also proposed that interactions between elements in a design vary in strengths that relate to the speed of the development process. McCord and Eppinger (1993) used interactions between components to structure system teams in a development project. Chen and Lin (2002, 2003) quantified the task coupling strengths for the interdependent task group decomposition in a notion to simultaneously consider various downstream activities throughout the entire product life cycle. The authors further concluded that numerical DSM is more appropriate and efficient than binary DSM at revealing the interrelationships among system elements. From the research reviewed above, DSM is good at capturing the interactions or interrelationships among the system elements and grouping them according to their coupling strengths.

Pimmler and Eppinger (1994) analyzed the decomposition of product design by understanding the complex interactions between components of a design, which helps define the product architecture and organize the development teams. The authors proposed four types of interactions among the design tasks, such as spatial, energy, information and material interactions. Multiple DSMs were constructed based on each of these interaction types. Systems analysis can be performed either aggregately by combining different interaction types of DSMs together or separately on each interaction type of DSM, so that the underlying structure for a complex system will be better understood and revealed.

As for the purpose of decomposing large size of numerical coupling DSM, cluster analysis is useful to help cluster the strongly coupled tasks into the same group. Cluster analysis has been widely used by many researchers in clustering similar objects or data into groups that ensure the objects within a group are similar to one another (Chu, 1989; Duran & Odell, 1974; Everitt, 1980; Gordon, 1981; Hartigan, 1975; Kusiak, 1990). Chen and Lin (2003) provided a comprehensive review of clustering methods and proposed an approach to clustering a large coupling DSM into smaller and manageable sizes based on numerical coupling strengths. The authors compared two types of clustering methods (i.e., similarity coefficient methods and sorting-based algorithms) and concluded that similarity coefficient methods are more appropriate for clustering the numerical coupling DSM. The authors also introduced a performance measure, numerical interaction density (NDd) that measures the total coupling strength outside the block diagonal of DSM, to select the best solution from several alternatives after the clustering process.

Besides the complex interrelationships among the tasks of a concurrent engineering project, the efficiency of a project depends on the assignment of individual tasks to project team members. Because the different areas and knowledge bodies required by the different tasks, functional knowledge level must be considered in task-member assignment. However, the assignment method must also consider the teamwork capacity and the working relationships among the members because the tasks in a concurrent engineering project are closely related to each other and require the team members to cooperate and communicate with each other. There is not much research specifically about the task-member assignment topic in literature review. The task-member assignment model developed by Chen and Lin (2004) includes the three important team member characteristics with quantitative representations (i.e., multifunctional knowledge rating, teamwork capability rating, and working relationship rating). 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 this department. This will increase the flexibility when a key functional member is needed during the team organization. Second, to build a successful team, teamwork capability of team members is needed by taking their experience, communication skill, 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 a successful team organization. With the quantitative ratings of team member characteristics, the goal of assigning the right team members to the right tasks will be carried out by using a mathematical model (Chen and Lin 2004).

In summary, DSM is a popular and useful tool for concurrent engineering. Previous research in DSM has inspired a new way to effectively manage the large-scale systems such as new product design projects or supply chain networks. DSM helps identify the interdependencies within the concurrent engineering project. Multiple dimensions of a given project can be characterized by several DSMs with different interaction types. Numerical DSM with quantifiable measures shows the various degrees of interactions among system elements in a project. Furthermore, clustering technique helps to decompose a complex task network of a project into smaller and manageable task groups. With the understanding of the task relationships, tasks are assigned to team members according to members' knowledge level, teamwork capability and their working relationships.

Learning Curve

People can gain experience and improve their knowledge level in the area of the tasks or activities that they have been working on. Wrights (1936) first discovered this fact in manufacturing assembly line and defined it as "learning curve". Learning curve is important because an experienced worker can finish a task with lower cost, shorter time and yet higher quality. 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.

The characteristics and parameters of learning curves in different areas can be different. Because of the nature of the work and the different engineering environment, it is difficult to find a general learning curve model. The speed of learning also depends on the people performing the work and their knowledge background. Everett and Farghal (1994) evaluated several mathematical models to determine which best describes the relationship between the activity time or cost and the cycle number in a construction field environment. The authors also presented a methodology for predicting future activity time or cost based on completed activity data. For the knowledge-based tasks, Hanakawa et al. (1998) studied learning curve in the software development context and developed a mathematical model of 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.

Estimating project completion time need to study learning curve because the knowledge level of team members is regarded as one the most important factor in a concurrent engineering project.

Heuristics in Project Management

Heuristics are "intelligent methods" usually used in complex simulations or mathematical problems. In such problems, the complexity of the complete method makes it impossible or difficult to achieve optimal solutions. Heuristics are developed to reduce the complexity and obtain reasonably good answer. Wilson et al. (1993) defined a heuristic as:

A rule of thumb or guideline (as opposed to an invariant procedure). Heuristics may not always achieve the desired outcome, but are extremely valuable to problem-solving processes.

Assigning limit resources to tasks in a project can be difficult when the project size is big. In a PERT network, Bailey, et al. (1995) found it difficult to determine the task start dates, labor levels and tours for a minimum-cost and on-time schedule at the same time. In order to coordinate the time, cost and resource constraints, the authors developed a heuristic, which was proved to outperform the non-integrated two-step scheduling procedure by reducing the cost of labor and overhead and also perform nearly as well as the optimization procedure. Li and Kim (1996) developed a search heuristic for project scheduling problems with multiple resource constraints as well as precedence constraints. In the heuristics, a solution is represented with a string of numbers each of which denotes priority of each activity. The priorities are used to select an activity for scheduling among competing ones. Similar study with different heuristic rules can be found in the work of Khattab and Choobineh (1990), Neumann, K. and Zhan (1995), Erbasi and Sepil (1999), Selle and Zimmermann (2003) and Valls et al. (2003).

Because heuristic rules are usually developed focusing on one or more the critical constraints, some heuristics with different concentrations are combinable. Boctor (1993) studied the resource-constrained project scheduling problems in which the duration of each activity depends on the amount of resources allocated to its execution. Twenty-one heuristic scheduling rules are compared on 240 test problems divided into two main groups containing 50 and 100 activities, respectively. Each group contains one-resource, two-resource and four-resource problems. The objective is to minimize the overall project duration. The results showed that a combination of five-heuristics had the best solution.

CHAPTER 3

THE PROPOSED RESEARCH FRAMEWORK

The Research Framework

Figure 3 shows the research framework that includes 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 or experts (i.e., member schedules and characteristics). The other data can be derived from mathematical models (i.e., the task clustering model discussed in Chapter 2).

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. Since the major portion of this thesis focuses on the numerical studies, using heuristic rules to improve the simulation performance can help validate the results from the analysis. And of course, managerial decisions or operational involvements are always preferred by project managers and easy to implement. They cannot be overlooked in any real projects.

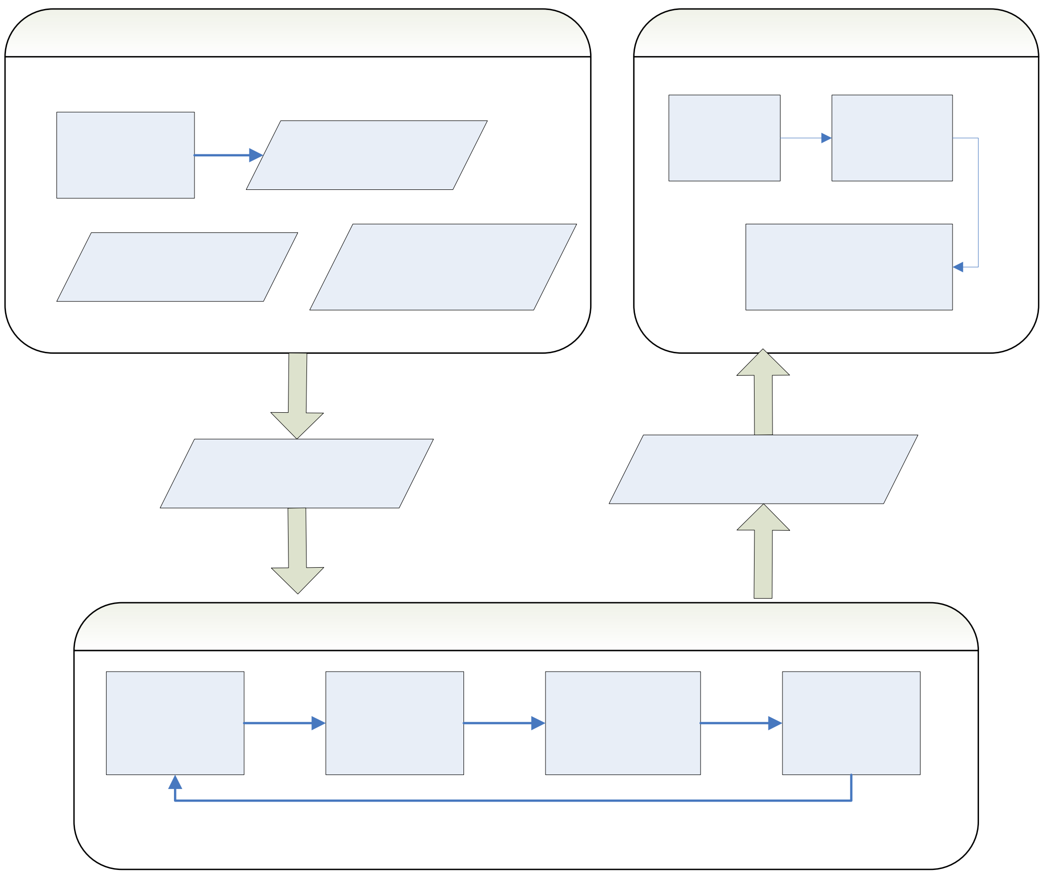


Figure 3. The Research Framework

<u>Task Clustering</u>

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 (1981) to reveal three basic task types in projects (i.e., independent, dependent and interdependent tasks). Although 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 (Johnson and Johnson 1991, Clark and Wheelwright 1992, Carmel 1994, Chung and Guinan 1994, Lanigan 1994). Chen and Lin (2002, 2003) 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 sub-groups based on numerical DSM and clustering technique. This decomposition model contains the following three steps:

Now that DSM has recorded the relationships among system elements, partitioning algorithm proposed by Steward (1981) can be performed to identify the interdependent groups of system elements. 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 group increases (Johnson & Johnson, 1991; Clark & Wheelwright, 1992; Carmel, 1994;

Chung & Guinan, 1994; Lanigan, 1994). Chen and Lin (2002, 2003) realized 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 sub-groups based on numerical DSM and clustering technique. This decomposition model with the following two procedures will be used in this step (Chen & Lin, 2002, 2003):

(1) Symmetrical Task Interaction Matrix: Since DSM is a matrix that only offers the information of task dependency for their 'from-to' descriptions, we need to further understand the task interaction by transforming numerical DSM into a symmetrical task interaction matrix for clustering purposes in the next step. If we assume that the input and output connections carry the same weight, the amount of interaction can be calculated by averaging each pair of symmetrical elements in a numerical DSM, because the interaction of any two tasks contain both information input and output connections. This symmetrical task interaction matrix, $SymDSM_{i,j}$, is expressed mathematically in the following for each pair of row i and column j:

$$SymDSM_{i,j} = (NumericalDSM_{i,j} + NumericalDSM_{j,i})/2 \quad (1)$$

(2) Decomposition of Large Interdependent Group: A large interdependent group is decomposed into smaller sub-groups using clustering technique. The key is to calculate the distance measures for the matrix. Quantified interaction strengths in

the symmetrical matrix, $SymDSM_{i,j}$, 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 the coupling values in the matrix do not affect the clustering results.

(3) Clustering Performance Evaluation by Numerical Interaction Density (NDd): For an n×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 (2003) developed a performance measure, Numerical Interaction Density (NDd), to help select the best clustering result. NDd, measuring the numerical interaction strengths outside the block diagonal of the clustered matrix, is formulated as $NDd = Ne\ /\ Outer\text{-}Cells$. *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. For example, a clustered matrix in Figure 4 shows that elements X1 and X2 are clustered in the same group while elements X3 and X4 are in

another group. The value of *Ne* is calculated by (0.1+0.2+0.3+0.3) = 0.9 and the number of *Outer-Cells* is 8, so that *NDd* is equal to (0.9/8) = 0.1125. The best task clustering is the one with the lowest NDd value.

	X1	X2	X3	X4
X1		0.5		
X2	0.9		0.3	
X3	0.1	0.2		0.7
X4		0.3	0.5	

Figure 4. A Simple Example for the NDd Calculation

Task-Member Assignment

The task-member assignment model in our framework is based on the research of Chen and Lin (2004). The model is able to help establish an efficient multifunctional team for each task group because team members will be able to communicate well with each other due to their multifunctional knowledge, teamwork capability, as well as good working relationships. In each functional department, project managers first select those qualified candidate members whose knowledge ratings for that functional department are above a chosen threshold. The higher the threshold, the fewer the candidate members can be selected from each functional department. Then the mathematical model shown below will form the best team composition based on the members' teamwork capability and working relationship ratings.

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

where:

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

= the descriptor of teamwork capability and working relationship of the *i*th candidate member in department *a* and *j*th candidate member in department *b*

T_{ai} = teamwork capability rating of the *i*th candidate member in department *a*

T_{bj} = teamwork capability rating of the *j*th candidate member in department *b*

$R_{ai,bj}$ = working relationship rating between the *i*th candidate member in department *a* and *j*th candidate member in department *b*

x_{ai} = the *i*th candidate member in department *a*

x_{bj} = the *j*th candidate member in department *b*

m = number of departments

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 often can 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 equations (Hanakawa et al. 1998) shown below will be used in the simulation:

$$L_{ij}(\theta) = \begin{cases} K_{ij} e^{-E_{ij}(\theta - b_{ij})} & (b_{ij} \le \theta) \\ 0 & (b_{ij} > \theta) \end{cases} \qquad (3)$$

where:

$L_{ij}(\theta)$ = the quantity of gain to knowledge of member *i* by 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*.

E_{ij} = the required knowledge level to execute task *j*.

The Simulation Model

The simulation model for estimating the project completion time is shown in Figure 5. 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. The 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 (1998a, 1998b) proposed "rework probability" and "rework impact" in their DSM study to analyze and control the task rework in simulation. 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 based on Chen and Lin's research are described in Chapter 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 mathematical 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 to see that a member involves 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 that project managers 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 Chapter 2, the simulation records the member's learning improvement over each time unit. When each task is finished, the simulation model will check 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 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.

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

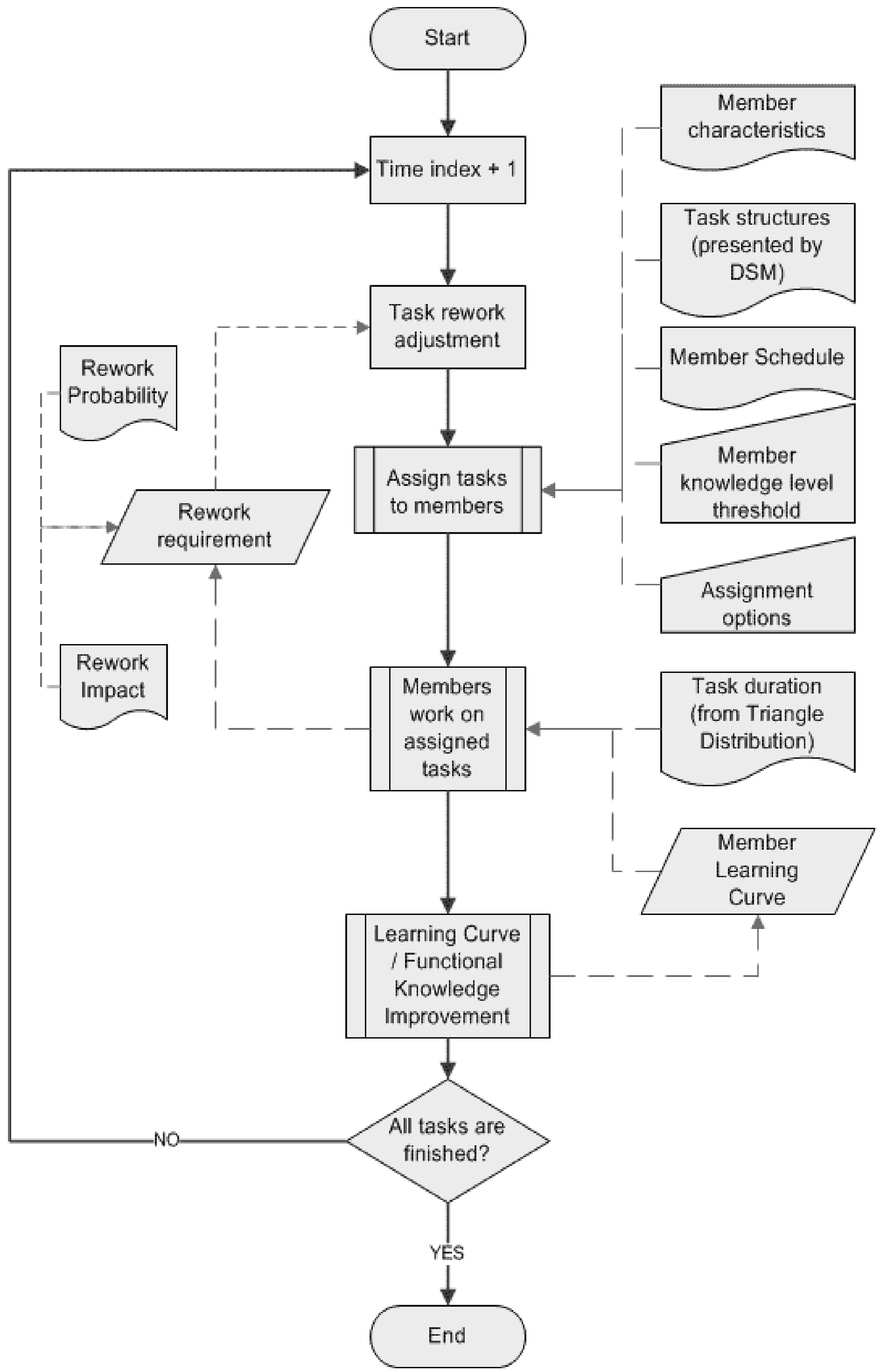


Figure 5. The Simulation Model

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 task are interdependent to each other. Factor 3 is about the 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" vs. "completing a task by rotating different available members".

Improving Task-member Assignment with Heuristics

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. After the ANOVA analysis, the major factors that delay the project completion time are identified. Heuristic rules can be developed to improve the task-member assignment performance. The heuristics will be focusing on the major factors and the reduction of project completion time. For example, if the analysis results show that the availability of members' schedule is a significant factor; the simulation will employ a heuristic in the assignment algorithm to assign tasks to the qualified members who have high availability. If the analysis shows that the learning curve is an important issue, the heuristic may give a higher priority to those members who are quick learners. If there are several significant factors, different rules will be developed and combined in the heuristics.

CHAPTER 4

AN ILLUSTRATIVE EXAMPLE

Figure 6 shows a binary DSM with 20 tasks for an engineering design project of a chemical processing system: 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. This 20-task project example is used to demonstrate the effectiveness of the research framework.

Task Clustering

Figure 7 shows the partitioned DSM for the 20-task project using Steward's partitioning algorithm. Tasks X_3 and X_7 are independent tasks so that they can be performed at the same time with affecting each other. Tasks $\{X_{12}, X_9, X_2, X_{13}, X_{15}\}$ are a set of dependent tasks, so they can be done sequentially. Two interdependent task groups found in the matrix are a 9-task group: $\{X_1, X_4, X_5, X_8, X_{10}, X_{11}, X_{17}, X_{18}, X_{19}\}$ and a 3-task group $\{X_6, X_{14}, X_{20}\}$. Due to various functional requirements among the interrelated tasks, multifunctional teams are needed for these two interdependent task groups. However, team performance usually is 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.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}	X_{18}	X_{19}	X_{20}
X_1	•			1				1			1		1		1				1	
X_2		•					1		1											
X_3			•																	
X_4	1			•	1			1							1		1			
X_5	1				•		1		1	1	1	1					1			
X_6						•		1							1		1			1
X_7							•													
X_8			1	1	1			•			1							1		
X_9			1						•			1								
X_{10}	1	1						1		•									1	
X_{11}			1		1		1				•	1						1		
X_{12}												•								
X_{13}		1					1					1	•							
X_{14}	1	1				1	1							•	1					
X_{15}		1	1									1			•					
X_{16}	1	1	1		1					1						•		1		
X_{17}				1					1				1				•		1	
X_{18}		1					1	1			1							•		
X_{19}					1					1								1	•	
X_{20}				1					1			1		1						•

Figure 6. Binary DSM (Unpartitioned)

According to the decomposition model developed by Chen and Lin (2002, 2003), the binary form of 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 7 is first transformed into a numerical DSM shown in Figure 8. The numerical values in DSM represent not only the task dependencies, but also the strengths of the related tasks. Using equation (1) in Chapter 3, a symmetric DSM is obtained and displayed in Figure 9.

Independent Tasks

Dependent Tasks

Interdependent Tasks

Interdependent Tasks

	X_3	X_7	X_{12}	X_9	X_2	X_{13}	X_{15}	X_1	X_4	X_5	X_8	X_{10}	X_{11}	X_{17}	X_{18}	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_{10}					1			1			1	•				1				
X_{11}	1	1	1							1			•		1					
X_{17}				1		1			1					•		1				
X_{18}		1			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 7. Partitioned Binary DSM

	X_1	X_4	X_5	X_8	X_{10}	X_{11}	X_{17}	X_{18}	X_{19}
X_1	1.000	0.654		0.232		0.140			0.253
X_4	0.495	1.000	0.451	0.310			0.165		
X_5	0.286		1.000		0.161	0.136	0.224		
X_8		0.137	0.351	1.000		0.089		0.118	
X_{10}	0.117			0.194	1.000				0.119
X_{11}			0.200			1.000		0.300	
X_{17}		0.160					1.000		0.485
X_{18}	0.177			0.239		0.071		1.000	
X_{19}			0.128		0.795			0.193	1.000

Figure 8. Numerical DSM

$I =$

	X_1	X_4	X_5	X_8	X_{10}	X_{11}	X_{17}	X_{18}	X_{19}
X_1	1.0000								
X_4	0.5752	1.0000							
X_5	0.1435	0.2259	1.0000						
X_8	0.1163	0.2242	0.1759	1.0000					
X_{10}	0.0585	0.0000	0.0806	0.0970	1.0000				
X_{11}	0.0705	0.0000	0.1686	0.0446	0.0000	1.0000			
X_{17}	0.0000	0.1634	0.1124	0.0000	0.0000	0.0000	1.0000		
X_{18}	0.0888	0.0000	0.0000	0.1788	0.0000	0.1862	0.0000	1.0000	
X_{19}	0.1268	0.0000	0.0644	0.0000	0.4577	0.0000	0.2428	0.0965	1.0000

Figure 9. Symmetric DSM

In order to decompose a large coupling block of tasks into clusters, the distance measures for the matrix is calculated. Based on the symmetric DSM, a squared Euclidean distance matrix D is shown in Figure 10.

$D =$

	X_1	X_4	X_5	X_8	X_{10}	X_{11}	X_{17}	X_{18}	X_{19}
X_1	0.0000								
X_4	0.4384	0.0000							
X_5	1.6273	1.4288	0.0000						
X_8	1.7124	1.4869	1.4234	0.0000					
X_{10}	2.2304	2.5405	1.9508	1.9370	0.0000				
X_{11}	2.0936	2.3516	1.5140	1.8875	2.2548	0.0000			
X_{17}	2.2138	1.8528	1.6979	2.1236	2.0867	2.1304	0.0000		
X_{18}	2.0337	2.3604	2.0745	1.4695	2.1792	1.3806	2.1353	0.0000	
X_{19}	2.0989	2.5025	2.0299	2.2606	0.6708	2.2925	1.4106	1.9733	0.0000

Figure 10. Squared Euclidean Distance Matrix

Next, the hierarchical cluster analysis using the AL method is applied to the distance matrix D to cluster the tightly coupled tasks. The AL method clusters the tasks hierarchically, beginning with each task in an individual cluster and continues to join clusters until all nine tasks have been joined into one cluster.

For an $n \times n$ matrix, there are n-1 possible clustering results. Based on the cluster results, task X1 and X4 have the strongest coupling strength, so these two tasks form the first cluster C1 = {X1, X4}. This is also obvious from the distance matrix in which the distance measure between task X1 and task X4 is the closest one (0.4384). The next cluster C2 = {X10, X19} is formed between task X10 and task X19 because they are the second closest. The similar process continues to form all the clusters as follows:

(1) C1 = {X1, X4}: C1 + {X5 + X8 + X10 + X11 + X17 + X18 + X19} (NDd = 0.0892)

(2) C2 = {X10, X19}: C1 + C2 + {X5 + X8 + X11 + X17 + X18} (NDd = 0.0784)

(3) C3 = {X11, X18}: C1 + C2 + C3 + {X5 + X8 + X17} (NDd = 0.0751)

(4) C4 = {X5, X8}: C1 + C2 + C3 + C4 + X17 (NDd = 0.0720)

(5) C5 = {X1, X5} $\in$ {C1, C4} = {X1, X4, X5, X8}: C5 + C2 + C3 + X17 (NDd = 0.0570)

(6) C6 = {X10, X17} $\in$ {C2, X17} = {X10, X17, X19}: C6 + C5 + C3

(NDd = 0.0519)

(7) C7 = {X1, X11} ∈ {C5, C3} = {X1, X4, X5, X8, X11, X18}: C7 + C6 (NDd = 0.0444*)

(8) C8 = {X1, X10}∈ {C7, C6} = {X1, X4, X5, X8, X10, X11, X17, X18, X19}.

With the numeric values, the 9-task interdependent group is then decomposed into two smaller groups: {X_1, X_4, X_5, X_8, X_{11}, X_{18}} and {X_{10}, X_{17}, X_{19}} shown in Figure 11, which is the best task clustering with the lowest NDd value (0.0444).

	X_1	X_4	X_5	X_8	X_{11}	X_{18}	X_{10}	X_{17}	X_{19}
X_1	•	0.6549		0.2325	0.1409				0.2536
X_4	0.4954	•	0.4518	0.3106				0.1658	
X_5	0.2869		•		0.1366		0.1612	0.2247	
X_8		0.1377	0.3517	•	0.0892	0.1184			
X_{11}			0.2006		•	0.3008			
X_{18}	0.1775			0.2391	0.0715	•			
X_{10}	0.1170			0.1940			•		0.1195
X_{17}		0.1609						•	0.4855
X_{19}			0.1288			0.1930	0.7959		•

Figure 11. Final Cluster Result

Task-Member Assignment

Figure 12 shows the quantitative ratings of three important team member characteristics (i.e., multifunctional knowledge, teamwork capability and working relationship) for a 30-member company. These ratings will serve as each member's weights in task-member assignments. In each functional department, project managers first select those qualified candidate members whose knowledge ratings for that functional department is above a chosen threshold, say 0.800. The higher the threshold, the fewer the candidate members can be selected from each functional department. When assigning a member to an individual task (i.e., independent task), there is no need to consider the member's teamwork capability or working relationship with the others. Therefore a qualified member (whose knowledge rating is above the chosen threshold) with the lowest teamwork capability is selected for the task. On the other hand, each member's teamwork capability and their working relationship have to be taken into account when assigning team members to each interdependent task group. The task-member assignment model shown in Chapter 3 (Chen and Lin, 2004) is implemented in our simulation model to help form the best team composition.

Learning Curve Improvement

As each task is assigned to a right member, the member's learning curve improvement with the task can be calculated by equation (2) shown in Chapter 3. This learning curve equation shows if a member's knowledge level to the task is above the required level, he/she will not gain any knowledge improvement.

Multifunctional Knowledge Rating	D_1	D_2	D_3	D_4	D_5	D_6	D_7
M_1	1.000	0.000	0.000	0.000	0.000	0.000	0.000
M_2	1.000	0.117	0.000	0.000	0.000	0.000	0.000
M_3	1.000	0.000	0.000	0.000	0.295	0.000	0.000
M_4	1.000	0.000	0.494	0.000	0.000	0.000	0.198
M_5	0.206	0.000	0.000	0.000	0.000	0.000	0.000
M_6	0.000	1.000	0.000	0.000	0.000	0.800	0.000
M_7	0.000	1.000	0.000	0.000	0.138	0.000	0.900
M_8	0.000	1.000	0.090	0.000	0.064	0.066	0.000
M_9	0.000	0.900	0.000	0.068	0.000	0.000	0.000
M_{10}	0.639	0.000	1.000	0.000	0.000	0.800	0.000
M_{11}	0.094	0.000	1.000	0.000	0.000	0.000	0.412
M_{12}	0.000	0.479	1.000	0.000	0.000	0.000	0.097
M_{13}	0.000	0.242	1.000	0.000	0.000	0.000	0.000
M_{14}	0.000	0.000	0.227	0.094	0.000	0.000	0.000
M_{15}	0.293	0.000	0.000	1.000	0.064	0.114	0.000
M_{16}	0.000	0.057	0.000	1.000	0.000	0.000	0.000
M_{17}	0.000	0.057	0.000	1.000	0.000	0.000	0.097
M_{18}	0.000	0.000	0.227	0.602	0.295	0.000	0.000
M_{19}	0.102	0.000	0.000	0.000	1.000	0.000	0.000
M_{20}	0.000	0.117	0.000	0.433	1.000	0.379	0.193
M_{21}	0.000	0.000	0.900	0.000	1.000	0.000	0.900
M_{22}	0.000	0.000	0.090	0.000	1.000	0.000	0.000
M_{23}	0.102	0.000	0.000	0.000	0.000	1.000	0.000
M_{24}	0.000	0.479	0.000	0.000	0.000	1.000	0.000
M_{25}	0.000	0.269	0.000	0.000	0.138	1.000	0.513
M_{26}	0.000	0.000	0.144	0.000	0.000	0.800	0.000
M_{27}	0.000	0.000	0.000	0.412	0.000	0.206	0.000
M_{28}	0.107	0.000	0.000	0.800	0.569	0.000	1.000
M_{29}	0.000	0.000	0.191	0.000	0.000	0.000	1.000
M_{30}	0.000	0.000	0.000	0.201	0.000	0.000	0.900

(a) Multifunctional Knowledge Rating for the 30-Member Example

Teamwork Capability Rating	T	Teamwork Capability Rating	T	Teamwork Capability Rating	T
M_1	0.9779	M_{11}	0.9352	M_{21}	0.8839
M_2	0.6758	M_{12}	0.4696	M_{22}	0.3457
M_3	0.8534	M_{13}	0.6988	M_{23}	0.7400
M_4	0.3367	M_{14}	0.3176	M_{24}	0.6785
M_5	0.5412	M_{15}	0.5564	M_{25}	0.5767
M_6	0.8824	M_{16}	0.6785	M_{26}	0.3869
M_7	0.7400	M_{17}	0.6869	M_{27}	0.3427
M_8	0.5537	M_{18}	0.4191	M_{28}	0.9779
M_9	0.3018	M_{19}	1.0000	M_{29}	0.5669
M_{10}	0.8167	M_{20}	0.7006	M_{30}	0.2925

(b) Teamwork Capability Rating for the 30-Member Example

R	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}	M_{16}	M_{17}	M_{18}	M_{19}	M_{20}	M_{21}	M_{22}	M_{23}	M_{24}	M_{25}	M_{26}	M_{27}	M_{28}	M_{29}	M_{30}
M	•																													
M_2	0.83	•																												
M_3	0.17	0.33	•																											
M_4	0.67	0.67	0.33	•																										
M_5	0.33	0.33	0.17	0.33	•																									
M_6	0.83	0.83	0.50	0.67	0.33	•																								
M_7	0.50	0.50	0.17	0.50	0.67	0.33	•																							
M_8	0.83	0.67	0.50	0.17	0.33	0.33	0.50	•																						
M_9	0.50	0.33	0.33	0.50	0.83	0.67	0.33	0.67	•																					
M_{10}	1.00	0.83	0.67	0.17	0.50	0.17	1.00	0.33	0.17	•																				
M_{11}	0.67	0.50	0.17	0.67	0.50	0.17	0.17	0.17	0.17	0.67	•																			
M_{12}	0.50	0.50	0.50	0.33	0.33	0.67	0.50	0.17	0.67	0.50	0.67	•																		
M_{13}	0.17	0.17	0.33	0.00	0.33	0.33	0.67	0.17	0.17	0.17	0.33	0.17	•																	
M_{14}	0.67	0.67	0.50	0.67	0.50	0.67	0.33	0.50	0.17	0.67	0.67	0.67	0.17	•																
M_{15}	0.33	0.33	0.17	0.50	1.00	0.33	0.83	0.17	0.33	1.00	0.67	0.17	0.33	0.67	•															
M_{16}	0.67	0.33	0.50	0.17	0.67	0.67	0.67	0.33	0.67	0.50	0.17	0.67	0.33	0.33	0.33	•														
M_{17}	0.33	0.67	0.33	0.50	0.67	0.33	0.33	0.67	0.50	0.50	0.50	0.33	0.00	0.67	0.33	0.33	•													
M_{18}	0.83	0.33	0.67	0.17	0.67	0.33	0.83	0.17	0.17	0.17	0.33	0.33	0.67	0.33	0.50	0.00	0.33	•												
M_{19}	0.50	0.00	0.17	0.67	0.33	0.33	0.50	0.50	0.67	0.67	0.50	0.83	0.67	0.50	0.67	0.33	0.50	0.33	•											
M_{20}	0.50	0.50	0.50	0.33	1.00	0.83	0.50	0.17	0.33	0.33	0.17	0.50	0.33	0.67	1.00	0.83	0.33	0.83	0.33	•										
M_{21}	0.17	0.67	0.33	0.83	0.83	0.17	0.17	0.50	0.83	0.17	0.67	0.33	0.83	0.33	0.33	0.17	0.50	0.33	0.83	0.50	•									
M_{22}	0.67	0.33	0.17	0.50	0.83	0.67	0.67	0.17	0.17	0.67	0.33	0.17	0.50	0.50	0.50	0.83	0.00	0.50	0.33	0.67	0.33	•								
M_{23}	0.33	0.83	0.50	0.17	0.67	0.33	0.33	0.50	0.83	0.33	0.83	0.50	0.17	0.67	0.50	0.17	0.83	0.17	0.50	0.83	0.50	0.50	•							
M_{24}	0.67	0.50	0.17	0.50	0.83	0.50	1.00	0.33	0.67	0.67	0.33	0.50	0.33	0.83	0.67	0.50	0.50	0.33	0.50	1.00	0.83	0.33	0.50	•						
M_{25}	0.33	0.50	0.50	0.17	0.50	0.33	0.33	0.33	0.17	0.83	0.67	0.33	0.50	0.33	0.50	0.50	0.50	0.50	0.83	0.50	0.67	0.50	0.50	0.83	•					
M_{26}	0.83	0.17	0.00	0.67	0.33	0.17	0.83	0.33	0.50	0.83	0.83	0.33	0.00	0.50	0.50	0.83	0.00	0.50	0.33	0.83	0.17	0.00	0.67	0.67	0.33	•				
M_{27}	0.50	0.67	0.50	0.33	0.50	0.17	0.50	0.33	0.17	0.83	0.67	0.33	0.17	0.83	0.67	0.67	0.83	0.67	0.50	0.50	0.67	0.50	0.33	0.50	0.17	0.17	•			
M_{28}	0.17	0.33	0.33	0.33	0.83	0.33	0.17	0.50	0.67	0.83	1.00	0.83	0.67	0.33	0.17	0.33	0.50	0.50	0.50	1.00	0.33	0.33	0.33	0.33	0.83	0.33	0.83	•		
M_{29}	0.17	0.17	0.17	0.33	0.50	0.67	0.17	0.33	0.83	0.67	0.83	0.67	0.50	0.67	0.50	0.83	0.67	0.33	0.33	0.17	0.17	0.17	0.33	0.33	0.50	0.17	0.50	0.33	•	
M_{30}	0.33	0.67	0.17	0.50	0.33	0.33	0.33	0.33	0.67	0.17	0.17	0.17	0.17	0.50	0.17	0.17	0.50	0.83	0.50	0.33	0.67	0.00	0.50	0.17	0.17	0.33	0.83	0.17	0.17	•

(c) Working Relationship Rating for the 30-Member Example

Figure 12. Ratings of Team Member Characteristics

The learning curve efficiency E_{ij}, which is a number between 0 and 1, indicates a member i's efficiency of gain to 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 gain to knowledge of 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 $K_{ij} = \theta - b_{ij}$.

We assume that the required knowledge level θ for the tasks belonging to a functional department follows a normal distribution with $\mu = 0.5$, as shown in Figure 13. The shadowed area represents the tasks that are easier to handle than task j. As b_{ij} is improved by the learning curve L_{ij}, the functional knowledge level of member i in a department where task *j* belongs to will improve accordingly. The amount of improvement will be equal to $L_{ij} \times \frac{ShadowedArea}{AllArea}$ (or $L_{ij} \times$ Cumulative Probability).

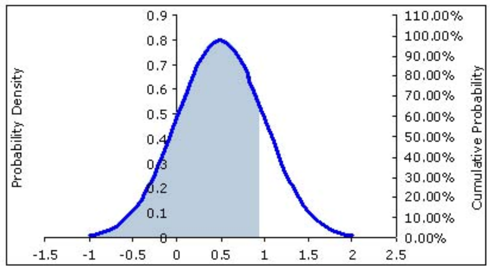


Figure 13. Normal Distribution of θ for the Tasks in a Functional Department

Simulation

Before the simulation runs, the decision between two task-member assignment options: "completing a task by the same member" vs. "completing a task by rotating different available members", has to be made by the managers. Then according to the simulation model shown in Figure 5, 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 period of times, which is estimated using triangle distribution. Task duration can be estimated by interviewing managers, experts and senior engineers from the corresponding functional areas. Task duration becomes more intuitive for experts or senior engineers, if we allow them concentrate on only estimating the three parameter values (i.e., optimistic, expected and pessimistic completion times) of triangle distribution, instead of other distributions. Figure 14 shows the detailed task information (i.e., workload, department and the estimated duration) for the 20-task example.

	Workload	Dept	Pessi	Expected	Opti
X1	0.5	1	4	6	8
X2	0.5	3	6	8	10
X3	0.25	1	3	7	11
X4	0.5	2	2	4	9
X5	1	3	4	5	7
X6	1	4	6	7	8
X7	0.75	2	3	7	9
X8	0.75	4	2	5	5
X9	0.25	3	2	4	7
X10	0.5	3	4	6	8
X11	1	5	3	7	9
X12	0.25	1	7	9	11
X13	0.5	4	4	6	8
X14	0.5	5	3	6	9
X15	0.75	5	6	8	10
X16	0.25	7	5	8	10
X17	0.75	5	6	7	8
X18	0.75	6	7	8	9
X19	0.5	7	4	7	8
X20	0.25	6	3	8	10

Figure 14. Task Information for the 20-Task Example

There are other frequently used distributions such as beta distribution and binomial distribution. Project managers needs to carefully select appropriate distributions to be used in the simulation for different engineering project scenarios.

The following shows one possible outcome of task-member assignments by simulation: $X_1 \rightarrow M_1$, $X_2 \rightarrow M_{11}$, $X_3 \rightarrow M_2$, $X_4 \rightarrow M_7$, $X_5 \rightarrow M_{10}$, $X_6 \rightarrow M_{18}$, $X_7 \rightarrow M_6$, $X_8 \rightarrow M_{16}$, $X_9 \rightarrow M_{11}$, $X_{10} \rightarrow M_{13}$, $X_{11} \rightarrow M_{22}$, $X_{12} \rightarrow M_2$, $X_{13} \rightarrow M_{17}$, $X_{14} \rightarrow M_{20}$, $X_{15} \rightarrow M_{28}$, $X_{16} \rightarrow M_7$, $X_{17} \rightarrow M_{19}$, $X_{18} \rightarrow M_{26}$, $X_{19} \rightarrow M_{25}$, and $X_{20} \rightarrow M_{26}$. 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 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. It should be noted that 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, the 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 15 shows the DSM's of rework probability and rework impact used in the 20-task example, which determine the probability and the number of iteration/rework required in the simulation. Since rework is common and important in concurrent engineering projects, it is recommended that the project manager should decide these values carefully with the opinions from the experts and senior engineers. Note that the locations of the probability and impact values in these two DSMs are identical with those in the original DSM (see Figure 6). 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 X13 is completed, the simulation identifies that tasks X1 and X10 are dependent on task X13 and knows that the two tasks both have a rework probability 0.5 with task X13 as indicated in the rework probability DSM. In the case of task X10, the simulation generates a random number between 0 and 1. If the number is not greater than 0.5, for example, rework is scheduled for task X10 and the simulation records its impact value, which is 0.6 as indicated in the rework impact DSM. So at the beginning of next iteration, the remaining work of task X10 will be set as 60%. Similar procedure will be taken for task X1 to determine the need of rework and to record the amount of rework in the next round, if necessary.

It is often that in real projects, if a task is reworked, the probability that this task is required for another cycle of rework will reduce. However, there is a scarce of previous research or case study about how the rework probability and impact will reduce. In the simulation, it is assumed that each time a task A is reworked because of the information input from another task B, the rework impact and probability of task A caused by task B will drop by 10 percent. That is, the value of the cell at row A column B in the rework probability DSM and rework impact DSM will be 10% less.

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 become 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) in simulation. The final value of time index is the estimated project completion time.

	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						

(a) Rework Probability DSM

	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						

(b) Rework Impact DSM

Figure 15. Rework Probability DSM and Rework Impact DSM

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: A) rework probability; B) rework impact; C) availability of member schedules; D) learning curve efficiency; and E) 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 in totally 72 treatment combinations. Each treatment combination will be given 50 replications in the experiment.

From the ANOVA table shown in Table 2 (using 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) It should be noted that 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.

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. It is recommended that project managers should pay more attention on clarifying

the entire project task structure and simplifying the task relations, so that the chance and the 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 project managers to 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 48.6 days, which generally is not considered as a long-term project. We expect learning curve may play a bigger role for 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 the strongest effect (F = 661.857) 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, better task performance will be

expected because having a member to stay with the same task will keep on this member's learning improvement 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.

Table 2. ANOVA Table

Source of Variance	Sum of Squares	Degree of Freedom	Mean Square	F	Sig
A	207462.172	2	103731.086	414.801	.000
B	381133.807	2	190566.903	762.041	.000
C	530615.121	1	530615.121	2121.830	.000
D	142.404	1	142.404	.569	.451
E	163216.000	1	163216.000	652.670	.000
A * B	37318.257	4	9329.564	37.307	.000
A * C	12436.584	2	6218.292	24.866	.000
A * D	223.141	2	111.570	.446	.640
A * E	7411.152	2	3705.576	14.818	.000
B * C	18550.882	2	9275.441	37.091	.000
B * D	533.376	2	266.688	1.066	.344
B * E	5165.407	2	2582.703	10.328	.000
C * D	99.334	1	99.334	.397	.529
C * E	165513.361	1	165513.361	661.857	.000
D * E	.871	1	.871	.003	.953
Error	893515.372	3573	250.074		
Total	10934560.000	3600			
Corrected Total	2423337.240	3599			

<u>The ANOVA Results with Constant Rework Probabilities and Impacts</u>

At the initial development of the simulation, the rework probabilities and rework impacts are kept as constant throughout the simulation. Later, in order to cope with the situations in real projects, non-constant rework probabilities and impacts are developed and implemented in the simulation. The logic is that if a task is reworked, the probability that this task is required to be reworked again will drop. Similarly, if this task needs to be reworked again, the amount of work in the next rework cycle will reduce.

Table 3. ANOVA Table with Constant Rework Probabilities/Impacts

Source	Sum of Squares	Degree of Freedom	Mean Square	F	Sig.
A	286435.941	2	143217.970	454.216	.000
B	442268.077	2	221134.039	701.326	.000
C	536776.022	1	536776.022	1702.385	.000
D	434.723	1	434.723	1.379	.240
E	161242.402	1	161242.402	511.380	.000
A * B	56216.353	4	14054.088	44.573	.000
A * C	9026.885	2	4513.443	14.314	.000
A * D	176.405	2	88.202	.280	.756
A * E	2888.795	2	1444.398	4.581	.010
B * C	18368.762	2	9184.381	29.128	.000
B * D	989.562	2	494.781	1.569	.208
B * E	2407.835	2	1203.918	3.818	.022
C * D	1582.714	1	1582.714	5.020	.025
C * E	162046.503	1	162046.503	513.930	.000
D * E	1.563	1	1.563	.005	.944
Error	1126596.453	3573	315.308		
Total	2807458.993	3599			

Compare the ANOVA tables with the constant (See Table 3) and non-constant (Table 2) rework probabilities/impacts, it is noted that most of the significant factors are identical. However, the interaction of C*D (schedule availability * learning curve efficiency), which is significant in constant rework probabilities/impacts situation, becomes not significant. The reason is that the reduction of rework probabilities and impacts results in the reduction of project completion time. With a longer project completion time, if a member has higher schedule availability and higher learning curve efficiency, it is more likely that the improvement of his/her knowledge level can reduce the project time. In a shorter project completion time, the member with higher schedule availability and learning curve efficiency can still gain experience to improve his/her knowledge level; however, the improvement is not significant enough to reduce the project completion time.

Developing Heuristic Rules to Improve Assignment Performance

From the analysis results in previous section, the following conclusions are made to assist the development of heuristic rules:

1) Members' schedule availability is the most important resource for task-member assignments.

2) The assignment options are closely related to members' availability.

3) The longer a member stay in the same task, the more likely the learning curve improvement can help reduce the project completion time.

4) The last task to be finished in the project is always within an interdependent task group, especially the large task group. Due to this fact, the priority should be given to the tasks in the interdependent task group (multiple-task) than the single task in the task-member assignment.

According to the analysis above, the following heuristic rules are developed and implemented in the simulation:

1) List the members in the order of members schedule ranking.

2) Always assign the qualified members with the highest schedule ranking for both single-task and multiple-task assignments.

3) Cancel the assignment options and try to let a member stay with the same task. If task rotation is needed, assign the member on top of the schedule ranking list to the task.

4) Start the task-member assignment with the interdependent task groups.

The following procedures show how to apply the heuristic rules in task-member assignment:

Step 1: Make a list of members sorting by their average schedule availability for the first 48 days in descending order, that is, the member with the highest average schedule availability from Day 1 to Day 48 is on the top of the list.

Step 2: Find the largest multiple-task group that no members has been assigned to the group tasks yet.

Step 3: Choose the members who have qualified functional knowledge to the task. The knowledge rating of the member on the top of the schedule ranking list is compared to the required knowledge threshold. If it is higher than the required level, the member is put into the qualified member pool for this task. The knowledge rating of the next member on the schedule ranking list is then compared in the same way until the number of the members in the pool reaches a predetermined limit. In this example, 5 is set for each task in the multiple-task group and 3 is used for single tasks.

Step 4: If a task is previously assigned to a member and this member is available in this period of time, the member is then set as the only member in the qualified member pool for this task.

Step 5: Use Chen and Lin's task-member assignment model to assign tasks.

Step 6: If all tasks are assigned, task-member assignment is finished. If not, go to Step 2.

Table 4 shows the comparisons of simulation results with and without applying the heuristic rules in the task-member assignment. It can be concluded that the proposed heuristics not only reduce the project completion time significantly (item 1 in Table 4) but also the variations. There are three types of variations:

1) The variations of all the estimated project time (item 2 and 3 in Table 4). The variations are cased by both different treatment levels and the probability functions used in the simulation.

2) The variations of the project completion time between treatment combinations (items 4 and 5 in Table 4). These variations are mainly caused by different treatment level (i.e., between treatments) in the simulation.

3) The variations within the same treatment combination (items 6 and 7 in Table 4), which are caused by the probability functions used in the simulation.

The comparisons show that not only the overall variations are reduced; the variations between treatments and within treatments are also dropped considerably. The heuristic rules, which aim at the significant factors and try to reduce the resource constraints of these significant factors, are the contributing reasons for the reduction of variations between different treatments. Furthermore, with the heuristic rules releasing the constraints of resources to some extent, the project completion times are reduced. Therefore the simulation model, which is divided and performed by time

Table 4. Comparison of Simulation Results with and without Heuristics

	Without Heuristics	**With Heuristics**
1. Average project completion time	48.62 days (3600 runs)	31.14 days (1800 runs)
2. Standard deviation (all simulation runs)	25.95 days	9.64 days
3. Standard deviation / Average project completion time	25.95 / 48.62 = 53.4%	9.64 / 31.34 = 31.0%
4. Standard deviation between treatment combinations	20.98 days (72 treatment combinations)	7.56 days (36 treatment combinations)
5. Standard deviation between treatment combinations / Average project completion time	20.98 / 48.62 = 43.16 %	7.56 / 31.14 = 24.3 %
6. Average of standard deviations within treatment	12.98 days	5.60 days
7. Average of Standard deviations within treatment / Average project time in this treatment	24.93%	17.36%
8. Lowest Average Project Time in Treatment Combinations (Average of 50 replications)	22.5 days	19.64 days
9. Highest Average Project Time in Treatment Combinations (Average of 50 replications)	116.64 days	47.72 days

units, has less dependence on the probability functions. This results in the lower variations within the same simulation setup (i.e., the same treatment combinations).

Before applying the heuristic rules, the lowest average project completion time of all treatments is 22.5 days, which happens when rework probability and rework impact are both in low levels, members' schedule availability is high, the learning curve efficiency is low and the member rotation is allowed. It becomes 19.6 days with the heuristic rules applied to our simulation, when rework probability and rework impact are both in low levels, members' schedule availability is high, and the learning curve efficiency is high. The results show that project managers should always pay attention on reducing the project structure complexity (i.e., the rework probability and impact) and providing enough human resources in order to shorten the project completion time. Learning curve is proven not a significant factor in this example.

It should be noted that the above heuristic rules developed to improve the task-member assignment performance is based on the analysis results in our example project. For different projects, different heuristic rules can be drawn from different simulation and ANOVA results about the significance of factors that delay the project completion time.

CHAPTER 5

CONCLUSION

In this research, 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 simplified and 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, which are often occurred in concurrent engineering projects. Each member's knowledge improvement in the simulation is modeled by 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 not only is 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 major limitation of this research is: without sufficient and dependable data as input, it is difficult to apply this research framework into real project. Garbage in garbage out, the result of the simulation and its conclusion will highly depend on the data input. Blindly believe the numerical simulations are dangerous. We suggest companies keep a database for historical project data and members' characteristic ratings.

To those enterprises with sufficient data available, the simulation model and performance can be validated and fine-tuned with minor changes in the simulation mechanism. This can be viewed as a "system learning" process.

Some of our future research extensions are summarized as follows:

1) 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.

2) This simulation model is developed for single project environment only. Although the members' schedule can reduce the conflict to other projects, the dynamic assignment is only applicable for a single project.

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 in 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 task-member assignment and the mechanism of the simulation model assume that a task is performed by one member at the same time. The limitation is that when a task is a key task which a number of other tasks are dependent on, assigning more human resources to this task can finish this task earlier so that accurate information is available for the dependent tasks and the duration of the project can be reduced.

5) The learning curved model used in this research is based on a software engineering environment. A general or solid learning curve is desirable to improve the applicability of this model to reality.

6) In the simulation, we assume a 10% constant reduction rate for both rework probability and impact. In the future research, we will try to find out the distributions (non-constant rates) of the rework probability and impact.

REFERENCES

Bandopadhyay, Sukumar and Sundararajan, Arunapuram. “Simulation of a longwall development-extraction network”, *CIM Bulletin*, v 80, n 903, Jul, 1987, p 62-70

Bailey, James; Alfares, Hesham; and Lin, Win Yuan. “Optimization and heuristic models to integrate project task and manpower scheduling”, *Computers and Industrial Engineering*, v 29, n 1-4, Sep, 1995, p 473-476

Browning, T.R. “Modeling and analyzing cost, schedule, and performance in complex system product development”, *Ph.D. Thesis (TMP), Massachusetts Institute of Technology, Cambridge, MA*, 1998a.

Browning, T.R. and Eppinger, S.D. “A model for development project cost and schedule planning,” *M.I.T. Sloan School of Management, Cambridge, MA,* Working Paper no. 4050, November 1998b.

Boctor, F.F. “Heuristics for scheduling projects with resource restrictions and several resource-duration modes”, *International Journal of Production Research*, v 31, n 11, Nov, 1993, p 2547-2558

Carmel, E. “Time-to-completion in software package startups,” *Proceedings of the IEEE International Conference on System Sciences*, 4, pp. 498-507, 1994

Chen, S.J., and Lin, L. “A project task coordination model for team organization in concurrent engineering,” *Concurrent Engineering: Research and Applications*, 10(3), pp. 187-203, 2002

Chen, S.-J., and Lin, L. “Decomposition of interdependent task group for concurrent engineering,” *Computers and Industrial Engineering*, 44: pp. 435-459, 2003

Chen, S.-J. and Lin, L. “Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering,” *IEEE Transactions on Engineering Management*, Vol. 51, No. 2, pp. 111-124, 2004

Chung, W.Y., and Guinan, P.J. “Effects of participative management,” *Proceedings of the ACM SIGCPR Conference*, pp. 252-260, 1994.

Clark, K.B., Fujimoto, T. Product Development Performance: Strategy, Organization and Management in the World Auto Industry. Harvard Business School Press, Boston, MA., 1991.

Clark, K.B., and Wheelwright, S.C. "Organizing and leading 'heavyweight' development teams," *California Management Review*, 34(3), pp. 9-28, 1992.

Duffy, M.R. and van Dorp, R.J. "Project risk analysis using simulation of activity networks: is it valid, practical, and teachable," *ASEE annual conference proceedings*, 1998.

Erbasi, Aybar and Sepil, Canan. "Modified heuristic procedure for materials management in project networks", *International Journal of Industrial Engineering : Theory Applications and Practice*, v 6, n 2, 1999, p 132-140

Everett, John G. and Farghal, Sherif, "Learning curve predictors for construction field operations", *Journal of Construction Engineering and Management*, v 120, n 3, Sept, 1994, p 603-616

Finley, E.D. and Fisher, D.J. "Project scheduling risk assessment using Monte Carlo methods", *Cost Engineering*, v 36, n 10, Oct, 1994, p 26-28

Hanakawa, N., Morisaki, S. and Matsumoto, K. "A learning curve based simulation model for software development," *Proceedings - International Conference on Software Engineering*, pp350-359, 1998.

Harold Kerzner, *Project management a systems approach to planning, scheduling, and controlling (8th edtion)*, Publisher: New York John Wiley, 2003.

Jergeas, George F. (Univ of Calgary); Revay, Stephen O. "Integrated value management approach", *AACE International. Transactions of the Annual Meeting*, 1999, p PM.12.1-PM.12.4

Johnson, D., and Johnson, F. *Joining together: group theory and group skills*. Prentice Hall, NJ. 1991.

Kandathil, George Mathew. "Prediction of spacecraft project duration: Adaptation of Markoff chain analysis", *Journal of Spacecraft Technology*, v 13, n 1, January, 2003, p 1-10

Khattab, M. and Choobineh, F. "New heuristic for project scheduling with a single resource constraint", *Computers & Industrial Engineering*, v 19, n 1-4, 1990, p 514-518

Kibler, Bruce E. "Integrated cost and schedule control: a piece of cake", *Cost Engineering*, v 34, n 7, Jul, 1992, p 15-22
Koufteros, Xenophon., Vonderembse, Mark,. and Doll, William. "Concurrent engineering and its consequences", *Journal of Operations Management*, v 19, n 1, Jan, 2001, p 97-115

Lanigan, M.J. "Task estimating: completion time versus team Size," *Engineering Management Journal*, 4(5), pp. 212-218, 1994

Lee, Jae-Kwan and Kim, Yeong-Dae. "Search heuristics for resource constrained project scheduling", *Journal of the Operational Research Society*, v 47, n5, May, 1996, p 678-689

Loch, Christoph; Mihm, Jurgen; and Huchzermeier, Arnd. "Concurrent Engineering and Design Oscillations in Complex Engineering Projects", *Concurrent Engineering Research and Applications*, v 11, n 3, September, 2003, p 187-200

Markevicius, V. and Rouphail, Nagui M. "Interactive model for project planning and scheduling", *Computing in Civil Engineering*, 1986, p 913-926

Mehrez, Abraham and David, Israel. "Two-phase simple model to determine the timing and duration of R&D project tasks", *Production Planning and Control*, v 10, n 1, Jan-Feb, 1999, p 48-53

Moselhi, Osama. Li, Ji. and Alkass, Sabah. "Web-based integrated project control system", *Construction Management and Economics*, v 22, n 1, January, 2004, p 35-46

Neumann, K. and Zhan, J. "Heuristics for the minimum project-duration problem with minimal and maximal time lags under fixed resource constraints", *Journal of Intelligent Manufacturing*, v 6, n 2, Apr, 1995, p 145-154

O'Grady, P. and Young, R.E. "Issues in concurrent engineering systems", *Journal of Design and Manufacturing*, vol. 1, pp. 27–34, 1991.

Orczyk, Joseph J. and Hancher, Donn. "Schedule risk analysis using computer simulation", *Transactions of the American Association of Cost Engineers*, 1987, p a. 7. 1-a. 7. 6

Pinto, J. K. and Covin, J. G., "Critical factors in project implementation: A comparison of construction and R&D projects", *Technovation*, vol. 9, pp. 49–62, 1989.

Ramani, S. “Simulation approach to time-cost trade-off in project network”, *Modeling and Simulation on Microcomputers*, 1986, p 115-120

Riggs, Leland S. “Risk management in CPM networks”, *Microcomputers in Civil Engineering*, v 4, n 3, Sep, 1989, p 229-235

Riggs, Jeffrey L., Brown, Sheila B. and Trueblood, Robert P., “Integration of technical, cost, and schedule risks in project management”, *Computers and Operations Research*, v 21, n 5, May, 1994, p 521-533

Schmidt, Craig W. and Grossmann, Ignacio E. “Exact overall time distribution of a project with uncertain task durations”, *European Journal of Operational Research*, v 126, n 3, Nov, 2000, p 614-636

Selle, Thomas and Zimmermann, Jurgen. “A bidirectional heuristic for maximizing the net present value of large-scale projects subject to limited resources”, *Naval Research Logistics*, v 50, n 2, March, 2003, p 130-148

Steward, D.V. “The design structure system: a method for managing the design of complex systems,” *IEEE Transactions on Engineering Management*, 28(1), pp.71-74, 1981.

Turino, Jon “Concurrent engineering: Making it work calls for input from everyone”, *IEEE Spectrum*, v 28, n 7, Jul, 1991, p 30-32

Valls, Vicente; Quintanilla, Sacramento; and Ballestin, Francisco. “Resource-constrained project scheduling: A critical activity reordering heuristic”, *European Journal of Operational Research*, v 149, n 2, Sep 1, 2003, p 282-301

Wang, Wei-Chih and Demsetz, Laura A. “Model for evaluating networks under correlated uncertainty – NETCOR”, *Journal of Construction Engineering and Management*, v 126, n 6, Nov, 2000, p 458-466

Wilson, B. G., Jonassen, D. H., and Cole, P. “Cognitive approaches to instructional design”, *The ASTD handbook of instructional technology,* New York: McGraw-Hill,1993, pp. 21.1-21.22

Wrights, T.P. “Factors affecting the cost of planes”, *J. of the Aeronautical Science*, No. 3. 1936.

Yan, Hong Sen and Jiang, Jian. “Agile concurrent engineering”, *Integrated Manufacturing Systems*, v 10, n 2, 1999, p 103-112

APPENDICES

APPENDIX A

ONE SAMPLE OF SIMULATION RESUTLS FOR TASK-MEMBER ASSIGNMENT AND TASK EVOLUTION

Table 5 shows the task-member assignment results at each time period (i.e. each day) when rework probability and rework impact are both in low levels, members' schedule availability is high and the learning curve efficiency is high. According to Table 5, task X1 is assigned to member M10 and task X2 is assigned to member M12 at the first day. The whole project is completed in 18 days.

Table 5. One Sample of Simulation Results for Task-Member Assignment

Time	Assignment						
1	X1--M10	X2--M12	X3--M2	X4--M1	X5--M14	X6--M18	X7--M6
	X8--M16	X9--M12	X10--M21	X11--M22	X12--M2	X13--M28	X14--M1
	X15--M20	X16--M7	X17--M19	X18--M27	X19--M25	X20--M26	
2	X1--M10	X2--M12	X3--M2	X4--M1	X5--M14	X6--M18	X7--M6
	X8--M16	X9--M12	X10--N/A	X11--M22	X12--M2	X13--M28	X14--M28
	X15--M19	X16--M7	X17--N/A	X18--M26	X19--N/A	X20--M26	
3	X1--M10	X2--M12	X3--M2	X4--M1	X5--M14	X6--M18	X7--M6
	X8--M16	X9--M12	X10--N/A	X11--M22	X12--M2	X13--M28	X14--M1
	X15--M20	X16--M7	X17--N/A	X18--M27	X19--N/A	X20--M26	
4	X1--N/A	X2--M14	X3--M2	X4--N/A	X5--N/A	X6--M18	X7--M6
	X8--N/A	X9--M12	X10--M10	X11--N/A	X12--M2	X13--M17	X14--M1
	X15--M19	X16--M7	X17--M22	X18--N/A	X19--M25	X20--M26	
5	X1--M10	X2--M12	X3--M2	X4--N/A	X5--M11	X6--M15	X7--M6
	X8--M17	X9--N/A	X10--M21	X11--M22	X12--M2	X13--M28	X14--M1
	X15--M20	X16--M7	X17--M19	X18--M27	X19--M25	X20--M26	
6	X1--M10	X2--M12	X3--M2	X4--M1	X5--M14	X6--M5	X7--M6
	X8--N/A	X9--N/A	X10--M21	X11--M22	X12--M2	X13--M16	X14--M21
	X15--M20	X16--M7	X17--M19	X18--M26	X19--M25	X20--M26	
7	X1--M10	X2--M12	X3--M2	X4--N/A	X5--M11	X6--M18	X7--M6
	X8--M16	X9--N/A	X10--M21	X11--M22	X12--M2	X13--M28	X14--M19
	X15--M20	X16--M7	X17--M1	X18--M27	X19--M25	X20--M26	
8	X1--M10	X2--M12	X3--M2	X4--N/A	X5--M10	X6--M5	X7--N/A
	X8--N/A	X9--N/A	X10--M21	X11--M1	X12--M2	X13--M16	X14--M20
	X15--M22	X16--M7	X17--M19	X18--M27	X19--M25	X20--M26	
9	X1--M10	X2--M12	X3--N/A	X4--N/A	X5--N/A	X6--M18	X7--N/A
	X8--M16	X9--N/A	X10--M21	X11--M22	X12--N/A	X13--M28	X14--M1
	X15--M20	X16--M7	X17--M21	X18--M27	X19--M25	X20--M26	

Table 5. One Sample of Simulation Results for Task-Member Assignment (Continued)

10	X1--M10	X2--M11	X3--N/A	X4--N/A	X5--N/A	X6--M18	X7--N/A
	X8--M15	X9--N/A	X10--M21	X11--N/A	X12--N/A	X13--M28	X14--M1
	X15--M20	X16--M7	X17--M22	X18--M27	X19--M25	X20--N/A	
11	X1--M10	X2--M12	X3--N/A	X4--N/A	X5--N/A	X6--M15	X7--N/A
	X8--M16	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--M28	X14--N/A
	X15--M20	X16--N/A	X17--M19	X18--M27	X19--M25	X20--M26	
12	X1--M10	X2--N/A	X3--N/A	X4--N/A	X5--N/A	X6--M15	X7--N/A
	X8--N/A	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--M28	X14--N/A
	X15--N/A	X16--N/A	X17--M19	X18--M27	X19--M25	X20--N/A	
13	X1--M10	X2--N/A	X3--N/A	X4--M1	X5--N/A	X6--N/A	X7--N/A
	X8--M16	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--M28	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--M25	X20--N/A	
14	X1--M10	X2--N/A	X3--N/A	X4--M1	X5--N/A	X6--M15	X7--N/A
	X8--M28	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--N/A	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--M25	X20--N/A	
15	X1--M5	X2--N/A	X3--N/A	X4--N/A	X5--N/A	X6--N/A	X7--N/A
	X8--M16	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--N/A	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--M25	X20--N/A	
16	X1--M10	X2--N/A	X3--N/A	X4--M1	X5--N/A	X6--N/A	X7--N/A
	X8--N/A	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--N/A	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--N/A	X20--N/A	
17	X1--M10	X2--N/A	X3--N/A	X4--N/A	X5--N/A	X6--N/A	X7--N/A
	X8--M15	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--N/A	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--N/A	X20--N/A	
18	X1--M10	X2--N/A	X3--N/A	X4--M1	X5--N/A	X6--N/A	X7--N/A
	X8--N/A	X9--N/A	X10--N/A	X11--N/A	X12--N/A	X13--N/A	X14--N/A
	X15--N/A	X16--N/A	X17--N/A	X18--N/A	X19--N/A	X20--N/A	

Table 6 shows the task evolution at each time period (i.e., each day) with the assignment results shown in Table 6. According to Table 6, all the tasks are 100% at the beginning of Day 1. At the start of Day 2, task X1 has 87.1% left and task X2 has 86.6% left. It can also be found that Tasks X1 and X8 are the last tasks to be finished in this project.

Task X4 has completed in 3 days, so at the beginning of Day 4, there are 0% left by task X4. However, at the end of Day 5, task X8 (one of X4's dependent tasks) is finished and generates information input that requires task X4 to rework. X4 then has a 10% rework requirement at the beginning of Day 6 and a member is assigned to X4 again to handle the rework.

Table 6. One Sample of Simulation Results for Task Evolution

Time	Task Performing									
	X1	**X2**	**X3**	**X4**	**X5**	**X6**	**X7**	**X8**	**X9**	**X10**
1	1	1	1	1	1	1	1	1	1	1
2	0.871	0.866	0.873	0.656	0.818	0.863	0.843	0.671	0.732	0.82
3	0.732	0.732	0.747	0.312	0.636	0.727	0.687	0.342	0.465	0.82
4	0.685	0.598	0.62	0	0.454	0.59	0.53	0.012	0.197	0.82
5	0.685	0.731	0.493	0	0.454	0.453	0.374	0.012	0	0.679
6	0.56	0.662	0.366	0.1	0.296	0.389	0.217	0	0	0.615
7	0.43	0.523	0.24	0	0.177	0.346	0.06	0.2	0	0.488
8	0.498	0.384	0.113	0	0.074	0.273	0	0	0	0.36
9	0.363	0.246	0	0	0	0.225	0	0.5	0	0.233
10	1	0.107	0	0	0	0.155	0	0.179	0	0.105
11	0.861	0.041	0	0	0	0.015	0	0.056	0	0
12	0.821	0	0	0	0	0.2	0	0	0	0
13	0.68	0	0	0.2	0	0.067	0	0.2	0	0
14	0.538	0	0	0.09	0	0.067	0	0.18	0	0
15	0.394	0	0	0	0	0	0	0.093	0	0
16	0.284	0	0	0.081	0	0	0	0	0	0
17	0.212	0	0	0	0	0	0	0.162	0	0
18	0	0	0	0	0	0	0	0	0	0
	X11	**X12**	**X13**	**X14**	**X15**	**X16**	**X17**	**X18**	**X19**	**X20**
1	1	1	1	1	1	1	1	1	1	1
2	0.845	0.868	0.883	0.867	0.873	0.894	0.844	0.886	0.929	0.886
3	0.69	0.737	0.764	0.807	0.814	0.789	0.844	0.818	0.929	0.773
4	0.535	0.605	0.643	0.711	0.739	0.683	0.844	0.755	0.929	0.659
5	0.535	0.473	0.572	0.521	0.674	0.577	0.7	0.755	0.85	0.546
6	0.397	0.341	0.505	0.33	0.607	0.471	0.627	0.629	0.765	0.432
7	0.258	0.21	0.422	0.259	0.472	0.366	0.481	0.567	0.675	0.319
8	0.119	0.078	0.419	0.184	0.336	0.26	0.406	0.511	0.58	0.205
9	0.049	0	0.428	0.11	0.256	0.154	0.334	0.398	0.481	0.092

Table 6. One Sample of Simulation Results for Task Evolution (Continued)

10	0	0	0.359	0.018	0.195	0.048	0.256	0.285	0.379	0
11	0	0	0.221	0	0.074	0	0.188	0.172	0.473	0.1
12	0	0	0.281	0	0	0	0.115	0.059	0.365	0
13	0	0	0.14	0	0	0	0	0	0.254	0
14	0	0	0	0	0	0	0	0	0.141	0
15	0	0	0	0	0	0	0	0	0.027	0
16	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0

APPENDIX B

SIMULATION PROGRAM CODE

The simulation is coded in Visual Basic using Microsoft Excel Macro. The following list showed some primary functions and subroutines used in this study.

’========CODE START======

```
Const Member_Num = 30 '30 members in this example
Const Dept_Num = 7 '7 Departments
Const Rs = 0.8 'Threshold for single task
Const Rm = 0.5 'Threshold for multiple task group


Sub cal_weight()
    'This sub calculate working relationships
    For i = 1 To Member_Num
        For j = 1 To i
            Sheet1.Cells(i, j + 1) = Sheet1.Cells(i, 1) + Sheet1.Cells(j,
1)
            If i = j Then Sheet1.Cells(i, j + 1) = 0
        Next
    Next
    alpha = CDbl(InputBox("Please input the ALPHA value:", "Input needed",
0.5))
    For i = 1 To Member_Num
        For j = 1 To i
            Sheet3.Cells(i, j) = Sheet1.Cells(i, j + 1) * alpha +
              Sheet2.Cells(i + 1, j + 1) * (1 - alpha)
            Sheet3.Cells(j, i) = Sheet3.Cells(i, j)
        Next
    Next
End Sub


Function GetJobTime(ByVal job_no As Integer) As Double
    'This function returns the required work time.
    'Input: task ID
    Dim min, likely, max As Double
    min = Sheet5.Cells(job_no + 1, 4)
    likely = Sheet5.Cells(job_no + 1, 5)
    max = Sheet5.Cells(job_no + 1, 6)
    GetJobTime = TriangSim(min, max, likely)
```

```
End Function


Function TriangSim(ByVal min As Double, ByVal max As Double, ByVal likely
As Double) As Double
    'Calculate random scenarios drawn from the triagular distribution
using uniform numbers to generate
     Dim c As Double
     Dim X As Double
     unif = Rnd()
     If min = max Then
     c = 0
     Else
     c = (likely - min) / (max - min)
     End If
     If unif <= c Then
     X = Sqr(c * unif)
     Else
     X = 1 - Sqr((1 - c) * (1 - unif))
     End If
     TriangSim = min + ((max - min) * X)
End Function


Function SimRun(ByVal Eij As Double) As Integer
    'The main simulation function, it returns the time needed by the

    On Error GoTo errorhandle
    'Error handler

    Dim AllJobDone, Overload As Boolean
    Dim JobLeft(20) As Double
    Dim tempi(10) As Integer
    Dim depts(10) As Integer
    Dim wl(10) As Double
    Dim Candidate(10, Member_Num), ChangeLog As Integer
    Dim Candi_Num(10) As Integer
    Dim av_wl(Member_Num), TDistTime(20), AvgAva(30) As Double
    Dim Assign(20)
    Dim TimeCounter, TimeStep, MaxRework, Load, TransitionTime, tempd As
Double
    Dim i, j, k, p, q, m, n, v As Integer
    Dim ReworkJob(20) As Integer
```

```
    Dim MemberInCharge(20) As Integer
    Dim AvgAvaList(30) As Integer

    TransitionTime = 0.5

    For i = 1 To 20
        JobLeft(i) = 1# 'initialize, all jobs are 100% undone
        ReworkJob(i) = 0 'initialize, no jobs require rework now
        MemberInCharge(i) = -1 'initialize, no job has fixed member
        TDistTime(i) = -1#
        For j = 1 To Member_Num
            Sheet13.Cells(j + 1, i + 1) = 0#
        Next
        For j = 1 To 310 'Empty The Cells
            Sheet8.Cells(j + 1, i + 1) = Null
            Sheet7.Cells(j + 1, i + 1) = Null
        Next
    Next

    'Copy the initial mulitfunctional knowledge level from Sheet12(KL B4
Run) to Sheet4
    For i = 2 To 31
        For j = 2 To 8
            Sheet4.Cells(i, j) = Sheet12.Cells(i, j)
        Next
    Next

    Randomize
    TimeCounter = 0#
    TimeStep = 1#
    q = 0

    For i = 1 To Member_Num
            av_wl(i) = CDbl(Sheet11.Cells(q + 2, i + 1)) 'Available
workload from Sheet11(Schedule)
    Next

    AllJobDone = False

    Do While Not AllJobDone

        TimeCounter = TimeCounter + TimeStep
        q = q + 1
```

```
        If q > 300 Then Exit Function

        For i = 1 To Member_Num 'get member schedule
            av_wl(i) = CDbl(Sheet11.Cells(q + 1, i + 1))
              'Available workload from Sheet11(Schedule)
        Next

        '--------------------------------------------
        For i = 1 To 20
          'This loop adds rework impact to jobs (which require rework)
            If ReworkJob(i) > 0 Then
                If JobLeft(i) < 0.0001 Then
                    JobLeft(i) = 0
                End If
                JobLeft(i) = JobLeft(i) + CSng(Sheet10.Cells(i + 1,
                     ReworkJob(i) + 1))
                If JobLeft(i) > 1 Then JobLeft(i) = 1
                Sheet10.Cells(i + 1, ReworkJob(i) + 1) = Sheet10.Cells
                  (i + 1, ReworkJob(i) + 1) * 0.9 'Reduce Rework
                Sheet9.Cells(i + 1, ReworkJob(i) + 1) = Sheet9.Cells
                  (i + 1, ReworkJob(i) + 1) * 0.9
                ReworkJob(i) = 0
            End If
            Sheet8.Cells(q + 1, i + 1) = JobLeft(i)
        Next


'*******************************************************************
        'Get Average Available Time -- Heuristic#1
        'Sort members by average available time
        For i = 1 To 30
            AvgAva(i) = Sheet11.Cells(308, i + 1)
            AvgAvaList(i) = i
        Next
        For i = 1 To 30
            For j = i + 1 To 30
                If AvgAva(i) < AvgAva(j) Then
                    tempd = AvgAva(i)
                    AvgAva(i) = AvgAva(j)
                    AvgAva(j) = tempd
                    tempi(1) = AvgAvaList(i)
                    AvgAvaList(i) = AvgAvaList(j)
                    AvgAvaList(j) = tempi(1)
```

```
        End If
    Next
Next


'*********************************************************
'*****************Start Assignment***********************
'********************************************************
For i = 1 To 20
    Assign(i) = -1
Next
Block_Num = CInt(Sheet6.Cells(1, 2))
For i = 1 To Block_Num
    j = 1
    p = 0
    Do While (VarType(Sheet6.Cells(i + 1, j + p + 1)) > 1)
        If JobLeft(CInt(Sheet6.Cells(i + 1, j + p + 1)))
         < 0.0001 Then
            p = p + 1
        Else
            tempi(j) = CInt(Sheet6.Cells(i + 1, j + p + 1))
            j = j + 1
        End If
    Loop
    j = j - 1
    If j < 1 Then 'jobs in this block are all done
        GoTo nextloop1
        'assign next task or task group
    End If


    If j > 1 Then '(Multiple Task GROUP)

        For k = 1 To j
            depts(k) = Sheet5.Cells(tempi(k) + 1, 3)
            wl(k) = Sheet5.Cells(tempi(k) + 1, 2)
            n = 1
            For m = 1 To Member_Num
                If Sheet4.Cells(AvgAvaList(m) + 1, depts(k) + 1)
                >= Rm And av_wl(AvgAvaList(m)) >= wl(k) Then
                        Candidate(k, n) = AvgAvaList(m)
                        'The k-th task in this group has
                           member-m as the n-th candidate
```

```
                n = n + 1
                If n > 5 Then
                    'Only top 5 members (multiple task
                      group) in time availability
                      ranking are eligible to be chosen
                    Exit For
                End If
            'End If
        End If
    Next
    Candi_Num(k) = n - 1
Next

For k = 1 To j
        If MemberInCharge(tempi(k)) > 0 Then
            If av_wl(MemberInCharge(tempi(k)))
               >= wl(k) Then
                Candidate(k, 1) =
                 MemberInCharge(tempi(k))
                Candi_Num(k) = 1
            End If
        End If
Next

For k = j + 1 To 7 'No Loop for empty tasks
    Candi_Num(k) = 1
Next

cur_high = 0
'the max value using Chen & Lin's member assignment model
tmpdbl = 0
For g1 = 1 To Candi_Num(1)
    For g2 = 1 To Candi_Num(2)
        If Candidate(1, g1) > 0 And
            Candidate(2, g2) > 0 Then
            W12 = Sheet3.Cells
               (Candidate(1, g1), Candidate(2, g2))
        Else
            W12 = 0
        End If
        For g3 = 1 To Candi_Num(3)
            If Candi_Num(3) >= 2 Then
                If Candidate(1, g1) > 0 Then
```

```
                W13 = Sheet3.Cells(Candidate
               (1, g1), Candidate(3, g3))
            Else
                W13 = 0
            End If
            If Candidate(2, g2) > 0 Then
                W23 = Sheet3.Cells(Candidate
                   (2, g2), Candidate(3, g3))
            Else
                W23 = 0
            End If
        Else
            W13 = 0
            W23 = 0
        End If
        For g4 = 1 To Candi_Num(4)
            If Candi_Num(4) >= 2 Then
                If Candidate(1, g1) > 0 Then
                    W14 = Sheet3.Cells(Candidate
                      (1, g1), Candidate(4, g4))
                Else
                    W14 = 0
                End If
                If Candidate(2, g2) > 0 Then
                    W24 = Sheet3.Cells(Candidate
                     (2, g2), Candidate(4, g4))
                Else
                    W24 = 0
                End If
                W34 = Sheet3.Cells(Candidate
                  (3, g3), Candidate(4, g4))
            Else
                W14 = 0
                W24 = 0
                W34 = 0
            End If
            For g5 = 1 To Candi_Num(5)
                If Candi_Num(5) >= 2 Then
                    If Candidate(1, g1) > 0 Then
                        W15 = Sheet3.Cells
                       (Candidate(1, g1),
                       Candidate(5, g5))
                    Else
```

```
                    W15 = 0
                End If
                If Candidate(2, g2) > 0 Then
                    W25 = Sheet3.Cells
                    (Candidate(2, g2),
                    Candidate(5, g5))
                Else
                    W25 = 0
                End If
                W35 = Sheet3.Cells(Candidate
                 (3, g3), Candidate(5, g5))
                W45 = Sheet3.Cells(Candidate
                 (4, g4), Candidate(5, g5))
            Else
                W15 = 0
                W25 = 0
                W35 = 0
                W45 = 0
            End If
            For g6 = 1 To Candi_Num(6)
                If Candi_Num(6) >= 2 Then
                    If  Candidate(1,  g1)  >  0
Then
                        W16 = Sheet3.Cells
                        (Candidate(1, g1),
                        Candidate(6, g6))
                    Else
                        W16 = 0
                    End If
                    If  Candidate(2,  g2)  >  0
Then
                        W26 = Sheet3.Cells
                        (Candidate(2, g2),
                        Candidate(6, g6))
                    Else
                        W26 = 0
                    End If
                    W36 =
Sheet3.Cells(Candidate(3, g3), Candidate(6, g6))
                    W46 =
Sheet3.Cells(Candidate(4, g4), Candidate(6, g6))
                    W56 =
Sheet3.Cells(Candidate(5, g5), Candidate(6, g6))
```

```
                                        Else
                                            W16 = 0
                                            W26 = 0
                                            W36 = 0
                                            W46 = 0
                                            W56 = 0
                                        End If
                                        For g7 = 1 To Candi_Num(7)
                                            If Candi_Num(7) >= 2 Then
                                                If Candidate(1, g1) > 0
Then
                                                    W17 =
Sheet3.Cells(Candidate(1, g1), Candidate(7, g7))
                                                Else
                                                    W17 = 0
                                                End If
                                                If Candidate(2, g2) > 0
Then
                                                    W27 =
Sheet3.Cells(Candidate(2, g2), Candidate(7, g7))
                                                Else
                                                    W27 = 0
                                                End If
                                                W37 =
Sheet3.Cells(Candidate(3, g3), Candidate(7, g7))
                                                W47 =
Sheet3.Cells(Candidate(4, g4), Candidate(7, g7))
                                                W57 =
Sheet3.Cells(Candidate(5, g5), Candidate(7, g7))
                                                W67 =
Sheet3.Cells(Candidate(6, g6), Candidate(7, g7))
                                            Else
                                                W17 = 0
                                                W27 = 0
                                                W37 = 0
                                                W47 = 0
                                                W57 = 0
                                                W67 = 0
                                            End If

                                            Overload = False
                                            'even if it's in one task
                                           group, a member can be
```

```
                                        overloaded

                                            For n1 = 1 To
                                            Member_Num
                                                Load = 0
                                                If Candidate(1, g1)
= n1 And MemberInCharge(tempi(1)) < 0 Then Load = Load +
Sheet5.Cells(tempi(1) + 1, 2)
                                                If Candidate(2, g2)
= n1 And MemberInCharge(tempi(2)) < 0 Then Load = Load +
Sheet5.Cells(tempi(2) + 1, 2)
                                                If Candidate(3, g3)
= n1 And MemberInCharge(tempi(3)) < 0 Then Load = Load +
Sheet5.Cells(tempi(3) + 1, 2)
                                                If Candidate(4, g4)
= n1 And MemberInCharge(tempi(4)) < 0 Then Load = Load +
Sheet5.Cells(tempi(4) + 1, 2)
                                                If Candidate(5, g5)
= n1 And MemberInCharge(tempi(5)) < 0 Then Load = Load +
Sheet5.Cells(tempi(5) + 1, 2)
                                                If Candidate(6, g6)
= n1 And MemberInCharge(tempi(6)) < 0 Then Load = Load +
Sheet5.Cells(tempi(6) + 1, 2)
                                                If Candidate(7, g7)
= n1 And MemberInCharge(tempi(7)) < 0 Then Load = Load +
Sheet5.Cells(tempi(7) + 1, 2)

                                                If Load > 1 Then
                                                    Overload    =
True
                                                    'if any of the
candidate is overloaded, this assignment cannot be used
                                                    Exit For
                                                End If
                                            Next

                                        If Overload Then
                                            tmpdbl = -1
                                        Else
                                            tmpdbl = W12 + W13 + W14
+ W15 + W16 + W17 + W23 + W24 + W25 + W26 + W27 + W34 + W35 + W36 + W37
+ W45 + W46 + W47 + W56 + W57 + W67
                                        End If
```

```
                                            If tmpdbl >= cur_high Then
                                                cur_high = tmpdbl
                                                Assign(tempi(1)) =
Candidate(1, g1)
                                                Assign(tempi(2)) =
Candidate(2, g2)
                                                If   j   >=   3   Then
Assign(tempi(3)) = Candidate(3, g3)
                                                If   j   >=   4   Then
Assign(tempi(4)) = Candidate(4, g4)
                                                If   j   >=   5   Then
Assign(tempi(5)) = Candidate(5, g5)
                                                If   j   >=   6   Then
Assign(tempi(6)) = Candidate(6, g6)
                                                If   j   >=   7   Then
Assign(tempi(7)) = Candidate(7, g7)
                                            End If
                                        Next
                                    Next
                                Next
                            Next
                        Next
                    Next
                Next

                For k = 1 To j 'record the assignment
                   If Assign(tempi(k)) > 0 Then
                     av_wl(Assign(tempi(k))) =
                     av_wl(Assign(tempi(k))) - wl(k)
                     If MemberInCharge(tempi(k)) < 0 Then
                        MemberInCharge(tempi(k)) = Assign(tempi(k))
                     End If
                   End If
                Next

            Else 'Single Task
                dept = Sheet5.Cells(tempi(1) + 1, 3)
                sel = 0 'member selected to the task

                    If MemberInCharge(tempi(1)) > 0 Then
                    'previously assinged
                        If av_wl(MemberInCharge(tempi(1))) >=
```

```
            Sheet5.Cells(tempi(1) + 1, 2) Then
            'the member is available
                sel = MemberInCharge(tempi(1))
                GoTo fixmember0
                 ' No change, no need to compare
            Else 'not available
                sel = -1
            End If
        End If

    cur_low = 1
    ChangeLog = 0
    'For k = 1 To Member_Num
    For k = 1 To Member_Num
        If ChangeLog >= 3 Then
            'Only top 3 members (single task) in time
            availability ranking are eligible to be chosen
            Exit For
        End If
        If Sheet4.Cells(AvgAvaList(k) + 1, dept + 1)
         >= Rs Then
            If Sheet1.Cells(AvgAvaList(k), 1)
             <= cur_low Then
                Load = 0
                For n = 1 To 20
                    If MemberInCharge(n) = AvgAvaList(k)
                        Then Load =
                       Load + Sheet5.Cells(n + 1, 2)
                    End if
                Next
                If av_wl(k) >= Sheet5.Cells(tempi(1) + 1, 2)
                 And Load <= (1 - Sheet5.Cells
                 (tempi(1) + 1, 2)) Then
                    cur_low =
                   Sheet1.Cells(AvgAvaList(k), 1)
                    sel = AvgAvaList(k)
                    ChangeLog = ChangeLog + 1
                Else
                    GoTo nextloop0
                End If
            End If
        End If
nextloop0:
```

```
                    If k = Member_Num And sel = 0 Then sel = -1
                    'the task is not assigned
                Next
fixmember0:
                If sel > 0 Then
                    av_wl(sel) = av_wl(sel) –
                     Sheet5.Cells(tempi(1) + 1, 2)
                    MemberInCharge(tempi(1)) = sel
                    Assign(tempi(1)) = sel
                End If

            End If

nextloop1: 'assign next task or task group
        Next i


        'display the assignment result
        For i = 1 To 20
            If Assign(i) = -1 Then
                Sheet7.Cells(q + 1, i + 1) = "X" & i & "--N/A"
            Else
                Sheet7.Cells(q + 1, i + 1) = "X" & i & "--M" & Assign(i)
            End If
        Next


        '********************************************************
        '****************End of Assignment*************************
        '**********************************************************

        '**********************************************************
        '*************Start Performing Tasks************************
        '**********************************************************

        For i = 1 To 20 'Do the job in this loop

            If JobLeft(i) > 0 And Assign(i) > 0 Then

                'recording learning curve improvement in sheet13
                If Sheet13.Cells(Assign(i) + 1, i + 1) <= 0 Then
                    Sheet13.Cells(Assign(i)   +   1,   i   +   1)   =
Sheet4.Cells(Assign(i) + 1, Sheet5.Cells(i + 1, 3) + 1)
```

```
                End If

                If q > 1 Then 'No assignment in time period 0
                        tmpint = CheckAssign(q - 1, i)
                        If tmpint <> Assign(i) And tmpint > 0 Then
                            tmpdbl = TransitionTime
                        Else
                            tmpdbl = 1
                        End If
                Else
                        tmpdbl = 1
                End If

                If TDistTime(i) > 0 And CheckAssign(q - 1, i) = Assign(i)
Then
                    'check if task is previously assigned
                Else
                    TDistTime(i) = GetJobTime(i)
                    'if never generated a triagular distribution or
                    'assign to a different member last time
                    'generate new triagular distribution
                End If

                JobLeft(i)  =  JobLeft(i)  -  tmpdbl  *  TimeStep  *
Sheet13.Cells(Assign(i) + 1, i + 1) / TDistTime(i) 'inch on
                'Performing job

                Call LCImprove(Assign(i), i, TDistTime(i), Eij)
                'Learning Curve Improvement

                If JobLeft(i) <= 0 Then
                    JobLeft(i) = 0
                    MaxRework = 0
                    For j = 1 To 20
                            If j <> i And VarType(Sheet9.Cells(j + 1, i
+ 1)) > 1 Then
                                If Rnd(1) < CSng(Sheet9.Cells(j + 1, i
+ 1)) Then 'Rework
                                    If   CDbl(Sheet10.Cells(i   +   1,
ReworkJob(i) + 1)) > MaxRework Then
                                        ReworkJob(j) = i
                                        MaxRework                    =
CDbl(Sheet10.Cells(i + 1, ReworkJob(i) + 1))
```

```
                        End If
                    End If
                End If
            Next
        End If
        'Record the rework adjustment for next time period

    Else
        'Task is finished
        If JobLeft(i) <= 0 Then
            JobLeft(i) = 0
        End If
    End If
Next i 'next task

'Check if all jobs are done
AllJobDone = True
For i = 1 To 20
    If JobLeft(i) > 0.001 Or ReworkJob(i) > 0 Then
        AllJobDone = False
        Exit For
    End If
Next i
Loop

For i = 1 To 20
    Sheet8.Cells(q + 1, i + 1) = JobLeft(i)
Next i
'record detailed single step task performing in sheet8

SimRun = q

Exit Function

errorhandle:
    'Debug use only
    Dim ErrMsg As String
    ErrMsg = "Error Number: " & Err.Number & Chr(13) & Chr(10)
    ErrMsg = ErrMsg & "Error Source: " & Err.Source & Chr(13) & Chr(10)
    ErrMsg = ErrMsg & "Error Description: " & Err.Description & Chr(13)
& Chr(10)
    MsgBox ErrMsg
```

```
End Function


Sub LCImprove(ByVal memberid As Integer, ByVal JobID As Integer, ByVal
TimeDivider As Double, ByVal Eij As Double)
    'Need TimeDivider to finish the task, so the Lij/TimeDivider is what
he learn
    Dim ThetaJ As Double
    ThetaJ = CDbl(Sheet5.Cells(JobID + 1, 7))
    'Eij = 0.02
    If ThetaJ > Sheet13.Cells(memberid + 1, JobID + 1) Then
        Lij = (ThetaJ - Sheet13.Cells(memberid + 1, JobID + 1)) *
Exp((Sheet13.Cells(memberid + 1, JobID + 1) - ThetaJ) * Eij)
        Sheet13.Cells(memberid + 1, JobID + 1) = Sheet13.Cells(memberid
+ 1, JobID + 1) + Lij / TimeDivider
        If Sheet13.Cells(memberid + 1, JobID + 1) > ThetaJ Then
Sheet13.Cells(memberid + 1, JobID + 1) = ThetaJ
        Sheet4.Cells(memberid + 1, Sheet5.Cells(JobID + 1, 3) + 1) =
CDbl(Sheet4.Cells(memberid + 1, Sheet5.Cells(JobID + 1, 3) + 1)) + Lij
* 0.788 / TimeDivider
        If Sheet4.Cells(memberid + 1, Sheet5.Cells(JobID + 1, 3) + 1) >
ThetaJ Then Sheet4.Cells(memberid + 1, Sheet5.Cells(JobID + 1, 3) + 1)
= ThetaJ
    End If
    Exit Sub

End Sub


Function MultiTask(ByVal JobID As Integer) As Boolean 'Return if a task
is a single task or a task in multiple task group
        MultiTask = False
        Block_Num = CInt(Sheet6.Cells(1, 2))
        For i = 1 To Block_Num
            Do While (VarType(Sheet6.Cells(i + 1, j + 1)) > 1)
                j = j + 1
                If JobID = CInt(Sheet6.Cells(i + 1, j + 1)) And j > 2 Then
                    MultiTask = True
                End If
            Loop
        Next i
End Function
```

```
Sub AUTO_ANOVA()
    'Automatically run the simulation for different treatment levels
    Dim FName(5, 3) As String
    reps = 50 'run 50 replication for each treatment
    counter = 1
    FName(1, 1) = "Low"
    FName(1, 2) = "Medium"
    FName(1, 3) = "High"
    FName(2, 1) = "Low"
    FName(2, 2) = "Medium"
    FName(2, 3) = "High"
    FName(3, 1) = "Medium"
    FName(3, 2) = "High"
    FName(4, 1) = "Fast"
    FName(4, 2) = "Slow"
    For f3 = 1 To 2
        Offset0 = 32 * (f3 - 1)
        For i3 = 2 To 31
            For j3 = 2 To 301
                Sheet11.Cells(j3, i3) = Sheet15.Cells(44 + j3, Offset0
+ i3 - 1)
            Next
        Next
        For f2 = 1 To 3
            For f1 = 1 To 3
                For f4 = 1 To 2
                    If f4 = 1 Then
                        Eij = 0.02
                    Else
                        Eij = 0.1
                    End If
                        Sheet14.Cells(counter + 2, 1) = FName(1, f1)
                        Sheet14.Cells(counter + 2, 2) = FName(2, f2)
                        Sheet14.Cells(counter + 2, 3) = FName(3, f3)
                        Sheet14.Cells(counter + 2, 4) = FName(4, f4)
                        For r = 1 To reps
                            Offset0 = 21 * (f2 - 1)
                            For i2 = 2 To 21
                                For j2 = 2 To 21
                                    Sheet10.Cells(i2,      j2)      =
Sheet15.Cells(22 + i2, Offset0 + j2 - 1)
                                Next
```

```
                            Next
                            Offset0 = 21 * (f1 - 1)
                            For i1 = 2 To 21
                                For j1 = 2 To 21
                                    Sheet9.Cells(i1,       j1)       =
Sheet15.Cells(i1, Offset0 + j1 - 1)
                                Next
                            Next
                            Sheet14.Cells(counter  +  2,  5  +  r)  =
SimRun(Eij)
                        Next
                        counter = counter + 1
                Next
            Next
        Next
    Next
End Sub


Sub RandomReworkProb()
    'Sample sub used to generate rework probability
    Randomize
    For i = 2 To 21
        For j = 2 To 21
            If Sheet9.Cells(i, j) > 0 Then
                If Rnd() < 0.8 Then
                    Sheet10.Cells(i, j) = CInt(Rnd() * 1 + 1) / 10
                Else
                    Sheet10.Cells(i, j) = CInt(Rnd() * 8 + 1) / 10
                End If
            End If
        Next
    Next
End Sub


Sub RandomSchedule()
    'Sample sub used to generate member schedule
    Dim tmpi As Integer
    For i = 1 To 30
        For j = 1 To 300
            tmpi = CInt(Rnd() * 19) + 1
            Select Case tmpi
```

```
            Case 1:
                Sheet11.Cells(j + 1, i + 1) = 0
            Case 2 To 6:
                Sheet11.Cells(j + 1, i + 1) = 0.25
            Case 7 To 10:
                Sheet11.Cells(j + 1, i + 1) = 0.5
            Case 11 To 15:
                Sheet11.Cells(j + 1, i + 1) = 0.75
            Case Else:
                Sheet11.Cells(j + 1, i + 1) = 1
            End Select
        Next
    Next
End Sub


Function CheckAssign(ByVal time_idx As Integer, ByVal task_idx As Integer)
As Integer
    'Return the member to which a task is assigned at a time period
    Dim tmpint As Integer
    tmpstr = Sheet7.Cells(time_idx + 1, task_idx + 1)
    tmplng = InStr(1, tmpstr, "M", vbTextCompare)
    If tmplng > 0 Then
        tmpint = CInt(Right(tmpstr, Len(tmpstr) - tmplng))
    Else
        tmpint = 0
    End If
    CheckAssign = tmpint
End Function
```