Dynamic Quality and Change Management For Large Scale Concurrent Design and Construction Projects

by

Sang Hyun Lee

B.E., Architectural Engineering, Dong-A University, 2000

Submitted to the Department of Civil and Environmental Engineering in
Partial Fulfillment of the Requirements
for the Degree of

Master of Science in Civil and Environmental Engineering

at the

Massachusetts Institute of Technology

June 2003

Signature of Author ....
Department of Civil and Environmental Engineering
May 9, 2003

Certified by ....
Feniosky Peña-Mora
Associate Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by ....
Oral Buyukozturk
Chairman, Departmental Committee on Graduate Studies

Dynamic Quality and Change Management For Large Scale Concurrent Design and Construction Projects

by

Sang Hyun Lee

Submitted to the Department of Civil and Environmental Engineering
on May 9, 2003 in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

ABSTRACT

Though concurrent design and construction has been lauded for reducing the time of the total processes, such an approach may make projects more uncertain and complex than those where a sequential design and construction process is used. The main sources of risk are iterative cycles that result from errors and changes introduced during the execution of concurrent design and construction. Such cycles create subsequent impacts on the project performance. In addition, traditional network scheduling tools are not adequate to deal with problems encountered during concurrent design and construction because they consider the project only as a combination of discrete activities and ignore dynamic interactions between different activities. In this context, Dynamic Planning and control Methodology (DPM), a simulation-based planning and control tool, was developed to help prepare a robust construction plan that would avoid these uncertainties.

However, DPM focuses only on iterative cycles caused by quality problems in construction, and it lacks the capability to deal with change cycles. In order to address this issue, this paper proposes a framework for quality and change management and a new generation of DPM based on this framework. The new DPM incorporates the traditional network tools and buffering strategies into system dynamics simulation models and aims to capture iterative cycles and their impact on design and construction performance in advance. Generated policy guidelines and analysis by the new DPM show comprehensive project profiles and help avoid unnecessary redundant subsequent actions. In addition, uncertainties of complex concurrent projects can be reduced by the buffering strategy of the new DPM. Finally, a case study of a bridge project demonstrates that DPM can help prepare robust and systemic planning for concurrent design and construction in the real world setting, and DPM allows managers or site engineers to analyze the unpredictable incidents that may occur during concurrent design and construction and to anticipate their impacts on project performance.

Thesis Supervisor: Feniosky Peña-Mora
Title: Associate Professor of Civil and Environmental Engineering

ACKNOWLEDGEMENT

First, I would like to thank Professor Peña-Mora for his continual guidance and support for this research and my family. Working with him is the most valuable and exciting experience in my MIT life and I believe our relationship has something beyond just research matters. As he has been my supporter and mentor, I hope that I would be his proud student and great friend throughout our continuous research and life.

Technically, I would like to acknowledge the contribution to this paper by Joe Peck, Corporate Planning and Scheduling Manager, Bill Lemoine, Vice President, and John Foster, Senior Project Manager at Modern Continental Co., Philip Helmes, Vice President, and Margaret Fulenwider, Senior Consultant at InteCap Inc., Dr. Mikio Shoji, Senior Managing Director at Kajima Corporation, and the financial support for this research received from InteCap Inc., Kajima Corporation, the National Science Foundation CAREER Award, and the White House PECASE Award CMS-9875557.

My gratitude goes to my friends in Korea and US. Sharing my small achievement with them is the unforgettable moment to me. I am very happy to have friends like you to share my personal feelings. In particular, I would like to show my thanks to MoonSeo Park, the former graduate student at IESL and professor at NUS, and his family, having the pleasant memory with them in US and Singapore.

I am thankful to my family for their love and understanding. Though we are live separately on the opposite side of the globe, they have been a constant source of encouragement during my studies at MIT. Finally, I would like to share this moment with my lovely wife, JungSook. Without her love and care, this moment would not have been a reality. I remember every moment with her and believe that these are the most valuable property throughout my life.

TABLE of CONTENTS

LIST of FIGURES

LIST of TABLES

CHAPTER 1

INTRODUCTION

Nowadays, concurrent design and construction is gaining popularity in the industry due to the increased demand for faster development time. However, despite the promise of speed, increased uncertainties and complexities in concurrent design and construction can make a project more difficult than ever to handle. Although traditional tools have been the most popular planning mechanisms used in construction projects, they are not sufficient to represent what really happens during concurrent design and construction. They have a discrete view, decomposing the project into activities that can have only individual relationships with adjacent activities. In addition, the impact of iterative cycles caused by errors and changes is not addressed by traditional tools. In particular, when construction is performed concurrently, the effectiveness of the traditional network-based tools needs to be questioned, since there are more possibilities of introducing iterative cycles in the concurrent design and construction process. In this context, a dynamic approach, which can deal with the uncertainty of the concurrent design and construction process, has emerged as an alternative to the traditional network-based tools. In an effort to meet this industry need, the Dynamic Planning and control Methodology (DPM) was developed. DPM aims to help prepare a robust construction plan against uncertainties and provides policy guidelines for unexpected events during actual execution by supplementing the network-based tools with system dynamics and reliability buffering [Park and Peña-Mora, 2002]. However, since DPM in its original version focused only on the quality aspect of construction performance, it lacks the capability to address change iterations caused by other factors, such as external requests and uncontrollable issues. Moreover, changes and consequent conflicts are very common to concurrent design and construction projects and can substantially increase the duration and total cost [Ibbs, 1997]. In addition, in its original version of DPM did not emphasize identifying construction changes in a timely manner to minimize possible conflicts in the project management. These issues heavily support the need for an extension of the capabilities of the original DPM.

As an extension of DPM, this paper presents a framework to explain the way in which quality and change management more adequately address problems in real-world concurrent design and construction projects. The quality and change management framework will be useful in identifying the dynamic behavior of concurrent design and construction processes and in helping to prepare a more effective construction plan by analyzing iterative error and change cycles and their impacts. Based on these identified impacts of iterative cycles, *reliability and stability buffering* is enhanced as a mechanism to reduce sensitivity to their impacts in concurrent design and construction. Finally, a web-based system, which incorporates these components, is developed to assist diverse interdisciplinary parties, in particular, in geographically distributed complex projects.

CHAPTER 2

RESEARCH METHODOLOGY

To provide a systemic approach to managing uncertainties and complexities in concurrent design and construction, system dynamics and the overlapping framework for construction projects are adopted by the quality and change management framework and DPM.

System Dynamics was developed to apply control theory to the analysis of industrial systems in the late 1950's [Richardson, 1985]. In this research, system dynamics is adopted to acquire the realities of dynamic complexities and feedback processes in the actual construction with its powerful analytical capability and simulation ability. System dynamics has been applied to many complex industrial, economic, social, and environmental systems of all kinds [Turek, 1995]. However, having realities based on system dynamics may lack the applicability and flexibility to be applied to diverse projects. On the other hand, the traditional network-based tools have demonstrated their applicability and flexibility, serving various projects without much resistance, though they have difficulty handling dynamic complexities in a real project. In that sense, DPM incorporates them into the system dynamics models to enhance its applicability and flexibility for ease of use.

In addition, the overlapping framework for construction [Peña-Mora and Li, 2001] is used and extended to analyze the activity characteristics to provide insight into the actual processes in concurrent design and construction. Their overlapping framework focuses on the transfer of physical production units between overlapped activities by applying concurrent engineering to the construction field. Basic idea of the overlapping framework is that overlapping practices should vary depending on the characteristics of construction activities, which includes activity production rate, production reliability in the predecessor activity, and sensitivity in the successor activity as key characteristics [Peña-Mora and Li, 2001]. Based on this framework, our research explores the way to represent iterative cycles and their impact on the performance during construction.

CHAPTER 3

FUNDAMENTALS OF DPM

The previous DPM had been developed as integrated methodology to handle dynamic behavior of complex construction projects. Construction projects are inherently dynamic and complex involving multiple feedback processes and non-linear relationships [Sterman, 1992]. These features make construction projects uncertain and consequently, they can generate schedule delay and cost overrun. To effectively handle these challenging issues, several concepts and logic of DPM have been derived from closer observations of construction processes in order to provide an integrated methodology [Park and Peña-Mora, 2002]. In this section, fundamentals of DPM are briefly introduced.

3.1. User-defined Modeling Approach

Although simulation-based planning tools provide effective handling for dynamic complexities, they have been limited to a specific set of activities on a project. To extend the applicability of simulation-based tool, DPM aims to provide a general pre-structured model and parameters that are common to all construction projects. This way, users can set it to their specific projects adjusting provided parameters.

3.2. Consideration of Feedbacks

DPM focuses on capturing the feedback processes in construction projects that make projects dynamic and uncertain and cannot be captured in the traditional network-based planning tools. For example, when a control action is taken to reduce variations from a planned performance, the action can fix the problems but at the same time its side effects can deteriorate the project performance. DPM aims to capture these kinds of feedback that are caused by human response to control actions.

3.3. Capturing Construction Dynamics

DPM considers change as an iteration trigger not just a result, so already made changes can be the source of subsequent changes in either concurrent, succeeding or preceding activities. For example, changes in design work that have been made by mistake can cause subsequent changes in construction if the error is not identified before construction starts. In this case, the design changes are a result to the designer, while they can be a need for changes to the construction crew. On the other hand, as main constraints to the construction progress, dependencies involved in processes are identified and modeled in detail. External dependency captures the dependency between activities in a way similar to the precedence relationship in network tools [Ford and Sterman, 1997]. However, external dependency deals with the relationship during the whole activity duration, while the precedence relationship only refers to the start and finish of activity. For example, if the successor activity is scheduled to start at 50% completion of the predecessor activity and the successor activity can proceed in proportion to the progress of the predecessor activity, external dependency can be represented as the left-side of Figure 1.

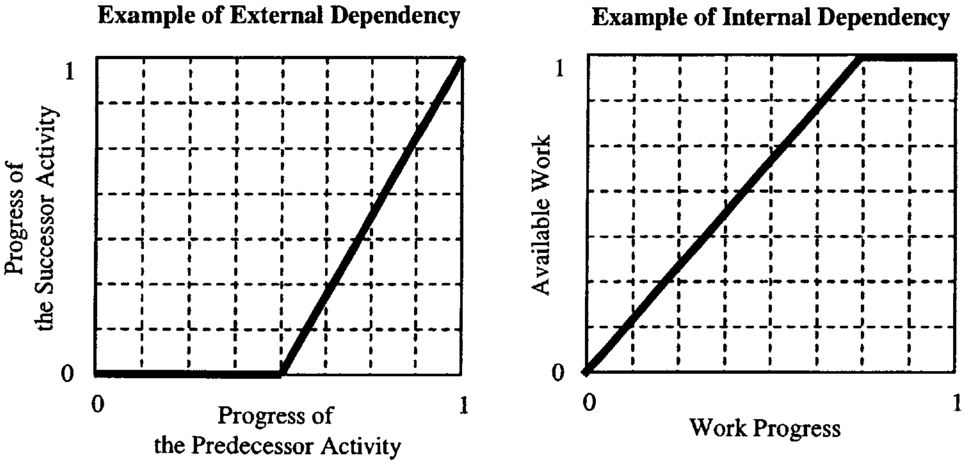


Figure 1. Example of Dependencies [Ford & Sterman, 1997]

On the other hand, internal dependency aims to capture procedural or physical constraints within an activity, which is not addressed by traditional network tools. For example, in the right-side of Figure 1, available work within activity can proceed in proportion to work progress due to certain constraints till 75% completion and, after then, it can proceed freely without the restriction from work progress. Those two dependencies are adopted in order to represent inter and intra relationships of activities.

3.4. Reducing Sensitivity to Changes

To decrease non-value adding iterations through reducing sensitivity of the design and construction process to unintended changes, DPM provides an appropriately pooled, located, characterized and sized reliability buffer. Usually, the traditional contingency buffer in construction planning is used to guarantee the completion time either of an activity or a project. In usual practice, it is positioned uniformly at the end of the activity duration having some more time. However, it tends to be used as part of an activity without clear distinction from the original duration. As a result, time added to the original duration may not effectively protect the planned schedule because when people realize that they have more time to complete a task than the time known, their work productivity usually goes down [Sterman, 2000]. It can also be explained by the Parkinson's law [Parkinson, 1957] that the work expands to fill the time available for its completion.

In contrast to the traditional contingency buffer, Park and Peña-Mora [2001]'s research argues that reliability buffer can systemically protect the whole project schedule from being disrupted by failures in individual activities. Reliability buffering first attempts to take off contingency buffers from individual activities when there exists contingency buffer, and makes each activity benefit from appropriate schedule pressure. Excessive schedule pressure may deteriorate workers' productivity. However, appropriate and well managed schedule pressure can increase their productivity [Sterman, 2000]. This fact is derived from not only the physiological and psychological effects but also logistical considerations. Suppose that Activity A has 10 days to finish it. But, due to the urgent request from the owner, the duration of Activity A is reduced as 8 days. The project manager explained this situation and announced it as a 'crisis' for the project. In that case, construction crews would not resist this shortened duration and try to keep the reduced schedule, even though overtime is applied. They would put off their personal activities for that period with tolerance, though they may be tired. In that case, construction crews' productivity is increased comparing to the planned productivity, when we measures it as work accomplished per hour of effort. However, if Activity A has a longer duration like 50 days and required to finish within 40 days, crews would start to resist this request after some time because of their fatigue from declining health, lack of a social life, and even family problem. Studies of the construction industry and other manual labor contexts indicate that long work hours begin to reduce productivity after a week or two, with the full effect requiring somewhat longer [Oliva, 1996].

After taking off contingency buffers from individual activities, the reliability buffer is positioned at the beginning of activities, rather than the end of activity in the traditional contingency buffer. These different logistics can handle ill-defined tasks by introducing a pre-checking process having time to capture and correct predecessors' hidden errors before being performed in unknown state. Meanwhile, the size of buffer needs to be varied depending on construction characteristics. The degree of overlapping between activities should be decided in a way that enough time to discover and fix problems made in the predecessor activity can be secured before the successor activity starts [Peña-Mora and Li, 2001]. Following this argument, the size of buffer is determined by appropriate simulation using activity characteristics and project control policies. In addition, the size and the location of the buffer are continuously updated in order to handle changing construction performance, which results from the updated information on construction system characteristics.

CHAPTER 4

FEEDBACK PROCESSES BY ERRORS AND CHANGES

Construction projects are inherently complex and dynamic, involving multiple feedback processes [Sterman, 1992]. The uncertainty and complexity of projects results mainly from these feedback processes in design and construction. Feedback processes can be represented as the simultaneity of the positive and negative effects of decisions. In other words, the decision to handle a problem that arises can have a positive effect, fixing the problem itself but, at the same time, it can have another side effect, one that may generate other unintended problems.

One of the main sources for these feedback processes is the gap between the planned work scope and the actual work scope, due to errors and changes. In other words, the work scope is usually increased as a result of the discovery of errors and the request for changes throughout the actual execution. To deal with this increase in the work scope as well as to keep the schedule as planned, the manager may take appropriate control actions such as adding more resources (ex. material, equipment, workforce, and etc.), or adopting overtime. Many other control actions can be taken in order to reduce this gap of work scope. These solutions for the confronted problem, however, may generate unintended negative side effects on the project performance, such as the decrease of productivity and quality. For example, although the adoption of overtime may handle the increased amount of the work being performed, the extended work hours may increase the workforce's fatigue. Ultimately, accumulated fatigue could worsen productivity and quality of work and would slow the project progress down as more errors and changes are introduced. This explains why errors and changes are iterative and make the project uncertain and complex by producing side effects from control actions that have been taken, as seen in Figure 2.

Negative side effects caused by errors and changes become more hazardous when fast-tracking and concurrent engineering techniques are applied. Projects applying both techniques could be more uncertain and complex due to the lack of finalized information about predecessor activities and the complex inter-relationships of activities than traditional sequential development [Lee et al, 2003].

Therefore, the understanding of error and change and its iterative impact is a key to reduce multiple feedback processes, particularly, in concurrent design and construction. On the following pages, as a foundation for the proposed framework, we discuss the ways in which error and change are generated and are represented in the actual construction processes.

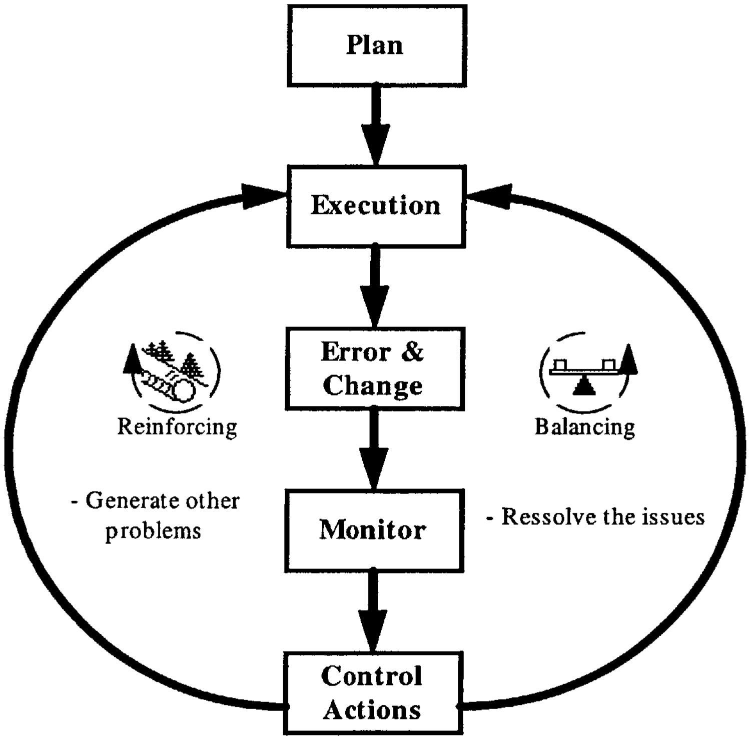


Figure 2. Feedback Processes throughout the Actual Execution [Modified from MoonSeo Park, 2000]

4.1. Definition of Error and Change

Most iterative cycles in design and construction are triggered by errors and changes. Error is defined as defective work of poor quality, such as the placement of piling in the wrong location or poor concrete performance; change is defined as any work required of the contractor or subcontractor that was not specified in the original contract document [Trauner, 1992]. An example of this would be the owner's request to change the building's purpose as from a usual office space to a library facility.

4.2. Causes for Error and Change

In the actual execution, errors and changes are not generated in a constant or predictable way. Their generation depends on non-linear relationships with several variables which are time-dependent and highly interactive. One such non-linear relationship is with the workers' learning curve of the applied technology or method. In the learning process, it usually takes a considerable amount of time to be proficient during the initial stages and there would be no notable improvement during the later stages. In other words, the proficiency caused by the learning process usually follows 'S' shape curve. Therefore, it can be said that errors have more possibility of being generated in these initial stages than the later stages in terms of workers' learning curve.

Table 1. Key Variables which Affect Errors and Changes

	Key Variables	Controllability
Error	Policy, Environmental Reliability	Uncontrollable
	Worker's Experience, Schedule Pressure, Learning Curve	Controllable
Change	Political, Social, Environmental Stability, External Change Request	Uncontrollable
	Information & Resource Availabillity, Different Conditions, Lack of Action	Controllable

Among many potential variables, Table 1 summarizes key variables which might affect errors and changes and their characteristics in terms of controllability during the actual execution. In the case of errors, most variables can be managed and controlled. For example, hiring experts or holding workshops of the adopted construction technology could increase worker's skillfulness and boost learning curve in its initial stages. At the same time, however, there are some uncontrollable variables which are related to policy and environmental reliability. For example of the policy reliability, the adoption of unproven process, technology, or material by the decisions of high-level management team, could include the inherent problems which would generate errors during the actual execution. In addition, the bad weather, as an example of the environmental reliability, could affect the concrete curing performance. In these examples, errors may be hard to be managed and controlled by the project level decision.

On the other hand, in the case of changes, political, social, and environmental stability which affect the change generated are not controllable and they are beyond the project level. External change requests by the owner and high-level management team are also uncontrollable. However, other change requests caused by information and resource unavailability, different conditions, and lack of action can be controlled by the appropriate processes on the project level. For example, close investigation of the site throughout precise methods could improve the accuracy of the information about site conditions and consequently reduce the potential changes. Meanwhile, there may be certain changes that don't require direct cost and consequent tasks, and this phenomenon is denoted as lack of action. One such example is late inspection by a third party who has a direct contract with owner. If he/she performs inspection two days later than stated date in the specification, it generates two day schedule delay. Though it does not drive up any cost or require unplanned additional tasks to catch up with usual change cases, it can be categorized as a change based on the above definition [Trauner, 1992]. This kind of change is common in design and construction projects due to a significant amount of interactions among related parties.

4.3. Error and Change Impact Patterns

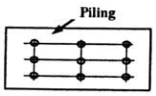
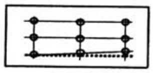
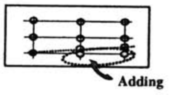
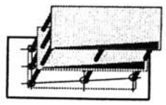
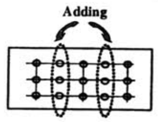
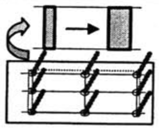
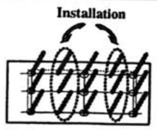
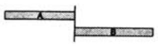
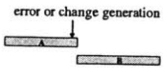
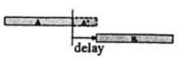
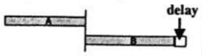
Among many possible ways to classify error and change impact patterns on construction, this paper focuses on their impacts on the work scope in order to quantify how much they generate additional unplanned works. As mentioned earlier, the increased work scope mainly caused by errors and changes is the source of feedback processes. Therefore, identifying how errors and change influences on the work scope can help to capture and reduce feedback processes.

Error and change impacts can have three different patterns, *rework, scope substitution,* and *extra work*, during construction processes as illustrated in Table 2. The first pattern is the *rework* of an activity required for it to achieve the planned goal, and its target is the problematic task or the issued task. In other words, it is to work again on the problematic or the issued task. Therefore, the work amount of rework is the same as the original work. For example, if the location of piles is different from the original plan, one possible reaction is to add piles on the planned location, keeping previous piles. Or piles can be added after removing previous piles. In the case of change, as illustrated in Table 2, assuming that change is requested to increase the structure capacity of piles to bear more load than planned, adding more piles can be a way to accommodate the change request.

The second one is the *scope substitution* on the project. It does not require any unplanned additional work, but substitutes the kind of work to be performed. Therefore, it does not generate any additional work amount. In the case of above wrong location of piles, manager can decide to change the subsequent works in order to keep the previous work. Due to the change of pile location, subsequent column and floor installation can be different from the original plan, but there is no need for additional works in succeeding activities.

Finally, the third on is the *extra work*. It is generated to supplement error or change and requires performing unplanned additional work in the other tasks or activities. Extra work is usually preferred to rework in the construction because construction rework is normally accompanied with the demolition of what has already been built [Park and Peña-Mora, 2003] and direct additional work amount is usually less than rework. As a result, construction managers tend to avoid rework on problematic tasks by adopting extra work modifying their design and specification. Therefore, the work amount can be represented as less than the possible rework and more than scope substitution.

Table 2. Example of Error and Change Impact Patterns in Construction
[Augmented from MoonSeo Park, 2000]

	Planned Performance	Error or Change Generation	Error or Change Impact Patterns in Construction		
			Rework	Scope Substitution	Extra Work
Error Case Example		 Error Generation: Piling is not located as planned	 Two piles are added to achieve planned goal.	 Column position is changed.	 To keep work so far, unplanned cantilevers are installed.
Change Case Example		Change Request & Approval: "Increase the structure capacity of piles to bear more load, since heavier equipment is going to be housed on the structure"	 Additional piles are added.	 The size of columns is changed.	 To keep work so far, unplanned columns are installed.
Gantt Chart Denotation	 A: Piling B: Structure		Object: Activity A 	Object: Activity B 	Object: Activity B 

However, in general, error and change impact pattern is not issued individually. In the above piling example, if the scope substitution is adopted for a wrong pile location, columns could not handle the planned area as expected. Therefore, additional cantilever work could be added to the successor activity to structurally cover the planned area as illustrated in Table 2, and it is extra work. This phenomenon can be interpreted as iterative cycle because the later extra work is apparently caused by the former scope substitution based on the relationships with each other, rather than being occurred individually. This is also applied to the above change case. Additional columns can be installed as the *extra work* in order to incorporate the change of the piling activity, and even the size of column can be enlarged to accommodate the increased bearing capacity as the *scope substitution.*

Therefore, in order to capture total impact of error and change, each direct error and change impact pattern is identified as seen earlier, for example, A causes B as rework and B causes C as extra work. And then the corresponding quantification is applied to get the quantity of each sequence and finally, total impacts of error and change are acquired by the sum of total sequence.

4.4. Derivative Activity

Error and change impacts on the construction performance vary depending on several factors such as an activity own characteristics and its relationships with other activities [Park and Peña-Mora, 2001]. However, the serious impact of error and change usually occurs when they are discovered at the later stages. For example, if error and change in the predecessor activity are found and immediately adopted in the predecessor activity, the possible reaction is executing that task again or executing additional work to accommodate the generated error or change. Therefore, their impact on the successor activity may not be very significant because the successor activity can have time to adopt them.

However, if error and change in the predecessor activity are discovered at the successor activity, its impact on performance requires careful attention due to the subsequent impacts of already performed work. Errors and changes that are not immediately discovered and approved after their generation are main sources that make the project uncertain and complex because they usually cause sudden work overflow at late stages of the project, which is called 'the last minute syndrome' [Lee et al, 2003].

In this case, two cases are possible in concurrent engineering as illustrated in Figure 3. One of them is that error or change in the predecessor activity is found before the predecessor activity finishes. In this case, the successor activity can request correction or adoption of the error or change during the execution of the predecessor activity. Due to the work re-executed on the predecessor activity, the successor activity can be delayed before the work in the predecessor activity is completed again. In addition, the predecessor and the successor activity may do more work to correct already completed work because of their inter-relationships such as physical and procedural constraints. The other case is after the predecessor activity finishes. If the only way to deal with error or change in the predecessor activity is to work again throughout the predecessor activity, its impact is more significant than the previous case, because workers and equipments in the predecessor activity were already withdrawn. In addition, if the predecessor and the successor activity are performed by different subcontractors, this situation can require new or additional contract to accommodate the discovered error or change, and it may generate a contractual dispute in order to avoid the responsibility of this error or change. We denote this situation as *'derivative activity'* and whether error or change is detected early and whether they are throughout an adequate error and change management process becomes a significant issue in the effective management of the scope and the corresponding performance.

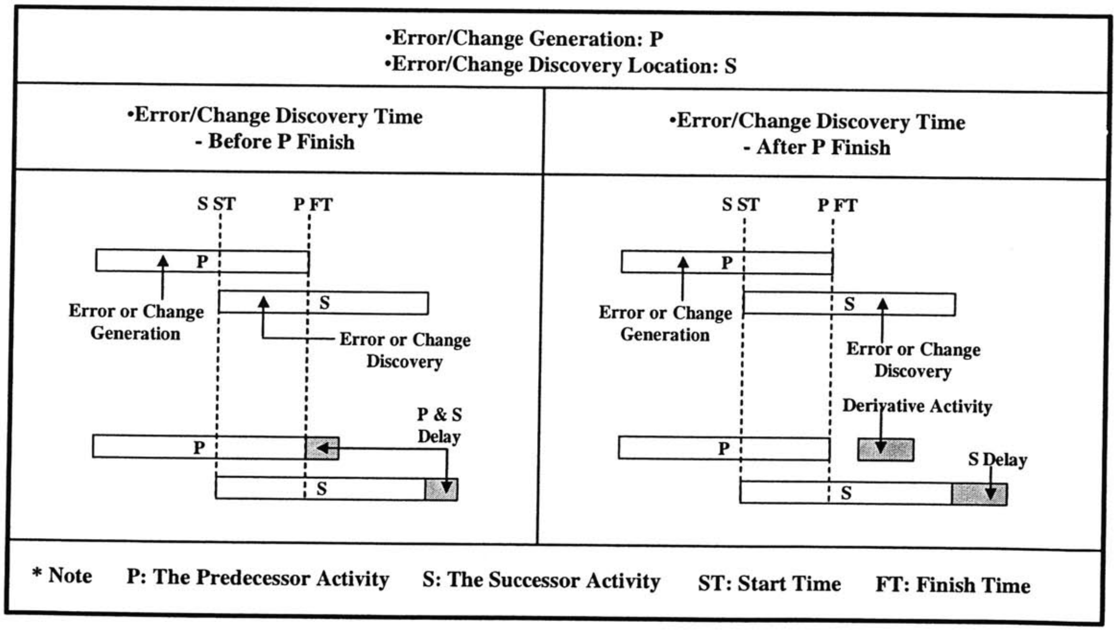


Figure 3. Impact of Late Discovery

CHAPTER 5

QUALITY AND CHANGE MANAGEMENT MECHANISM

Discussions on different impact patterns of error and change show that they cannot be treated as discrete events. Therefore, identifying how error and change behave and are managed during the actual execution is a critical point to effectively deal with them. In this sense, the framework for quality and change management is proposed with dynamic relationships among design and construction activities. For this, a holistic view is adopted, rather than the discrete view of traditional network planning tools.

5.1. Internal Quality and Change Management Mechanism

Before exploring the whole project network, we discuss how error and change behave internally within the activity. As summarized in Figure 4, first, work is performed based on the given work scope. However, all the work being performed does not guarantee that this work is done correctly and can have errors. To represent this situation, *reliability* is here defined as the degree of how much performed tasks has been done correctly during the actual execution. High reliability means that a small number of errors are introduced, while low reliability represents the possibility of a great number of errors during design and construction. For example, if Activity A has 90% reliability, error in the Activity A is expected to be 10% of the total work scope during the actual execution. In addition, this reliability can be varied depending on diverse variables over time. For instance, if the schedule pressure becomes high as the project progresses, the worker's fatigue would be increased and consequently, more errors would be introduced, which means reliability becomes low.

However, some portions of errors may not be uncovered and become hidden during the quality management process. In order to explain this situation, we use the dynamic variable, the *quality management thoroughness*, which is defined as the degree of how many of the existing quality problems are identified during a quality management process. For example, if Activity A has 90% of the quality management thoroughness, 90% of total errors are discovered and remaining 10% would be hidden throughout the quality management process.

This concept is based on the fact that applied quality management techniques are not perfect and constant during the execution. In reality, diverse techniques for quality management can be applied and they have a different effect on error discovery depending on several conditions such as their effectiveness on a particular kind of construction project and users' familiarity of applied techniques. Suppose welding is performed in the pipe installation activity in the building construction. Reinforcement welding is often used as a quality management technique if welding volume is not sufficient. The inspector found that welding volume of pipes in the second floor is not sufficient, and that means error is uncovered during the quality management process by the inspector. He asked the worker to do reinforcement welding on that already performed pipe and left for the third floor. However, the worker assigned for reinforcement has only experience on basic welding tasks not reinforcement welding. Reinforcement needs to be performed carefully because excessive reinforcement can affects the fatigue-strength of a welding part. Due to his/her no knowledge on this fact, excessive reinforcement may occur since the worker may think that more is better. In that case, welding parts with excessive welding will become hidden errors. Therefore, quality management thoroughness represents the actual effectiveness of applied techniques on the particular work as seen in the above example.

Last component for the internal quality management process is the *elasticity for error*. The elasticity for error represents how much the schedule pressure affects the personnel who perform the quality management process. In other words, if inspectors have more works to be inspected than usual and feel the pressure to meet the schedule, he/she would force them to be progressed aggressively. Therefore, there are more chances that errors become hidden due to his/her hurry. In this sense, having enough quality management staff would be one of the solutions to manage unexpected additional work, and this phenomenon has not been paid much attention to by traditional project management practice. In summary, the elasticity for error is adopted to represent the effect of the schedule pressure on the quality management thoroughness, and their relationship, together with the quantification, is derived from the interview and survey with the quality management staff.

Having these components, each uncovered and hidden error generation ratio can be represented as the form in Eq. (1) and (2), respectively:

$$E_u = (1 - R(t)) \times \prod_{i=1}^{n} (M_{qi}{}^{\alpha i}) \tag{1}$$

$$E_h = (1 - R(t)) \times \prod_{i=1}^{n} (1 - M_{qi}{}^{\alpha i}) \tag{2}$$

where, E_u is uncovered error generation ratio, E_h is hidden error generation ratio, $R(t)$ is reliability in a given activity, M_{qi} is the quality management thoroughness of the overall effectiveness of several applied techniques, αi is the elasticity for each error generation ratio, i represents each applied technique, and n is total number of techniques in the quality management process. The left-side of equations, $(1-R(t))$, are the projected total error generation ratio and the right-side product explains the dynamic effect of the applied techniques. In this case, the product is used in order to combine impacts of different techniques because the failure of one of the applied techniques means the system failure.

Meanwhile, change order process has a slightly complex management mechanism. It has two major elements. One is *the scope management process*, and the other is the claim and change group. First, change is triggered based on *stability* having a given initial scope. Stability is defined as the degree of how many tasks will be done without change request and is insusceptible to scope management issues. High stability means that small number of changes would be issued to a particular activity, while low stability represents the possibility of significant number of change requests during the execution of a particular activity. For instance, if activity A has 90% stability, 10% of the work scope would be subject to change. Among potential changes, some changes can or can't be identified through scope management thoroughness similar to the quality management process. Change that is not identified during scope management process is represented as *latent change*. Latent changes are changes that have not been identified and have the potential to be reactivated as *identified change* at a later stage. Also, change generation process has *the elasticity for change* with respect to scope management thoroughness. Based on these factors, *identified and latent change generation ratio* is formulated in the form of Eq. (3) and Eq. (4):

$$C_i = (1 - S(t)) \times \prod_{i=1}^{n} (M_{si}^{\beta_i}) \tag{3}$$

$$C_l = (1 - S(t)) \times \prod_{i=1}^{n} (1 - M_{si}^{\beta_i}) \tag{4}$$

where, C_i is identified change generation ratio, C_l is latent change generation ratio, $S(t)$ is stability in a given activity, M_{si} is the scope management thoroughness of the overall effectiveness of several applied techniques, β_i is the elasticity for each change generation ratio, i represents each applied technique, n is total number of techniques in the scope management process. The left-side of equations, $(1\text{-}S(t))$, are the total error generation ratio and the right-side product explains the dynamic effect of the applied techniques.

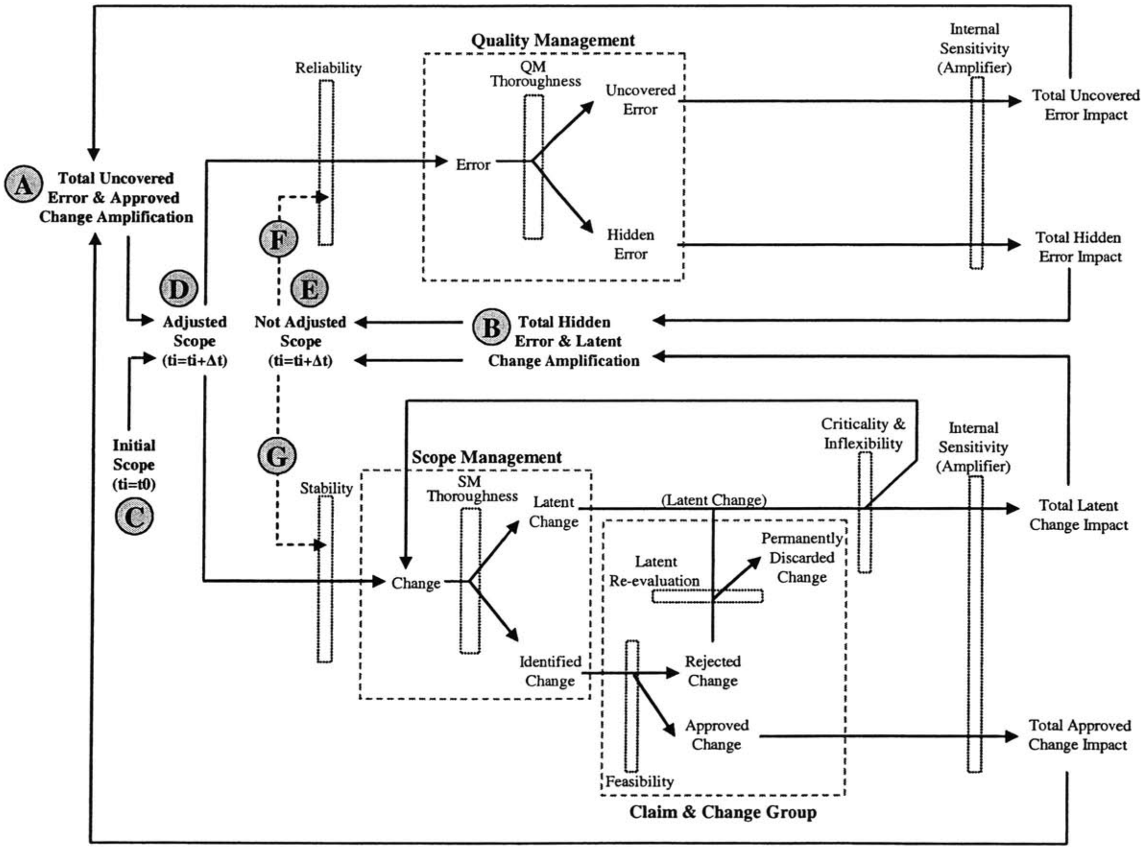


Figure 4. Internal Quality and Change Management Process

However, another process is involved in changes after the above categorization. That is the role of the claim and change groups in a project, which is initiated in order to deal with the claim and change order process. Identified change isn't always approved by the claim and change group because of its *feasibility*. For example, the explosion method was planned for excavation work for Project A. But, the owner failed to get permission to use that method from the local authorities due to community's objection to noise and dust. In that case, the scrape and excavate method can be an alternative to the explosion method in Project A. However, if equipment and subcontractors for the explosion method were already contracted and another equipment and subcontractors for the scrape and excavate method are contracted additionally, it would create significant cost increase. Therefore, the claim and change group on the project can object to the scrape and excavate method and suggest that the owner persuade the community aggressively to prevent paying double cost. In other words, whether identified change becomes *approved change* or *rejected change* depends on the feasibility of the requested change.

In addition, rejected changes become latent change due to the potential for reactivation, particularly, if certain events work against particular action been undertaken. For example, even though the scrape and excavate method was rejected by the claim and change group, it may be brought back to the table if the owner fails to get permission from the community. Rejected change can also be *permanently discarded change*, if there is no need for further consideration. In this change example, though the scrape and excavate method is rejected due to the budget and contractual problem, but it still remains as latent change in order to prepare the worst case, like the community's final objection. If the explosion method is performed without any further problems or is too infeasible to adopt in this stage due to cost or contractual problems, the scrape and excavate method will be permanently discarded. The process for re-consideration for rejected changes is denoted as *latent re-evaluation*. Latent re-evaluation determines if rejected changes are adopted as latent changes or permanently discarded changes. In addition, latent changes also have a possibility to be reactivated due to certain factors related to *criticality* and *inflexibility* of certain activities. Continuing with the above example, while the owner tries to persuade the community to get permission, the scrape and excavate method, latent change, may be back to the table of the claim and change group due to the schedule delay. Though permission from the community is still progressing and the owner reported their positive reaction to the claim and change group, the schedule delay brought by this process becomes intolerable to wait any more, because the excavation work is a very critical and inflexible activity as a starting point of a chain of subsequent structural activities such as the foundation and the superstructure work. In other words, the other subsequent activities can not progress without the completion of the excavation work.

Though change was not introduced as approved change, this latent change may be issued again because it is very critical to the whole network and the activity is very inflexible, which means there are very few or uneconomical choices as the alternatives to this latent change.

5.2. Work Multipliers

So far, it could be observed how error and change can be generated and differentiated based on their behaviors and state in the internal construction process. However, without capturing the dynamic relationships among activities, they may not reflect the actual conditions and effects on construction processes. For instance, if identified change in the scope management process is approved by the claim and change group, its effect can vary depending on the relationships with internal or external activities. In our previous change example, the scrape and excavate method may be approved as a change replacing the explosion method. In that case, it can generate subsequent changes to the remaining tasks within that activity or successor activities such as their late start, because it is less effective than the explosion method in terms of required duration. Consequently, this late start may generate other contractual problems in the worst case and that means its effects can not be limited to just that issue having different impact. Work multiplier is contrived in order to capture this dynamic relationship and to quantify its effect. For this, sensitivity is adopted here to address these impact relationships within or among activities. The former is denoted as *internal sensitivity* and the latter as *external sensitivity*. In addition, sensitivity is related to not only change but also error impact. For example, if piling is wrongly located as seen before, subsequent column and floor work may be affected by different impact strength based on their relationship with piling work.

The one point to note is that sensitivity also governs hidden errors and latent changes. Though they are not uncovered or approved, they can impact successor activities such as generating quality problems. In the piling case, if manager didn't catch the error in piling location, this hidden error may also generate subsequent errors on successor activities like continuously wrongly located columns.

Therefore, *total hidden error and latent change impact* as well as *total uncovered error and approved change impact*, which are denoted as A and B in Figure 4, are finally determined by sensitivity. It implies that the reduction of hidden errors and latent changes is necessary to avoid late and sudden overflow of work to meet the scheduled date, usually known as 'the last minute syndrome'.

5.3. Impact of Adjusted Scope and Not Adjusted Scope

The initial scope, C in Figure 4, is given before the activity starts. However, as the activity progresses, newly introduced work scope may be added to the initial scope due to uncovered errors and approved changes. Of course, these errors and changes include their impacts within the activity through internal sensitivity to account for its impact on the activity under study. Therefore, *adjusted scope*, D in Figure 4, is denoted as the newly added scope with the initial scope and is updated as the activity goes on. In other words, it is apparently perceived as the work to do by managers. Meanwhile, hidden errors and latent changes are also errors and changes though they are not uncovered or approved as errors and changes. These errors and changes, with their impact on that activity through internal sensitivity, are denoted as *not adjusted scope*, E in Figure 4. Not adjusted scope represents a crucial meaning to the project performance, because it is not perceived by managers even though errors may be corrected or considered inevitable at a later time as seen in F and G in Figure 4.

Detailed procedure can be explained using the feedback structure illustrated on Figure 5. First, Loop A and B show change management mechanism and these loops explain the impact of latent change on the stability. If latent changes are accumulated in the process, they may make the stability of the system deteriorate and consequently, more change requests are addressed. In the previous excavation work example, assume that the scrape and excavate method is rejected and becomes latent change because the claim and change group decided to suggest the owner persuade the community with additional research on dust and noise issues by commissioning a third authority. However, in order to keep the original plan, they may use significant amount of unplanned budget and time to prepare the meeting with the community and to do additional commissioning. Furthermore, the start of successor activities can be further delayed till they get permission from the community. Though it is less expensive to reject the scrape and excavate method than to approve it, several unplanned effects are derived from this rejected and latent change. Similarly, hidden errors may also deteriorate the reliability of the process as denoted by Loop C and D in Figure 5. In the piling example, unnoticed wrongly located piles may generate other quality problems on subsequent column and floor works such as the failure of welding between column and floor.

The important point is the connection between these loops as denoted by E and F in Figure 5. If hidden errors are accumulated through the process, it may affect the stability, which represents the possibility of change generation. In the previous piling example, let's assume that wrongly located piles were not found in the quality management process and became a hidden error. At that time, the owner requested the change of building purpose from normal office space to another space which can deal with library facilities. Normally, the library requires greater load-bearing capacity than normal office space.

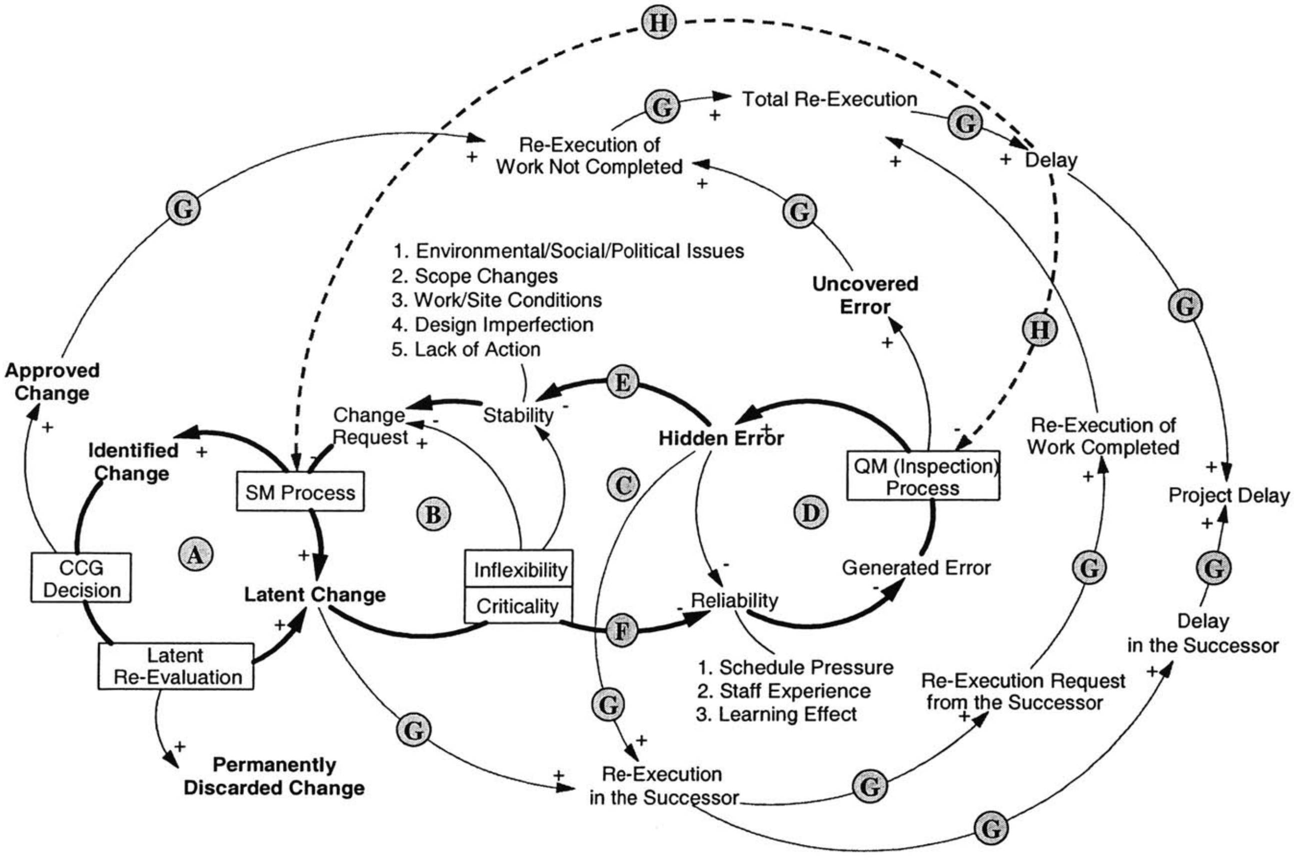


Figure 5. Feedback Processes on Quality and Change Management Process

The claim and change group accepted the owner's request because the original structural plan had redundant dead load capacity. However, it can be hazardous to adopt the requested change, because wrongly located piles may reduce the capability to bear planned dead load. When the manager noticed this hidden error, the decision of the claim and change group had been already made and it might generate the subsequent changes to solve this hidden error problem. Therefore, hidden error as well as latent change may deteriorate stability of the process. This phenomenon is also applied to latent change, which can affect reliability of the process as denoted by F in Figure 5.

In addition, these iteration cycles introduce delay through several steps as denoted by G in Figure 5. Finally, delay can deteriorate scope and quality management processes denoted by H in Figure 5 and, consequently, generate more hidden errors and latent changes. If we assume that Activity A is in the critical path and delayed significantly, the manager may decide to reduce scope and quality management thoroughness on purpose in order to drive up the execution speed. However, that action may not catch errors and changes and it has a potential to be just a temporary expedient. Therefore, it may generate more hidden errors and latent changes and the project can suffer at the later stage due to the last-minute syndrome.

5.4. External Quality and Change Management Mechanism

External sensitivity can make error and change within one activity influence predecessors and successors, which is displayed in Figure 6. For example, *total uncovered error impact* (A) and *total approved change impact* (B) in the activity under study become respective *total uncovered error amplification* (C) and *total approved change amplification* (D) to the predecessor activity through the predecessor activity's external sensitivity (E). And then *total uncovered error amplification* (C) *and total approved change amplification* (D) transfers to *total uncovered error and approved change amplification* in the predecessor activity (F). The reason that uncovered error and approved change are combined is because they are perceived by the manager and the scope will be adjusted to incorporate them. In addition, total uncovered error and approved change amplification of the predecessor activity (F) includes the impact of the activity under study (C, D) and its own total uncovered error and change impact. The same flow is also applied to the successor activity.

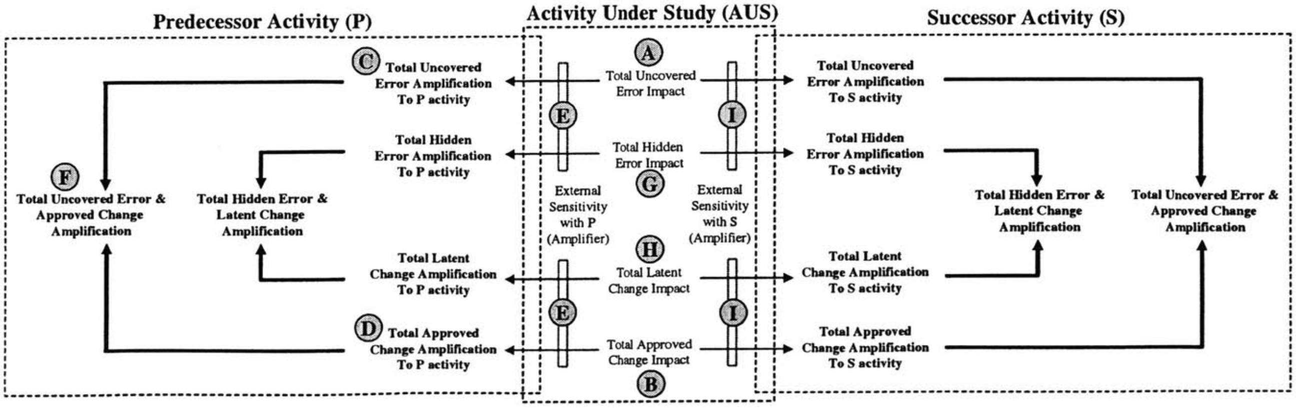


Figure 6. External Quality & Change Management To Predecessor and Successor Activities

Meanwhile, *total hidden error impact* (G) and *total latent change impact* (H) in the activity under study also amplify their effect to the predecessor and the successor activity through corresponding external sensitivity (E, I). After transferred to the predecessor and the successor activity, all amplifications adjust the predecessor and the successor scope. In case of hidden error and latent change, they are accumulated on not adjusted scope of the respective predecessor and successor activity. This process is iterated till the end of the project.

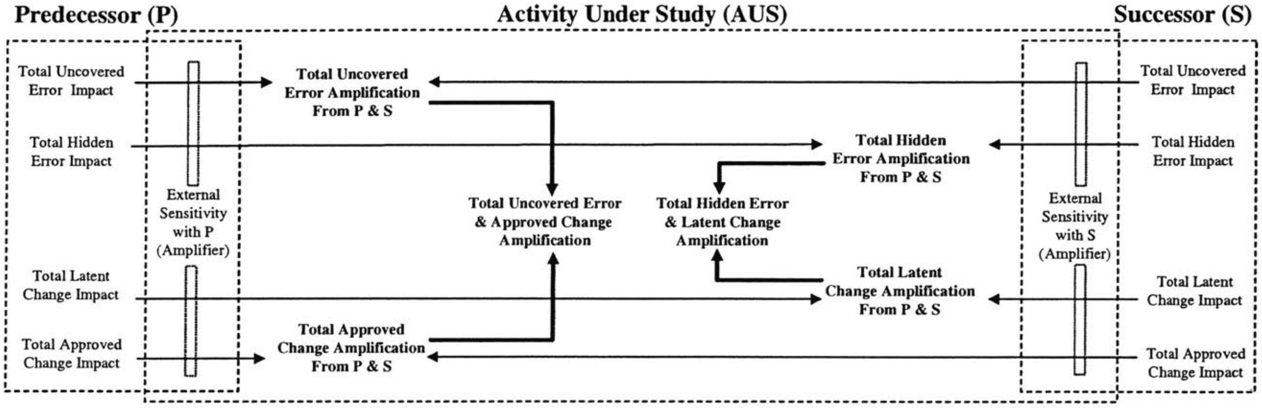


Figure 7. External Quality & Change Management Impact From Predecessor and Successor Activities

Finally, this interaction between activities can be extended to the whole network of the project as seen in Figure 8. If one error or change is generated at the activity, it may affect the activity itself as well as can propagate to the subsequent activities and even the whole network, particularly, in concurrent design and construction.

This propagation of iterative cycles shows the need for the holistic approach to deal with the impact of uncertain events. Traditional static approach may address the propagation impact on the schedule network, however, it lacks the capability to explain what causes what and the quantification of the propagation impact. In addition, the above propagation impact on the project performance can be addressed by the performance differentiation, which will be covered at next section.

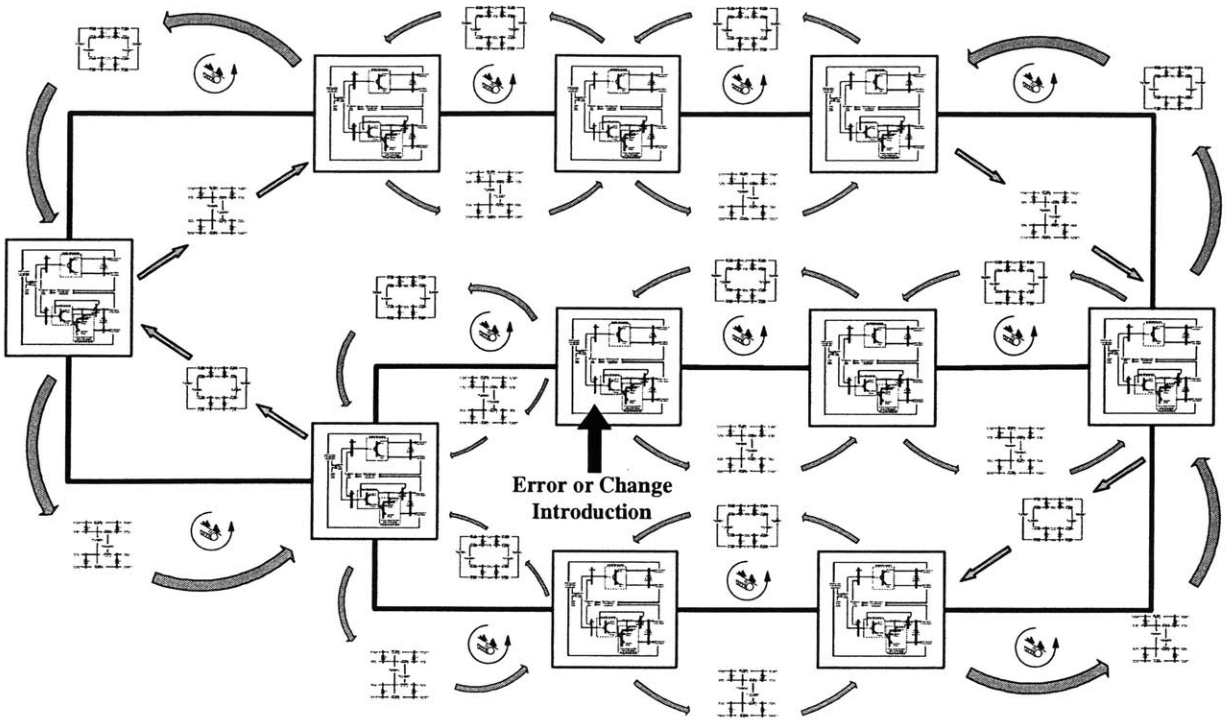


Figure 8. Quality and Change Management Propagation Effect on the Whole Network

CHAPTER 6

PERFORANCE

Having the understanding of the impact of iterative cycles allows us to differentiate the project performance as *planned, perceived, and real performance*. Planned performance is the intended performance determined during the planning stage and it can be expressed as a kind of goal before the actual execution. After a project starts, the manager checks the ongoing performance in several ways, and the monitored performance can be referred to as perceived performance. In other words, the gap between goal and ongoing status can be explained by the effect of uncovered errors and approved changes, as denoted by A in Figure 9.

Usually, the manager changes or corrects the subsequent plan based on this gap. However, this way of comparison may ignore the impact of dynamic relationships on the project. Real performance considers dynamic relationships between the effect of hidden errors and latent changes, and it is what really happens in the project, as denoted by B in Figure 9. The difference between perceived performance and real performance is often disregarded during usual project management. If we can perceive and understand this gap, it would help manage the performance profiles and the reduction of the subsequent ill effects of missing actions, such as sudden work overflow at late stage.

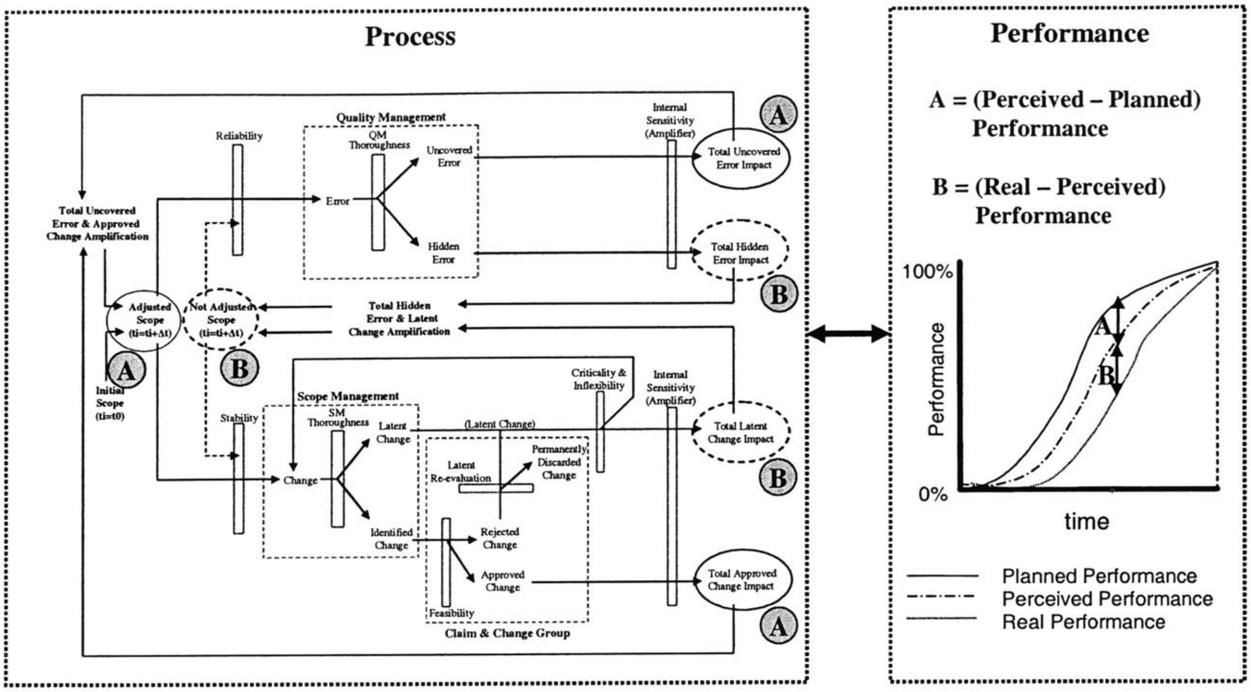


Figure 9. Perceived vs. Real Performance

CHAPTER 7

DYNAMIC PROJECT MODEL

In order to represent iterative cycles and their impact on the construction performance, a dynamic project model was developed by incorporating the framework of quality and change management, and reliability and stability buffering into system dynamics models as well as schedule networking concepts. The dynamic project model can be applied to the general construction process with the ability to be customized to a specific project or activity.

In this section, we discuss general issues on system dynamics modeling, and then the generic process model, which is the backbone of the dynamic project model. Finally, the schema of the dynamic project model is introduced by briefly presenting supporting model structures.

7.1. Issues on System Dynamic Modeling

The dynamic project model developed in this research focuses on the strategic analysis as well as the operational analysis of the traditional network-based tools. The differentiation between strategic and operational analyses can be highlighted with the following example. When the project manager estimates the project completion time, subjective and informal factors for productivity such as workforce morale, schedule pressure, workers' experience level, and the technical difficulty level of a project are assumed based on his/her experience. However, if these assumptions do not match the actual state, the estimate on the project completion time will become wrong. In this case, the traditional network tools lack the ability to provide explicit analyses for this wrong schedule estimate due to their static approach. In contrast, system dynamics modeling adopted by this research focuses on the analysis of the dynamics in associated with those factors and their impact on the project performance. As a result, a dynamic modeling approach based on system dynamics can provide important strategic insights into the planning and control of construction projects.

However, the operational analysis provided by the traditional network-based tools cannot be ignored due to their capability to deal with the detailed logic of the work structure, normally, represented in a network. Therefore, integrating the operational detail of the traditional approach and the systemic view of system dynamics modeling approach can offer accurate estimations for construction projects [Rodrigues and Bowers, 1996]. Based on this recognition, the dynamic project model aims to deal with strategic issues as well as operational analysis by incorporating scheduling network concepts, such as precedence relationships, critical path, and probabilistic duration estimation, into system dynamics models.

7.2. Generic Process Model

The generic process model is developed to represent the basic construction process focusing on iterative cycles during the actual execution and it is based on the existing system dynamics project model, which has been developed by many researchers and practitioners. This existing model was extended to deal with iterative cycles by quality problems and their settlement along with 'Request For Information' (RFI) at the original DPM model.

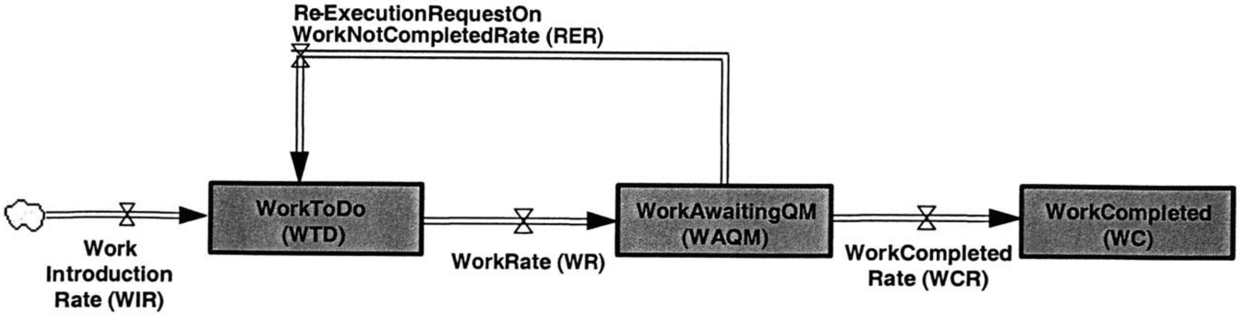
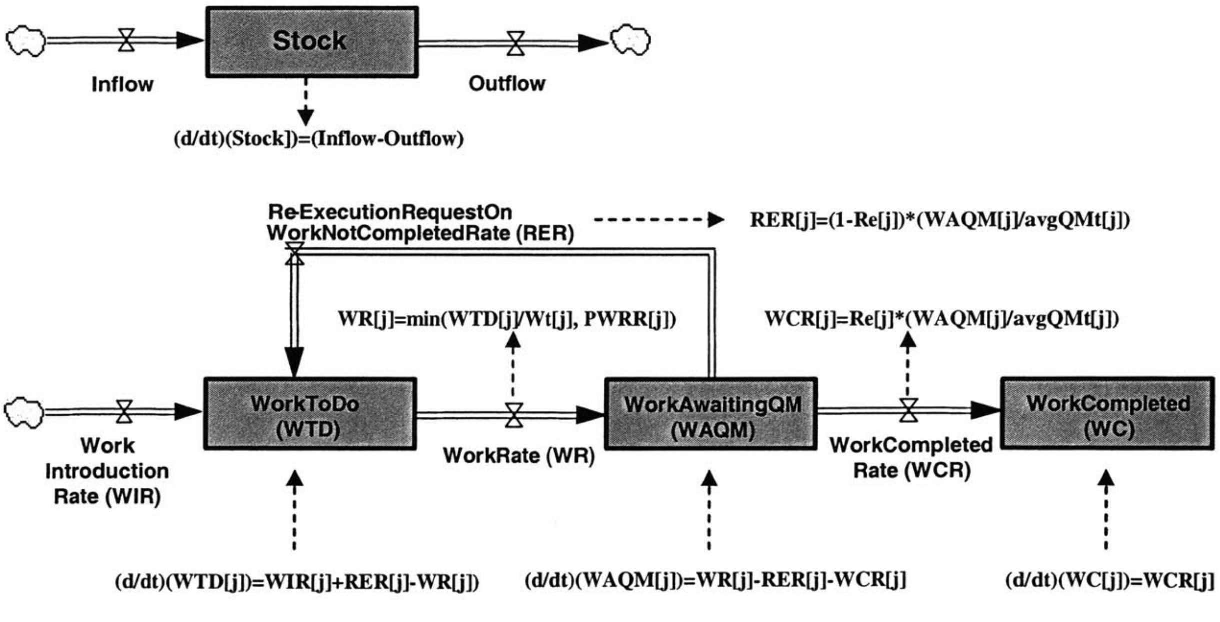


Figure 10. Generic Process Model with Quality Management Process at the Existing Project Model [Adopted from Ford & Sterman, 1997; MoonSeo Park, 2000; Lyneis et al, 2001]

Figure 10 illustrates the basic stock and flow structure of the construction process with iterative cycles. In the stock of *WorkToDo*, work is done based on normal work rate and resource availability. And, then tasks done are moved to and accumulated in the stock of *WorkAwaitingQM* where those tasks await quality inspections. If they pass inspection, the approved tasks go to the stock of *WorkCompleted* and are released to succeeding activities. Otherwise, they need to be performed again and consequently, go back to *WorkToDo*. Whether tasks pass inspection or not depends on errors, which are governed by reliability. Here, the error cycle due to uncovered error is apparently addressed by the stock and flow diagram.

Based on the above model description, equations for each stock and flow are formulated as seen in Figure 11. Each flow is constructed based on related variables and time change, and then the stock is formulated as the integral of netflow (inflow-outflow) as seen in Figure 11. By representing as the integral of the netflow, the stock can represent its current amount which is changed by time-dependent netflow.



*Note j = activity, k=preceding, and j, k ∈(1,2,3…,n}

PWRR: PotentialWorkRatefromResource Wt: WorkTime Re: Reliability avgQt: AverageQMTimeavgQMt

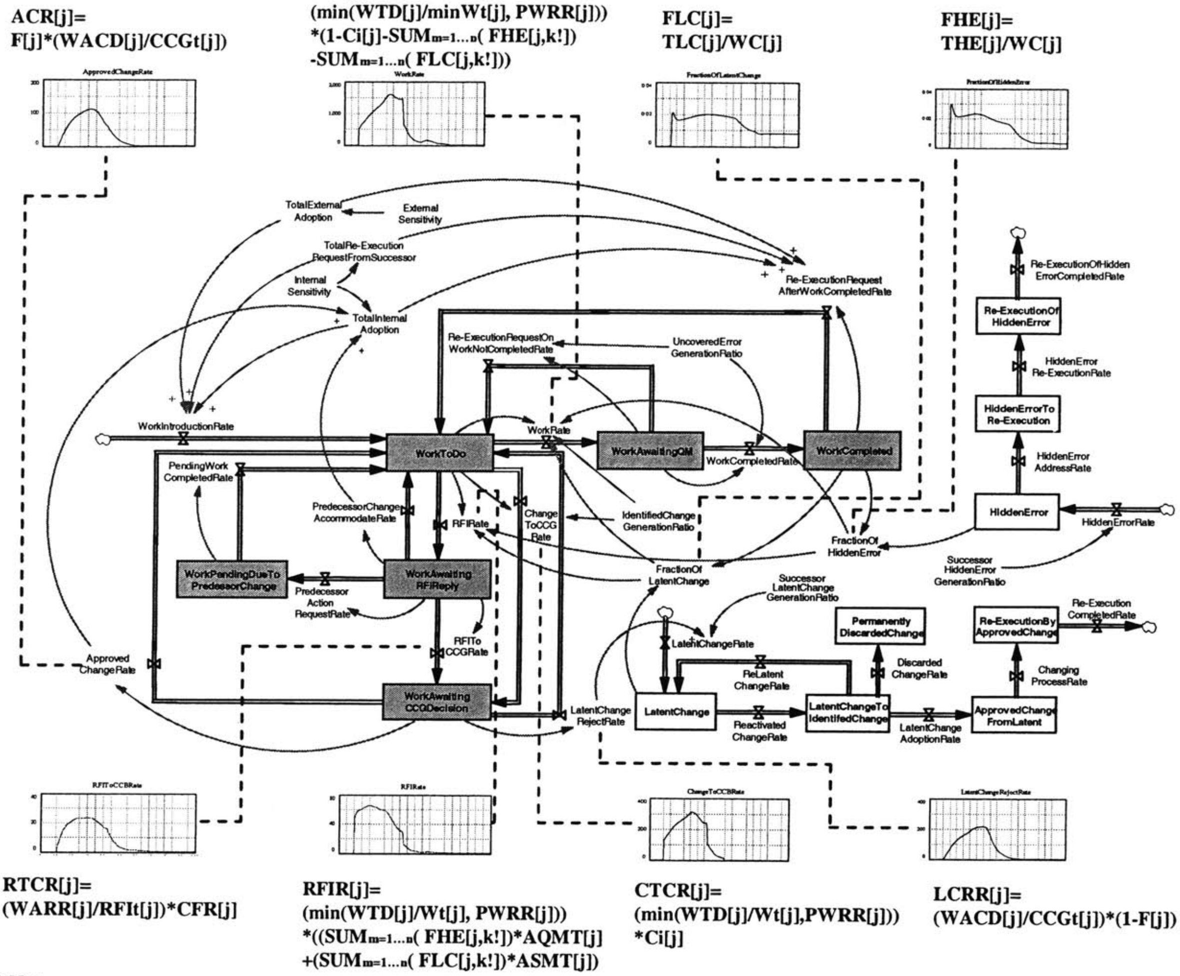
Figure 11. Generic Process Model and Equations [Adopted from MoonSeo Park, 2000]

However, the real construction process is not that simple, as we have discussed so far. Accordingly, the model structure in Figure 10 is not adequate to properly address dynamic and complex cycles involved in construction such as settlement through RFI, change request, and work multipliers. To incorporate these issues, the generic process model in Figure 10 is extended, as illustrated in Figure 12. The generic process model in Figure 12 has six main stocks (gray boxes), which represent iterative cycle processes. In addition, the co-flow structure in Figure 12 (white boxes) accounts for error and change generation in detail.

In associated with *WorkToDo*, the scope management process is conducted before the start of actual task execution. If changes are identified at this scope management process, problematic tasks go to the stock of *WorkAwaitingCCGDecision* to be discussed in the claim and change group (A). However, if a hidden error or latent change of the predecessor activity is found at this stage, it is settled down through RFI to the predecessor activity (B). In addition, in the stock of *WorkAwaitingRFIReply*, there are three ways to flow out. The first option is going back to *WorkToDo* (C), because it may be decided to accommodate the change made in the predecessor activity. If not, the successor activity can request the correction of the change (D). Therefore, the associated task accumulates in the stock of *WorkPendingDueToPredecessorChange*. It will be back to *WorkToDo* and performed after correction in the predecessor activity is completed (E). Back to *WorkAwaitingRFIReply*, the last option is to flow into the stock of *WorkAwaitingCCGDecision* (F). If it is revealed as a latent change in the predecessor activity, the manager may request the decision to the claim and change group. In short, *WorkAwaitingCCGDecision* have two inflows one from *WorkAwatingRFIReply* (F) and one from *WorkToDo* (A). At the same time, a change can be approved or rejected based on the decision of the claim and change group. If a change is rejected, it goes back to *WorkToDo* and will be performed as latent change (G). In the other case, it also goes back to *WorkToDo*, but can introduce additional work depending on sensitivity (H).

On the other hand, work multiplier can be represented by multiplying work scope with sensitivity. There are three ways in requesting additional work. First, additional work can be requested by *TotalInternalAdoption*. *TotalInternalAdoption* of the activity under study is composed of two cases; one is when change is approved by the claim and change group (I_1) and the other is when accommodating change made in the predecessor activity through RFIs (I_2). Therefore, it is required to do additional work (I_3) or re-execution on already completed tasks of the activity under study (I_4). One point to note is that internal sensitivity is used to decide the amount of newly requested work. The second case is

TotalExternalAdoption from the predecessor and the successor activity, which can generate additional work (J1) or re-execution (J2), through respective external sensitivity with the predecessor and the successor activity. Finally, as we observed earlier, when hidden errors or latent changes are discovered, the request of re-execution from the successor activity can also generate additional work through internal sensitivity (K1) as well as re-execution (K2).



*** Note**

j = activity, k=preceding, and j, k ∈(1,2,3…,n)

ACR: ApprovedChangeRate F: Feasibility WACD(t): WorkAwaitingCCGDecision CCGt: Claim&ChangeGroupDecisionTime FLC:FractionofLatentChange TLC: TotalLatentChange FHE: FractionofHiddenChange THE: TotalHiddenError RTCR(t): RFIToCCGRate RFIt: RFITime CFR:ChangeFromRFI AQMT: ActualQMThoroughness ASMT: ActualSMThoroughness CTCR:ChangeToCCGRate LCRR(t): LatentChangeRejectRate

Figure 12. Equation and Behavior of Extended Generic DPM Process Model

Having understood these mechanisms, the equations in Figure 12 can be formulated for the generic process model. In addition, illustrated graphs show the behaviors of each stock and flow during the construction period. With the extended generic process model, the dynamic project model more effectively simulates the impact of the iterative cycle on the project performance. In addition, other supporting model structures assist in addressing the relationship between iterative cycles and the project performance, which will be detailed in the following section.

7.3. Schema of the Dynamic Project Model

As illustrated in Figure 13, the schema of the dynamic project model is introduced to provide supporting model structures to the dynamic project model. It supports the dynamic project model with the project scope, resource acquisition and allocation, the project target, and the project performance. Illustrated interactions in the schema of the dynamic project model are explicitly transplanted to the dynamic project model.

In detail, before a project starts, the project has its own initial scope, resource, and target as inputs for project execution. After the project starts, the process having initially three inputs will generate performance profiles, and it reflects the status of the construction process. As the performance shows its reaction to the applied process status, the three inputs can be adjusted through the process. These adjusted inputs are transferred to the process, which will generate the different performance profiles and finally these interactions are iterated until not adjusted scope is identified and real performance is achieved.

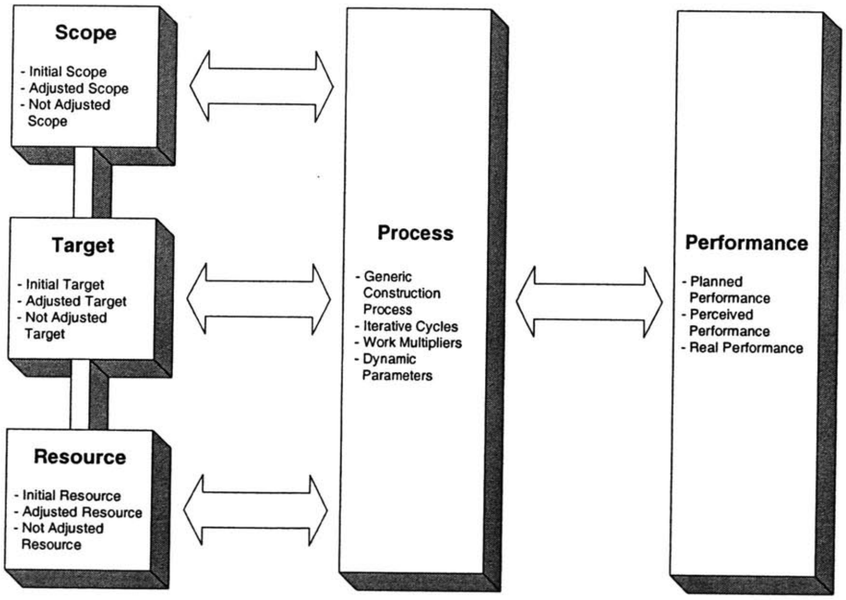


Figure 13. Schema of Dynamic Project Model
[Modified from Ford & Sterman, 1998]

CHAPTER 8

THE EFFECT OF RELIABILITY AND STABILITY BUFFERING

One of the major components of the original DPM is reliability buffering, and it provides the mechanism to update the planned schedule with a different approach than the traditional buffering as introduced earlier. In this section, reliability buffering is denoted as *reliability and stability buffering* to be able to incorporate both quality and change management issues, and is enhanced in order to effectively deal with uncertainties in concurrent design and construction.

8.1. Reliability and Stability Buffer Split

Though reliability and stability buffering has great promise for schedule protection from unpredictable events, it may be ineffective to deal with the uncertainties of successor activities, if the overlapping period is physically significant in related activities. For instance, suppose errors and changes are more generated at Zone B than Zone A of Activity A as illustrated in Figure 14. The reliability and stability buffer may not handle them effectively though it uses up the whole possible buffer size, because the covering period that the reliability and stability buffer of successor activity can handle may be only for Zone A not B. In other words, the reliability and stability buffer only has limited information of the predecessor activity.

To avoid this situation, the reliability and stability buffer split is introduced as the form of a second buffer, if the buffer which is located at the beginning of the activity can be said to be the first buffer. *The second buffer* is only activated in the case that two related activities share much overlapped period or the predecessor activity has almost been overlapped by the successor activity. If activities have a serial scheduling network such as finish-to-start relationship with positive lag, the first reliability and stability buffer can deal with the uncertainty of the successor activity without the second buffer because it has firm information on the successor activity. Otherwise, the second buffer may be activated and can supplement the first buffer in excessive concurrent development. On the other hand, in terms of location, the second buffer occurs at the time when the predecessor activity finishes. As the actual process is performed, the location of second buffer may be varied depending on the predecessor activity finish time.

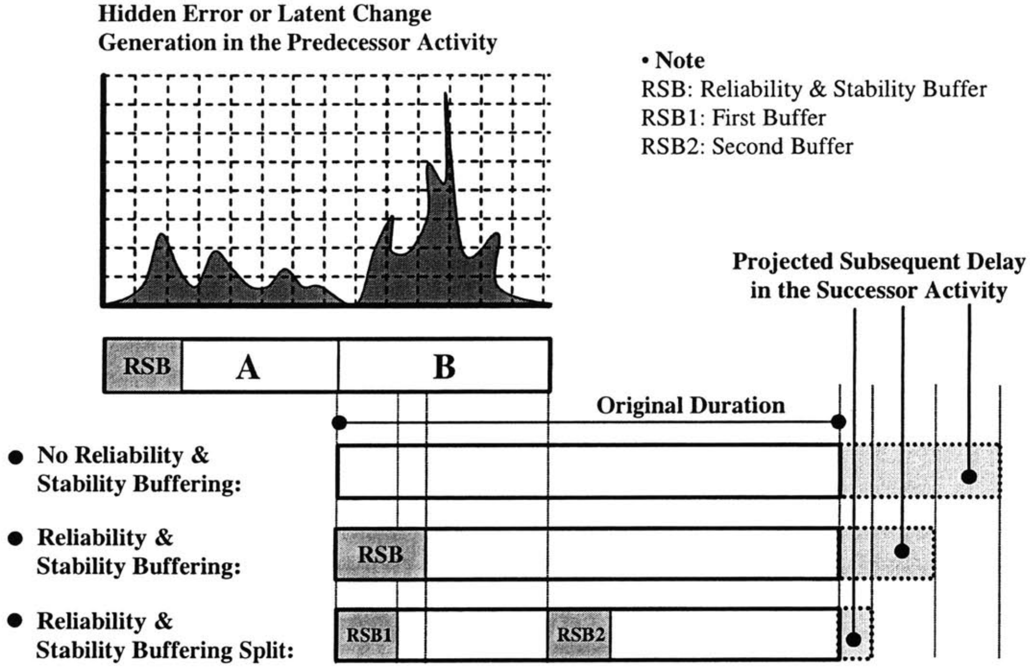


Figure 14. Reliability and Stability Buffer Split

8.2. The Effect of Reliability and Stability Buffering

Reliability buffering can effectively deal with error and change impact to prevent the possible domino effect during the construction process [Peña-Mora and Park, 2001]. In addition, the reliability and stability buffer split can have more positive impact on error and change iteration during the construction process.

At the beginning, we discussed error and change impact on performance by discovery time and location. If error or change in the predecessor activity is discovered at the successor activity, especially, after the predecessor activity finishes, it can have a strong impact on the successor activity as well as the predecessor activity. Reliability and stability buffering can reduce error or change impact and avoid the occurrence of derivative activity, even though the predecessor errors or changes are discovered at the successor activity.

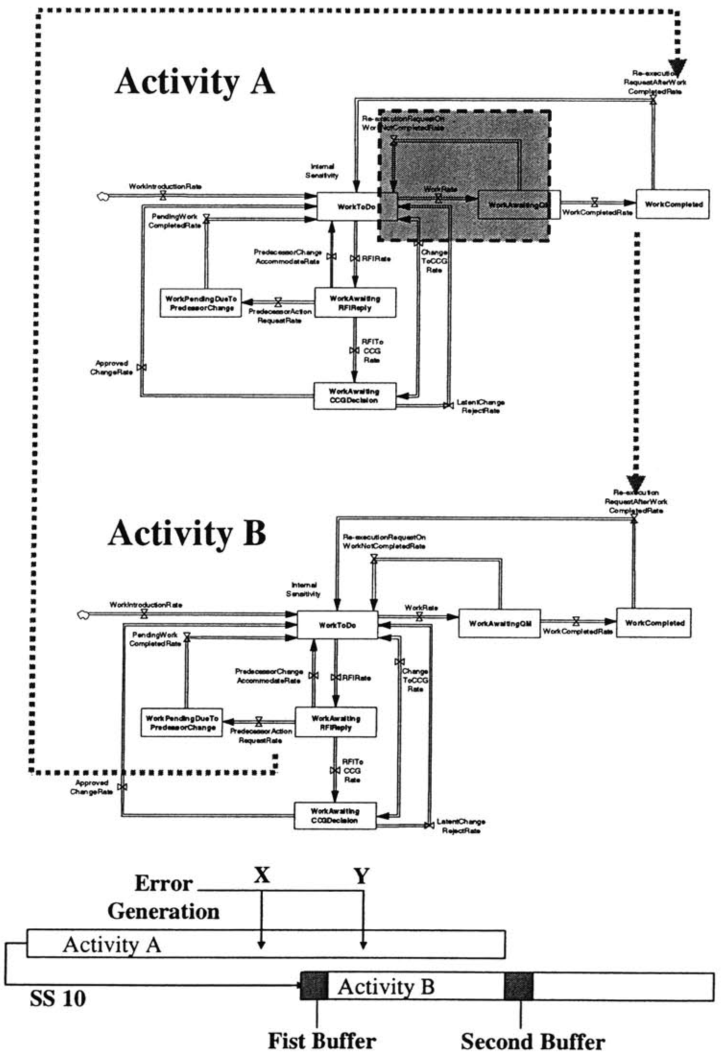


Figure 15. The Effect of Reliability and Stability Buffering

For example, assume that Activity A and B are developed concurrently having start-to-start relationship with lag 10 as illustrated in Figure 15. During Activity A's execution, if error is uncovered during the quality management process at the point of X and it requires rework at that time, it generates time delay but does not require any additional work to either activities as denoted by the gray dashed box. However, if extra work is adopted by a managerial decision, it can generate subsequent scope substitution or extra work to the successor activity. In this situation, the reliability and stability buffer in Activity B, which is located at the beginning of the activity, can make sure that the subsequent change is checked before Activity B starts. If error is generated at the point of Y, the second buffer, which is located at the time when the predecessor activity finishes, can reduce this subsequent impact on Activity B.

However, late discovery of hidden errors or latent changes in the predecessor activity may generate more serious and iterative impact on related activities with quality deterioration and a ripple effect on the successor and concurrent activities. In the same Figure 15, error is generated at the point of X, but it is found during the execution of the successor Activity B as hidden errors of the predecessor activity. In that case, it may request additional work to the predecessor activity. However, the adoption of the successor activity's requests may generate additional work to already completed tasks in the successor activity. In this case, the reliability and stability buffer provides more pre-checking time to discover hidden errors of the predecessor activity before actual execution of Activity B. Even if error is generated at the point of Y and becomes hidden error, it can be revealed by the second buffer and possibly avoid the ripple effect on the successor activity.

8.3. The Validation of Reliability and Stability Buffer

To validate the effect of proposed reliability and stability buffering for concurrent design and construction, three cases are simulated as no buffering, only first buffer, and first and second buffer case with the base scenario which is detailed in Table 3. Both Activity A and B have 60-day duration having start-to-start precedence relationship with lag 30. In buffering cases, we assume that 20% of the activity duration is used for schedule contingency and 50% of that contingency is fed as the reliability and stability buffer. In the case that the first and the second buffer are considered at the same time, respective buffer takes 50% of total reliability and stability buffer size. Therefore, if the only first buffer is considered, first buffer size is 6 days and in the case of the first and second buffer, each buffer takes 3 days for buffer size.

In this model setting, if the only first buffer is considered, it plays a role to find problems in the predecessor activity as the form of a pre-checking process. However, in the case that the first and the second buffer are considered together, it may not be realistic to stop all progressed works due to the second buffer period in terms of resource idle time and work continuity. Therefore, 50% of the work amount is assumed in the planning stage to be performed during the second buffer period. In addition, no policy parameters are considered, such as overtime and flexible headcounts, to estimate the pure effect of reliability and stability buffering.

Table 3. Model Setting and Simulation Result

Descriptions		Activity A	Activity B
Common Parameters		**Value Setting**	
Duration (Days)		60	60
Precedence Relationship		Start-to-Start 30 (Lag)	
Construction Characteristics	Production Type	Slow	Slow
	Reliability	0.9	0.9
	Stability	0.9	0.9
	Sensitivity	1	1
Quality Management Thoroughness		0.8	0.8
Scope Management Thoroughness		0.8	0.8
RFI Period (Day)		6	6
CCG Period (Day)		7	7
Buffering	Schedule Contingency	0.2	0.2
	Fraction of Buffering	NA	0.5
Specific Reliability & Stability Buffering Parameters			
Only First Buffer	Buffer Size (Days)	NA	6
First Buffer & Second Buffer	First Buffer Size (Days)	NA	3
	Second Buffer Size (Days)	NA	3
Each Case Simulation Result		**Activity Duration**	
No Buffering Case		80	108
Only First Buffer Case		71	95
First Buffer & Second Buffer Case		71	86

As seen at the bottom of Table 3, cases applying the reliability and stability buffer show better ability to reduce durations than the no buffering case. This result is mainly due to the absorption of error and change impacts of Activity A during buffer periods in Activity B. In addition, appropriate schedule pressure achieved through reduced target duration (60 → 48 days) enables improvement in the workers' productivity of both activities in the buffering applied case.

On the other hand, all simulated durations of the three cases are longer than the duration by CPM, 90 days. This can be explained by the lack of considerations for error and change iterative cycles on CPM while the dynamic project model of DPM can consider their impact on performance based on activity characteristics as discussed earlier. In addition, the duration of CPM may be difficult to achieve accurately, if the actual conditions of the activities were materialized [Park and Peña-Mora, 2001].

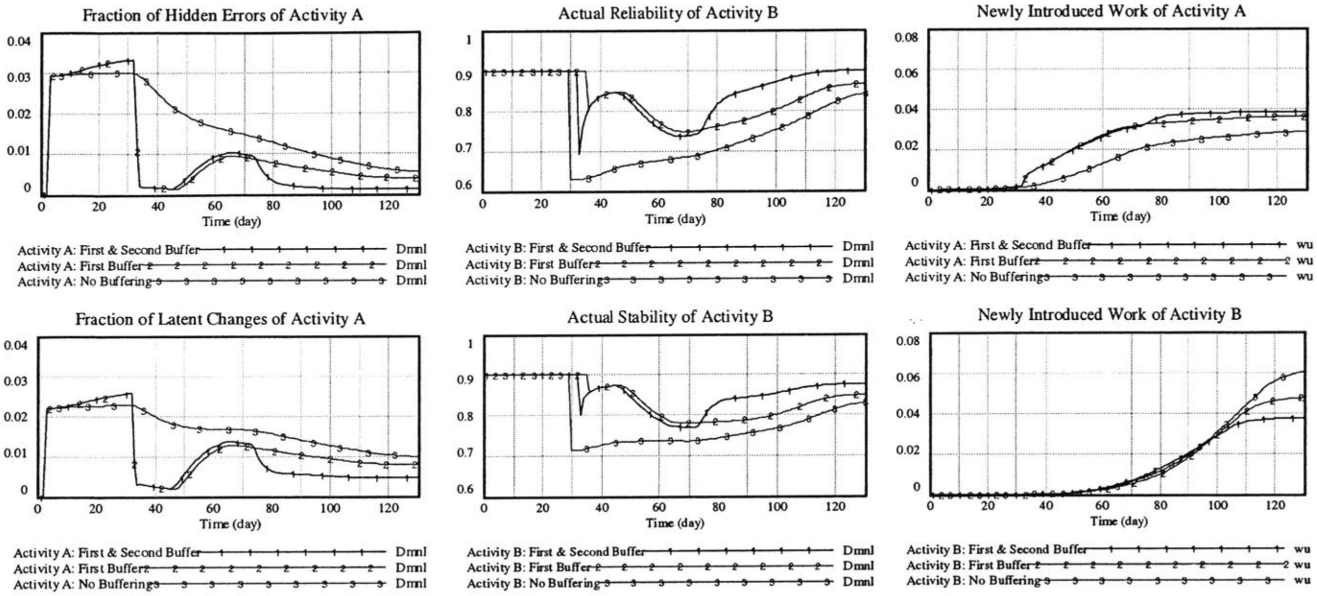


Figure 16. Simulation Result of No Buffering and Buffering Adaptation

Meanwhile, the simultaneous consideration of the first and the second buffer reduces duration of Activity B as much as 22% from the no buffering case (108 → 86 days) and 9% from the only first buffer case (95 → 86 days). This result can be achieved in that the second buffer can deal with error and change impact of the later part of Activity A, where the first buffer can't. In Figure 16, hidden errors and latent changes of Activity A are reduced due to the discovery of them at the first and second buffer period. These reduced fractions of hidden errors and latent changes allow reliability and stability to avoid their harmful ripple effects on performance. In addition, though newly introduced work of Activity A in the buffer case are more numerous than the no buffering case, their early adoption makes newly introduced work of Activity B fewer than the no buffering case. This effect finally contributes to avoiding 'the last minute syndrome'.

The one point to note is the importance of production type for the second buffer. Task production type describes the ability of the predecessor activity's physical units to provide finalized information. For example, fast production type means that the most amount of work is performed at early stages and then it slows down at later stages with a non-linear behavior. On the other hand, slow production type denotes the reverse case as illustrated in Figure 17.

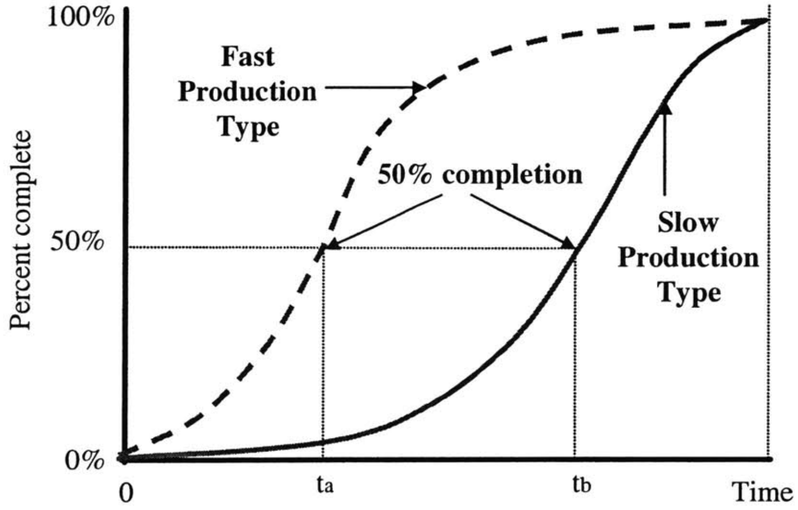


Figure 17. Fast and Slow Production Type

In this simulation cases in Table 3, the second buffer contributes to reducing the impact of errors and changes with both slow production rates. However, if one of the activities has fast production rate, the second buffer may not handle the iterative cycles as detailed in Table 4. For example, if the predecessor activity has fast production rate, the work amount of the predecessor activity is relatively small during the time between the end of the first buffer and the start of the second buffer, so the second buffer may deal with only small portions of errors and changes accordingly. On the other hand, if the successor activity has fast production rate, relatively large portion of tasks is already completed before the second buffer start. Therefore, significant amount of re-execution is required to the already completed tasks in the successor activity without the effect of early adoptions in reliability and stability buffering. Therefore, if one of the activities that have precedence relationships has fast production type, the second buffer may not provide much better effect than the only first buffer case.

Table 4. Second Buffer vs. Production Type

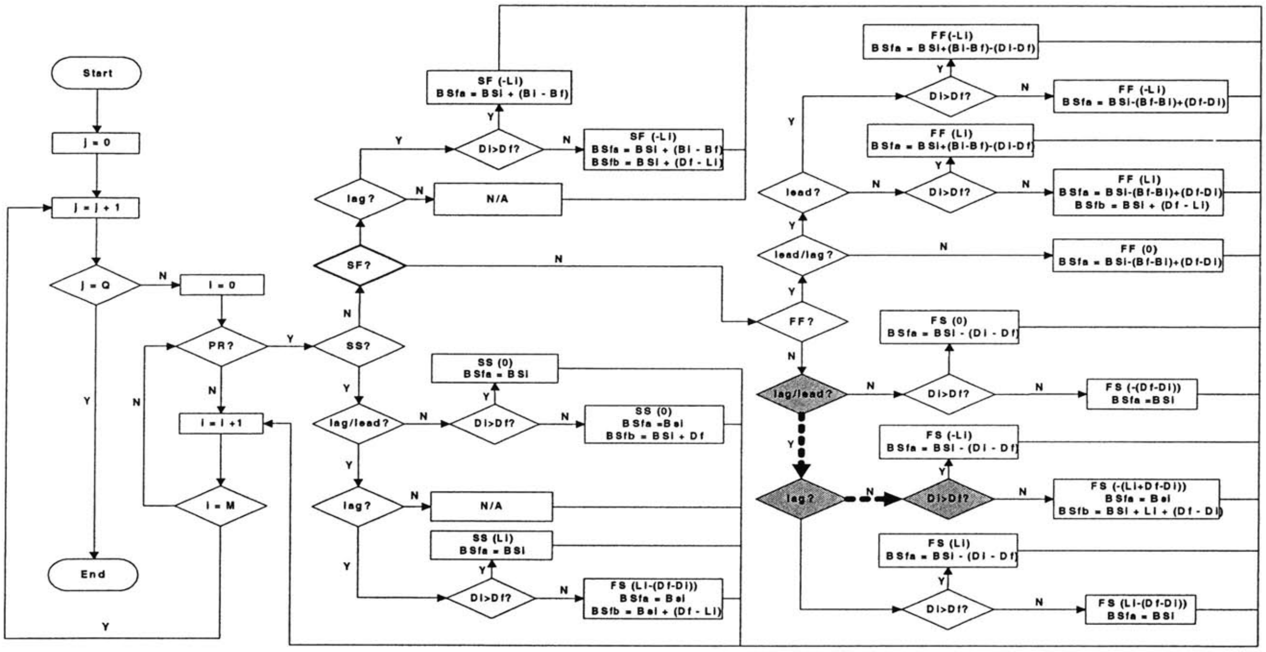
Activity	P	S	P	S	P	S
Production Type	Slow	Fast	Fast	Slow	Fast	Fast
No Buffering	79	108	78	110	78	106
Only First Buffer	71	92	71	84	71	83
First & Second Buffer	71	88	71	80	71	81

* **Note** - P: Predecessor Activity S: Successor Activity Unit: Days

8.4. Schedule Update by Reliability and Stability Buffering

One of the important characteristics in reliability and stability buffering is the dynamic behavior of its location and size based on the information obtained from the actual performance and the forecast of the remaining performance through a simulation process. Consequently, the planned schedule is updated by adopting a dynamic behavior of reliability and stability buffering. The patterns of buffer location and size are varied based on precedence relationships and the simulation result of the predecessor activity duration and characteristics.

Figure 18 represents as the flow chart of how the schedule is updated. General updating procedure starts with the existence of precedence relationships and their type if any. And then it investigates the existence of lag or lead and finally suggests new relationships with the updated buffer location and size comparing the magnitude of initially planned duration (D_i) and forecasted duration (D_f). For example, if Activity A and B have finish-to-start relationship having lead 5, it follows the dashed line and grayed decision diamonds in Figure 18. The initially planned duration of exemplified activity is 20 days, but the forecasted duration is 25 days, having delay. In this case, the initial relationships, lead 10, will be added as much as Activity A's delay, consequently, the lead will be 15 days. This is because the start time of the reliability and stability buffer is not affected by the predecessor activity's delay. If a static approach is applied to the schedule, the activity start time will be changed accordingly, but reliability and stability buffering forces the successor activity to be started though the predecessor activity is delayed. If the initially planned reliability and stability buffer size is bigger than the magnitude of the predecessor delay, the location and size are the same as the planned buffer. However, if the initially planned buffer size is less than the predecessor activity's delay, only the size is extended to catch up with the predecessor's delay based on the simulation result. Having lead in the finish-to-start relationship makes some periods to be overlapped. In this case, the reliability and stability buffer split is applied to handle concurrent design and construction. The location of the second buffer is the projected finish time of the predecessor activity. In addition, the size is varied based on the forecasted activity performance and it can be zero in spite of concurrent design and construction if the forecasted performance in the successor activity is quite stable to follow the planned schedule. In this case, the magnitude of delay (D_f-D_i) and the size of the initially planned lead are added to the initially planned reliability and stability buffer - first buffer - location.



*** Note**

Q: Number of Activities in the Project
Di: Initially Planned Duration
Df: Forecasted Duration
Li: Initially Planned Lead or Lag

M: Number of Predecessors of an Activity
Bi: Initially Planned Buffer Size
Bfa: Forecasted First Buffer Size
Bfb: Forecasted Second Buffer Size

PR: Precedence Relationship
BSfa: Forecasted First Buffer Start Time
BSfb: Forecasted Second Buffer Start Time
BSi: Initially Planned Buffer Start Time

Figure 18. Schedule Update by Reliability and Stability Buffering

CHAPTER 9

THE IMPLEMENTATION OF DPM

So far, several concepts and components that are explicitly embedded in DPM have been discussed. Having all components, DPM as an integrated methodology is introduced in this section, including its collaboration scheme.

9.1. System Overview

Figure 19 illustrates and summarizes how integrated components are interacted with each other and how DPM integrates all components. First, DPM input interface implements the concept of the smart cell on the form of Dependency Structure Matrix (DSM) [Eppinger, 1997], as seen in Figure 20. A smart cell represents the cell which includes all information about each activity characteristics. The role of a smart cell is to receive data input from users and to organize it so that it can be effectively utilized by DPM functions, especially, the dynamic project model. Though a main objective of DSM is to provide ordering of the sequence with acquired information, a smart cell utilizes DSM as input interface only adopting its format. A smart cell shows the inter-relationships among activities with a mark X and each mark contains information on each activity's characteristics such as production type, reliability, sensitivity, and stability. In addition, a smart cell incorporates other network tools' information having inputs such as the probabilistic duration spectrum from Program Evaluation and Review Technique

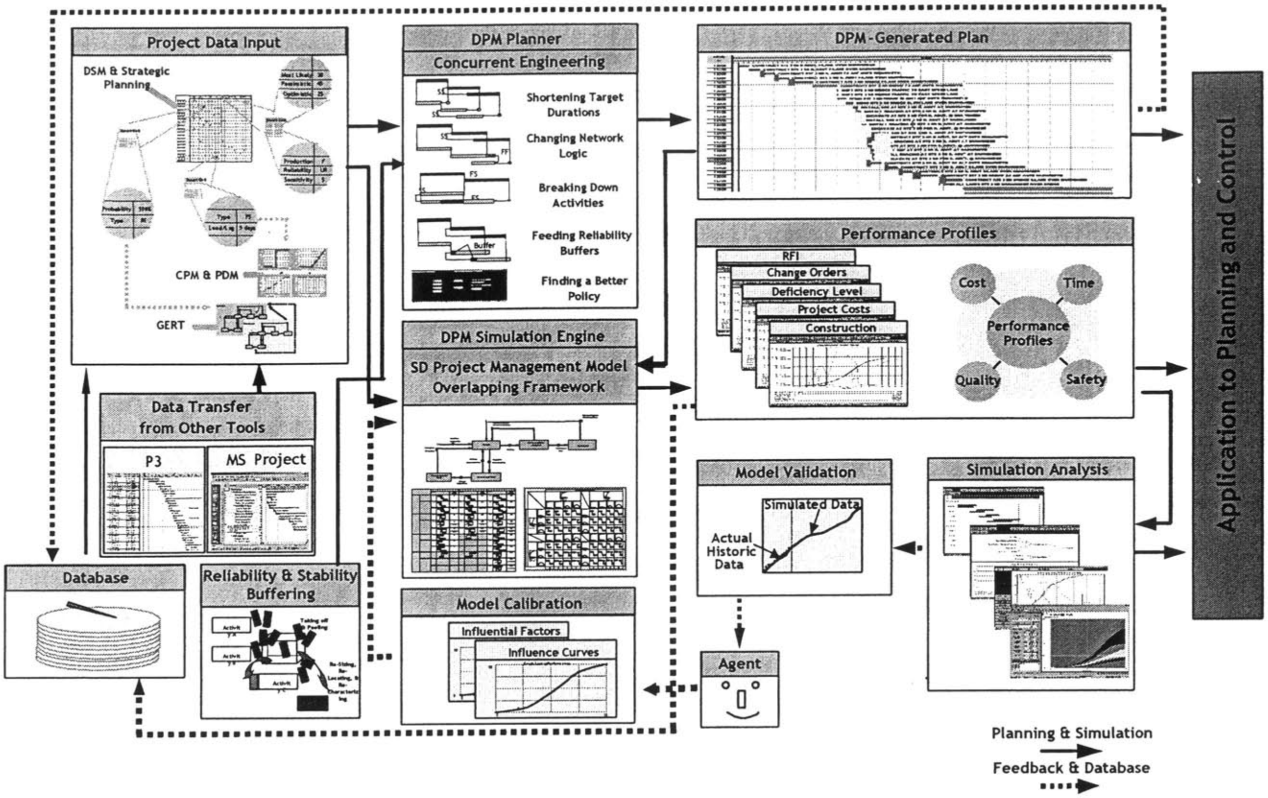
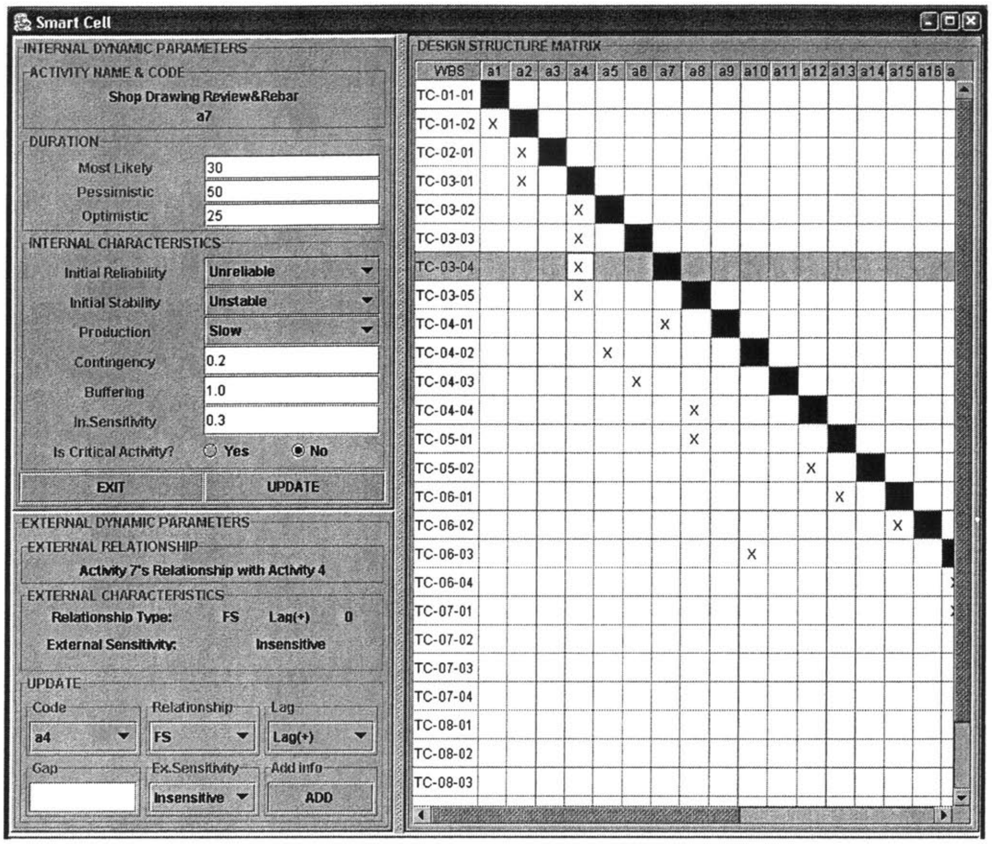


Figure 19. DPM Integrated Components
[Extended from MoonSeo Park, 2000]

(PERT), precedence relationships from Critical Path Method (CPM) and Precedence Diagramming Method (PDM), and iteration factors for simulation with similar concept as Graphical Evaluation and Review Technique (GERT), Queue-Graphical Evaluation and Review Technique (Q-GERT), and Simulation Language for Alternative Modeling (SLAM). For applicability in real application, a smart cell transfers data from P3e™ (Primavera Project Planner for Enterprise by Primavera Co.) and Microsoft Project (Microsoft Corporation), which are widely used in the construction industry, and extracts the necessary information for DPM. In addition, a smart cell stored all input to Oracle 9i (Sun Microsystems) database.

After a smart cell receives all information from the user, it transfers it to DPM planner. DPM planner generates the schedule plan and possible policies with reliability and stability buffering based on user's selection and concurrent engineering concept.



INTERNAL DYNAMIC PARAMETERS

ACTIVITY NAME & CODE

Shop Drawing Review&Rebar
a7

DURATION

Most Likely: 30
Pessimistic: 50
Optimistic: 25

INTERNAL CHARACTERISTICS

Initial Reliability: Unreliable
Initial Stability: Unstable
Production: Slow
Contingency: 0.2
Buffering: 1.0
In.Sensitivity: 0.3
Is Critical Activity? ○ Yes ● No

EXTERNAL DYNAMIC PARAMETERS

EXTERNAL RELATIONSHIP

Activity 7's Relationship with Activity 4

EXTERNAL CHARACTERISTICS

Relationship Type: FS Lag(+) 0
External Sensitivity: Insensitive

UPDATE

Code: a4
Relationship: FS
Lag: Lag(+)
Gap:
Ex.Sensitivity: Insensitive
Add info: ADD

DESIGN STRUCTURE MATRIX

WBS	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a
TC-01-01	■																
TC-01-02	X	■															
TC-02-01		X	■														
TC-03-01		X		■													
TC-03-02				X	■												
TC-03-03				X		■											
TC-03-04				X			■										
TC-03-05				X				■									
TC-04-01							X		■								
TC-04-02					X					■							
TC-04-03						X					■						
TC-04-04								X				■					
TC-05-01								X					■				
TC-05-02												X		■			
TC-06-01													X		■		
TC-06-02															X	■	
TC-06-03										X							■
TC-06-04																	X
TC-07-01																	X
TC-07-02																	
TC-07-03																	
TC-07-04																	
TC-08-01																	
TC-08-02																	
TC-08-03																	

Figure 20. Smart Cell

Having generated plan and policy, the DPM simulation engine based on the system dynamics model simulates its effectiveness based on necessary factors from the smart cells. The result of DPM simulation provides the performance profiles capturing the impact of unanticipated events on project performance. In addition, various simulation analyses are performed and provided with an appropriate format depending on planning purpose as a graphical representation. This result is also stored in the database for future use.

Moreover, the implementation for project control is also provided. During the actual execution, DPM can get the input for current progress and compare it with the stored simulated performance and, consequently, provide the future policy for better project profiles. Or it re-simulates and forecasts the project performance using the actual information it has had so far. Therefore, once simulated data are obtained, they are tuned up for more accurate planning and projection and provide the learning mechanism, by comparing the simulated performance with the actual performance.

9.2. Collaboration Scheme

Quickly and reliably sharing information and finding a solution to a problem are the major issues in terms of the speed and quality of a project. As a project becomes more complex and even occurs at geographically distributed places, these issues become more crucial than ever. In this context, DPM aims to provide a platform where information can be effectively communicated with very few restrictions.

To implement easy access from any place and at any time, DPM is developed to run within the web environment. Web-based DPM can contribute to easy access to all involved parties with permissions without special software installation on local computers. In addition, DPM is developed to be accessed from multiple devices such as PC, PDA, and Java enabled phones with both wire and wireless networking connections in order to free the users from any device restriction. Due to the nature of construction, construction crew does not always get access to a desktop computer on the site, allowing only the use of wireless or portable devices. For this reason, effective monitoring and controlling require a system that can overcome the dependency of information on a desktop computer [Peña-Mora and Dwivedi, 2000]. Therefore, the DPM developed for the use of multiple devices makes it possible for engineers on the site to collaborate with a project manager in the headquarter and designers in their office by implementing collaboration scheme as seen in Figure 21.

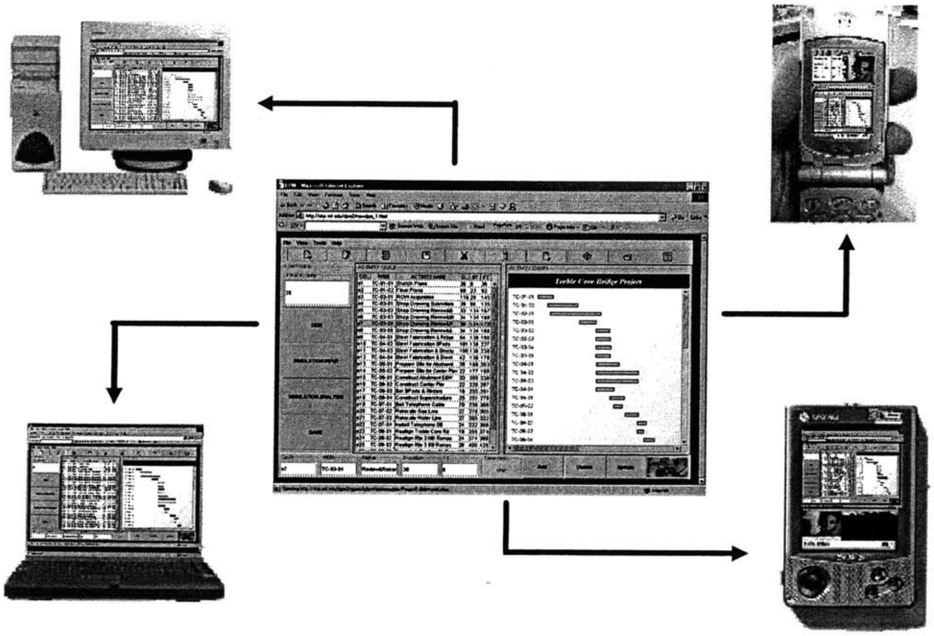


Figure 21. Collaboration Scheme with Same View

The system architecture is developed based on three-tier architecture as seen in Figure 22. This type of architecture clearly can progress to the next stage by making distinctions between the presentation layer, application layer, and data management layer and can support reliability, availability, and increased security of system [Peacock, 2000]. For example, Java client-side application in DPM only provides graphical user interface as a Java Applet and does not involve actual processes in the application server. In addition, DPM system has a capability to distribute remotely its objects for components' communication using Java technology. To connect and transfer their objects, Java Remote Method Invocation (RMI) (Sun Microsystems) and Java Data Base Connectivity (JDBC) (Sun Microsystems) are used as object-based middleware. Among several development tools, Java language (Sun Microsystems) is adopted due to its cross platform portability, which makes it an ideal tool for an Internet based collaborative application. In addition, the evolution of Micro-Java for handheld devices enables Java applications to be used or all kinds of handheld devices without the limitation of operating systems.

In detail, Oracle 9i and SQL Server (Microsoft Corporation) are used for DPM database. DPM adopts Oracle 9i as the main database because it enables Java based database application making it an ideal tool for database centric applications, and SQL Server is for P3e data storage. Java server-side application connects and manages these two databases using JDBC. For example, if it is requested to convert from P3e to DPM, Java server-side application extracts data from SQL Server and transfers to Oracle 9i database. In addition, it also sends user input and simulated results to Oracle for future use.

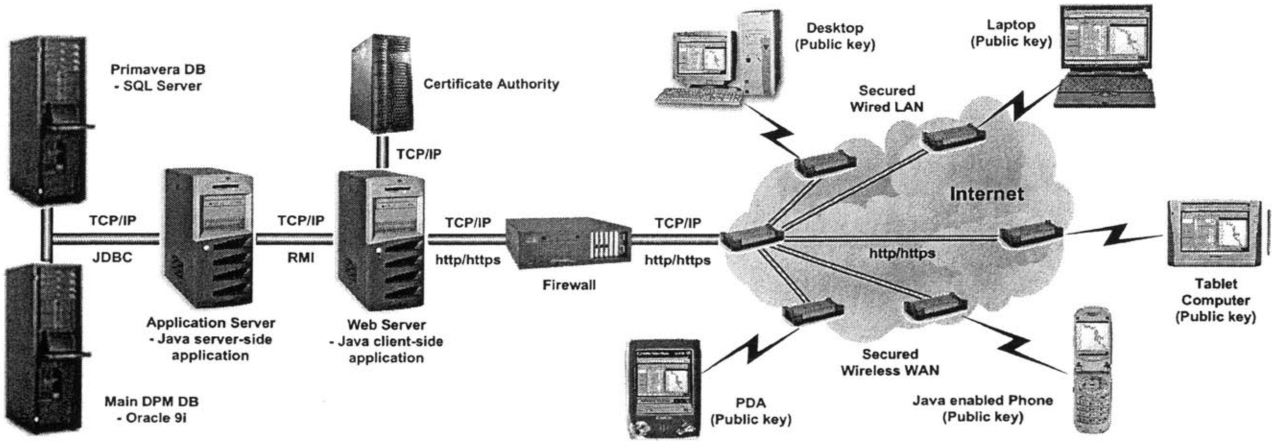


Figure 22. DPM System Architecture

Moreover, Java server-side application has modules to operate the simulation engine that controls Vensim™ (Ventana Software, Inc.), which is a system dynamics modeling software and has a dynamic project model for DPM. These modules allow DPM to simulate system dynamics model through Vensim DLL (Ventana Software, Inc.), which enables other programming languages to control Vensim model. Finally, Java client-side application takes care of only graphical user interface for a data input and simulated data output. As illustrated in Figure 23, Java Applet shows project profiles by combining simulated result with other network-based tools.

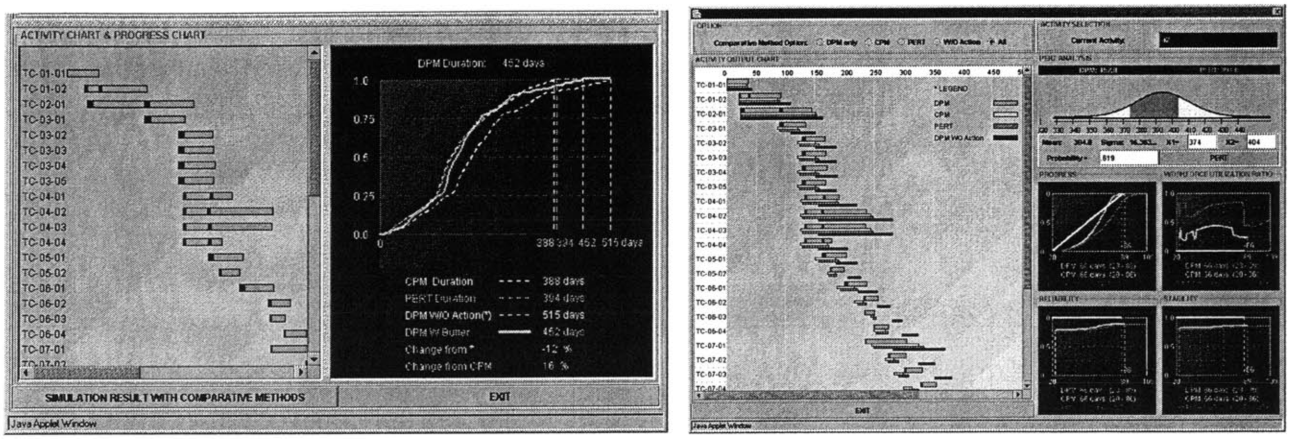


Figure 23. Project Profiles through Java Applet

CHAPTER 10

A CASE STUDY

In order to examine the applicability of DPM in real-world construction and to demonstrate its capability as integrated methodology, DPM is applied to a bridge project that is a part of a $400 million Design/Build/Operate/Transfer (DBOT) project awarded to Modern Continental Companies, Inc.. The aim of this project is for roadway improvements along State Route 3 from its intersection with State Route 128 in Burlington, MA north to its terminus at the New Hampshire border as seen in Figure 24.

The expected project completion date is February, 2004 with 42 month's span from design to construction completion. The project scope is composed of widening the 21 mile of the state roadway, and renovating 15 underpass and 12 overpass bridges among total 27 bridges. In this paper, the Treble Cove Road Bridge construction, which is one of the overpass bridges, is presented as a case project. It consists of 42 design and construction activities at a level 2 schedule. This case project provides a good opportunity to the effectiveness of DPM in real world, having activities similar to other 26 bridge construction projects.

To investigate the case project, several policy options are selected in order to provide real-project settings, according to the options that are used in the case project. For example, the case project adopts

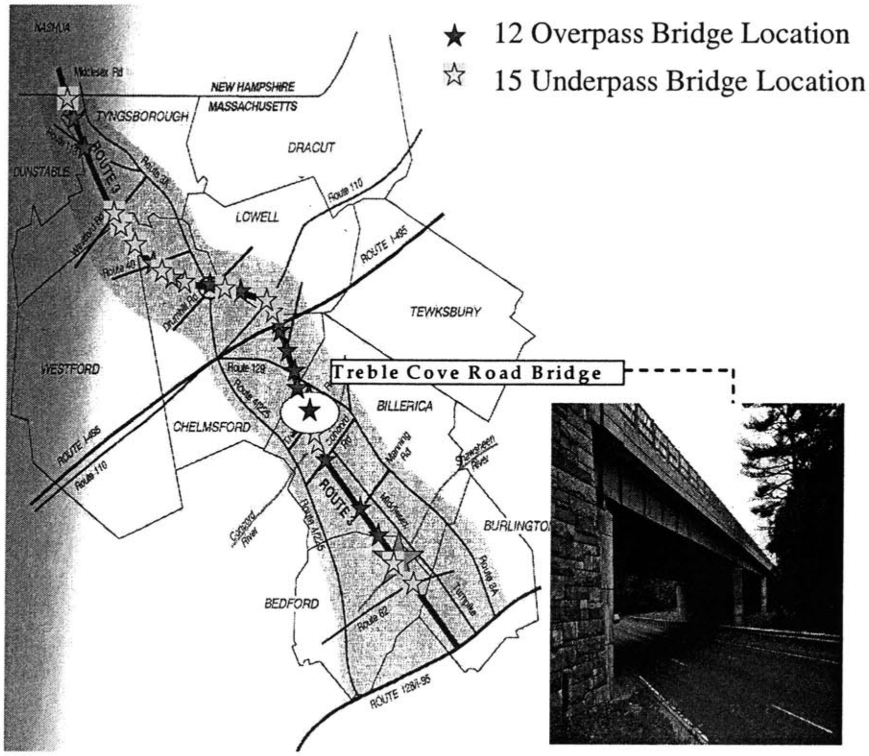


Figure 24. Route 3 North Case Project Location

overtime and flexible headcount. For example, if the manager perceives that the schedule is delayed, he/she can adopts overtime or add more workers to the delayed activity. In addition, quality and scope management thoroughness are estimated as 0.9. That means their overall thoroughness is projected to leave unidentified 10% hidden errors or latent changes during actual execution. However, these options are not applied to *CPM Case* and *PERT Case*, because they don't consider these variables to calculate the project duration.

Simulated durations for all cases are listed in Table 5. *No Action Case* is a predicted actual duration with applied options, and is used as a base case for comparison. *CPM Case* and *PERT Case* durations are derived from DPM's another functionality to calculate network schedule estimation. Finally, *DPM Case* is simulated with reliability and stability buffering.

Table 5. Project Duration of the Case Project

Case	Duration (Days)	Deviation from No Action		Policy Option (No Action & DPM)
		Days	Percentage	
No Action	515	-	-	Overtime
CPM	388	-127	-25%	Flexible HC
PERT	394	-121	-23%	QM Thoroughness = 0.9
DPM	452	-63	-12%	SM Thoroughness = 0.9

Though *CPM Case* and *PERT Case* provide relatively short durations, *No Action Case* represents the case project is significantly delayed. This implies that a lot of iterative cycles and their impacts affect the construction sequences but are not caught in CPM and PERT. Actually, the case project was delayed in a way that *No Action Case* shows. The design development in the Treble Cove Bridge Project is significantly delayed and, consequently, the construction team is working to address these issues.

DPM Case with reliability and stability buffering shows that it can reduce the duration as much as 12% from *No Action Case* as seen in Figure 23 and Table 5. Applied reliability and stability buffering contributed to reduce the impact of iterative cycles of predecessor activities having early adoption. Therefore, the reduction of hidden errors and latent changes allows reliability and stability not to deteriorate as activities progress. In other words, early adoption due to reliability and stability buffering prevents errors and changes from being transferred to successor activities. In addition, this reduction makes it possible to effectively utilize the given resources without idle time and waste.

CHAPTER 11

CONCLUSIONS

This research presents an integrated methodology to overcome uncertainties and complexities in concurrent design and construction processes. Today's wide adoption of concurrent design and construction makes projects difficult to handle by traditional network-based tools mainly due to their static view of the project. In an effort to address these issues, DPM has been developed with the dynamic project model based on system dynamics, in order to capture various dynamic relationships and feedbacks during the design and construction processes. The proposed framework for quality and change management is used as the backbone of the dynamic project model, and reliability and stability buffering is enhanced and adopted by DPM in order to effectively reduce the sensitivity to iterative cycles. The research results from various simulation scenarios and a case study show that DPM can help to provide robust planning and control actions, protecting the process against increased uncertainties and analyzing various dynamic impacts of construction feedbacks and iterative cycles. Furthermore, the system adopting web technology can contribute to the improvement of productivity by sharing fast and reliable information without the limitations of space, time, or devices.

11.1. Potential Impact

The proposed framework for quality and change management can provide the basic understanding of the dynamic behavior of iterative cycles during actual execution. Therefore, it can be a way to enhance change management by capturing the insight of iterative cycles that may occur during concurrent design and construction. On the other hand, in terms of the industry impacts, generated policy guidelines by web-based DPM with the adoption of the traditional network tools allow managers or site engineers to easily analyze unpredictable incidents and their impact on performance without specific knowledge of the simulation system or physical limits.

11.2. Further Work

In the proposed framework, change management is an ongoing research effort. Although this research presents how change can be identified and transferred during actual execution stage with simple numerical expressions, further simulation and experimentation are required to establish the robust mathematical formulations between variables for the change management process. For example, though stability is used here as a general concept to filter out possible changes in a given scope, it needs to be specified and formulated such as stability for material, workers, environment, and technology with numerical expressions based on each characteristic. This property will provide DPM with robust and diverse applicability to real-world projects. The other effort is to enhance DPM simulation capability, applying probabilistic variance of design and construction performance to the dynamic project model with numerical relationships. Although DPM aims to provide the operational view in the system dynamics approach, the adoption of the probabilistic approach can enhance DPM analysis to be more reliable and realistic.

REFERENCE

1. Awad H., Richard C., Pehr P., and Erik N. (2001), "Quantitative Definition of Projects Impacted by Change Orders." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, January/February, 2002, Vol. 128, No. 1, pp. 57-64.

2. Cooper, K. (1993), "The Rework Cycle: Why Projects Are Mismanaged." PM NETwork Magazine, PMI, Drexel Hill Pennsylvania, February, 1993, pp. 25-28.

3. El-Rays, K. and Maselhi, O. (2001), "Optimizing Resource Utilization for Repetitive Construction Projects." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, January/February, 2001, Vol. 127, No. 1, pp. 18-27.

4. Eppinger, S. (1997), Three Concurrent Engineering Problems in Product Development Seminar, Sloan School of Management, MIT, Cambridge, MA.

5. Ford, D. and Sterman, J. (1998), "Dynamic Modeling of Product Development Processes." System Dynamics Review, System Dynamics Society, Albany, NY, Spring, 1998, Vol. 14, No. 1, pp. 31-68.

6. Ibbs, C. (1997), "Quantitative Impacts of Project Change: Size Issues." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, September, 1997, Vol. 123, No. 3, pp. 308-311.

7. Kartam, N. (1996), "Making Effective Use of Construction Lessons Learned in Project Life Cycle." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, March, 1996, Vol. 122, No. 1, pp. 14-31.

8. Lee, S., Peña-Mora, F., and Park, M. (2003), "The Impact of Dynamic Quality and Change Management on Performance in Large Scale Concurrent Construction Projects" Construction Congress Research Conference (Presentation Slot), ASCE, March 19-21, 2003, Hawaii.

9. Love, P. (2002), "Influence of Project Type and Procurement Method on Rework Costs in Building Construction Projects." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, February, 2001, Vol. 128, No. 1, pp. 18-29.

10. Microsoft Corp. (2000), Microsoft Project, <http://www.microsoft.com/Office/Project/PRK/2000>, Last Visit: September, 2002.

11. Microsoft Corporation. (2002), Microsoft SQL Server, <http://www.microsoft.com/sql/>, Last Visit: November, 2002.

12. Mokhtar, A., Bédard, C., and Fazio, F. (2000), "Collaborative Planning and Scheduling of Interrelated Design Changes." Journal of Architectural Engineering, ASCE, Reston, Virginia, June, 2000, Vol. 6, No. 2, pp. 66-75.

13. Ng, W., Khor, E., and Lee, J. (1998), "Simulation Modeling and Management of Large Basement Construction Project." Journal of Computing in Civil Engineering, ASCE, Reston, Virginia, April, 1998, Vol. 12, No. 2, pp. 101-110.

14. Oliva, R. (1996), "A Dynamic Theory of Service Delivery: Implication for Managing Service Quality." Doctoral Thesis, Sloan School of Management, MIT, Cambridge, MA, June, 1996, pp.23-38.

15. Oracle Corp. (2002), Oracle 9i Database, <http://otn.oracle.com/products/oracle9i/content.html>, Last Visit: October, 2002.

16. Park, M. (2001), "Dynamic Planning and Control Methodology for Large-Scale Concurrent Construction Projects." Doctoral Thesis, Department of Civil & Environmental Engineering, MIT, Cambridge, MA, June, 2001, pp. 37-62.

17. Peacock, R. (2000), "Distributed Architecture Technologies." IT Pro May | June 2000, IEEE, Piscataway, NJ, Vol. 2, Issue 3, pp. 58-60.

18. Peña-Mora, F. and Dwivedi, G. (2001), "A Multiple Device Collaborative and Real Time Analysis system for Project Management in Civil Engineers." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, January, 2002, Vol. 16, No. 1, pp. 23-38.

19. Peña-Mora, F. and Li, M. (2001), "A Robust Planning and Control Methodology for Design-Build Fast-Track Civil Engineering and Architectural Projects." Journal of Construction Engineering and Management, ASCE, Reston, Virginia, January/February, 2002, Vol. 127, No. 1, pp. 1-17.

20. Primavera System Inc. (2001), Primavera Project Planner, <http://www.primavera.com>, Last Visit: August, 2002.

21. Richardson, G. (1985), "Introduction to the System Dynamics Review", System Dynamics Review, Albany, NY, Summer, 1985, Vol. 1, No. 1, pp. 1-3.

22. Rodrigues, A. and Bowers, J. (1996), "System Dynamics in Project Management: A Comparative Analysis with Traditional Methods.' System Dynamics Review, System Dynamics Society, Albany, NY, Summer, 1996, Vol. 12, No. 2, pp. 121-139.

23. Safoutin, M. and Smith R. (1998), "Classification of Iteration in Engineering Design Processes." ASME Design Engineering Technical Conference, September 13-18, 1998, Atlanta, GA, ASME, New York, NY, pp. 1-11.

24. Shin, K. and Molenaar, K. (2000), "Prediction of Construction Disputes in Change Issues." Construction Congress VI, February 20-22, 2000, Orlando, FL, ASCE, Reston, Virginia, pp. 534-542.

25. Sterman, J. (1992), "System Dynamics Modeling for Project Management." Sloan School of Management, MIT, On-line Publication, <http://web.mit.edu/jsterman/www/>, Last Visit: October, 2002.

26. Sterman, J. (2000), "Business Dynamics: System Thinking and Modeling for a Complex World." McGraw-Hill Companies, New York, NY, pp.55-61, 469-511, and 587-595.

27. Sun Microsystems. (2000), "Java Remote Method Invocation (RMI)", <http://java.sun.com/products/jdk/rmi/>, Last Visit: April, 2002.

28. Sun Microsystems. (2000), "What is JAVA 2 Platform, Micro Edition", <http://java.sun.com/features/1999/06/j2me.html>, Last Visit: May, 2002.

29. Trauner, T. (1992), "Managing the Construction Project." John Wiley & Sons, Inc., New York, NY, pp. 126-142.

30. Ventana Software Inc. (2001), Vensim Software, <http://www.vensim.com/software.html>, Last Visit: July, 2002.