RELIABILITY AND STABILITY BUFFERING APPROACH IN CONCURRENT DESIGN AND CONSTRUCTION PROJECTS

SangHyun Lee[1], Feniosky Peña-Mora[2], and Moonseo Park[3]

ABSTRACT

Current construction projects often need to reduce the development time while keeping their quality and budget. This increased demand for reducing development time has introduced fast tracking and concurrent engineering into the construction industry. However, the adoption of these techniques can make the process more uncertain and complex than the traditional sequential design and construction process. In this paper, we focus on iterative cycles due to error and change as the main source of uncertainty and complexity. To deal with this issue, Reliability and Stability Buffering is presented as a mechanism to reduce the impact of iterative cycles by using a simulation-based approach and different buffer locations and sizes from those used in traditional contingency buffering. Early adoption of errors and changes identified by the proposed buffering approach can help to minimize their ripple effect on the later stages of the project.

KEY WORDS

Concurrent Design and Construction, Reliability and Stability Buffering, Quality and Change Management, Construction System Dynamics, Simulation.

---

[1] PhD Candidate, Dept. of Civil and Envir. Engrg., MIT, 77 Massachusetts Avenue, Cambridge, MA 02139, PH 617-258-9468, lsh@mit.edu

[2] Professor, Dept. of Civil and Envir. Engrg., UIUC, 3129 Newmark Lab, 205 n Mathews, Urbana, IL 61801

[3] Assistant Professor, Dept. of Bldg., Schl. of Degn. and Envir., NUS, 4 Architecture Drive SDE 1: 5-33, Singapore 117566; PH 65-6874-6430; bdgmp@nus.edu.sg

INTRODUCTION

Design and 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 negative side effect, one that may generate other unintended problems.

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

However, the traditional contingency buffering approach may not handle these issues because the positioning of the buffer at the end of an activity may just provide for time to recover from schedule disruption without any prevention mechanism.

In response to these problems, this paper briefly examines how feedback processes are generated in the construction process and provides a proactive mechanism to reduce the impact of iterative cycles from a view of lean construction, which aims to remove the non-value added tasks during actual execution.

ITERATIVE CYCLES CAUSED BY ERRORS AND CHANGES

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, or workforce), adopting overtime for the workforce, or changing the construction method. Any or all of these control actions can be taken in order to reduce this gap of work scope. These solutions, however, may generate unintended negative side effects on the project's performance, such as a decrease in productivity or quality. For example, although the adoption of overtime may increase the amount of work being performed, the extended work hours may increase the workforce's fatigue. Ultimately, accumulated fatigue could worsen productivity and the quality of work thereby slowing the project progress down as more errors and changes are introduced. Therefore, these negative side effects which were not intended make the project uncertain and complex, as seen in Figure 1.

DERIVATIVE ACTIVITY

The impact caused by errors and changes on the work scope varies depending on the nature of the activity and its relationship to other activities [Park and Peña-Mora, 2001]. In addition, another important point is the discovery time of errors and changes. If errors and

changes in the predecessor activity are identified early and are immediately implemented, then subsequent activities are not significantly delayed.
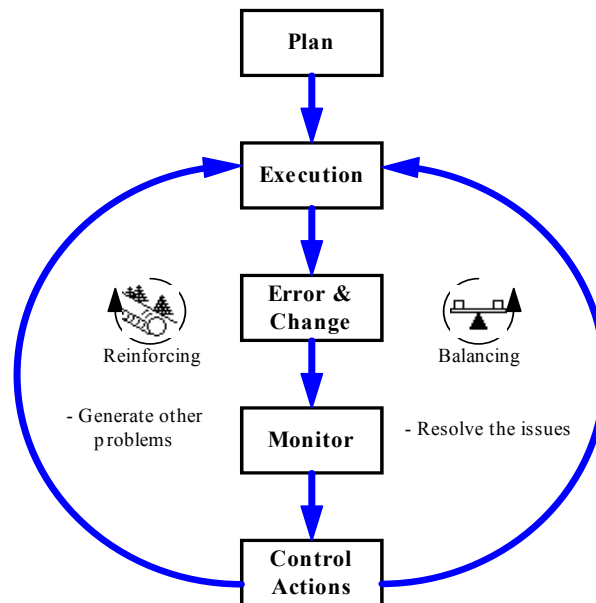

Figure 1: Feedback Processes throughout Actual Execution

However, when the predecessor's errors and changes are discovered late in the project, such as in the successor activities, there is more of a serious impact on the project's rate of completion. Errors and changes that are not immediately discovered and approved after their generation are denoted as *hidden errors* and *latent changes*, respectively. These late discoveries can create work overflow and lead to 'last minute syndrome'. For example, as illustrated in Figure 2, when errors and changes occur in the predecessor activity but are not identified until the successor activity is performed, two activities must be coordinated to effectively respond to these errors and changes. In this case, the successor activity could be delayed before the work in the predecessor activity is completed.

In another instance, when errors and changes are discovered after the predecessor activity is complete, workers and equipment from completed activities may have to be called back to the site, in particular, if the only way to make adjustments is to go back and re-implement the predecessor activity. In addition, if the predecessor and the successor activities are performed by different subcontractors, this situation may require new or additional contracts to accommodate the discovered errors or changes, and it may generate a contractual dispute over the liabilities of these errors or changes. In Figure 2, we denote this situation as *derivative activity*, and it should be avoided for the successful project completion.

One way to avoid this *derivative activity* is to have the mechanism that discovers errors and changes as soon as they are generated. For this, next section introduces the new simulation-based buffering approach which is different from traditional contingency buffering.
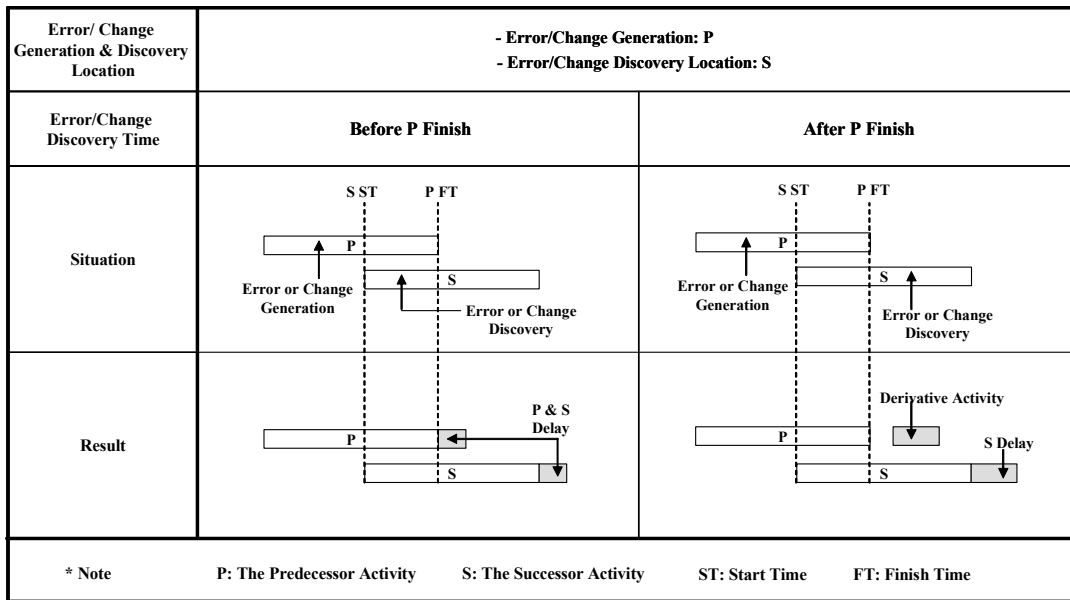
Error/ Change Generation & Discovery Location	- Error/Change Generation: P - Error/Change Discovery Location: S	
Error/Change Discovery Time	**Before P Finish**	**After P Finish**
Situation	S ST P FT P Error or Change Generation S Error or Change Discovery	S ST P FT P Error or Change Generation S Error or Change Discovery
Result	P P & S Delay S	Derivative Activity P S Delay S
* Note	P: The Predecessor Activity S: The Successor Activity ST: Start Time FT: Finish Time	

Figure 2: Impact of Late Discovery

RELIABILITY AND STABILITY BUFFERING

To provide a mechanism to reduce sensitivity to errors and changes, *reliability and stability buffering* is proposed in this paper, which extends the reliability buffering concept [Park and Peña-Mora, 2001].

Reliability buffering aims to provide a mechanism to help a project absorb the impact of errors and uses an approach that is different than traditional contingency buffering. The traditional contingency buffer in construction planning allocates certain time at the end of an activity to absorb certain delay on the execution of the activity. However, the positioning of the buffer at the end of an activity does not provide for any prevention of possible schedule disruption, rather it just gives time to project personnel to recover from a disruption. In other words, it is a reactive mechanism against delay instead of proactive mechanism against delays. In contrast to the traditional contingency buffer, reliability buffering is positioned at the beginning of an activity to absorb delays from predecessor activities and to plan for the succeeding activities [Park and Peña-Mora, 2001]. This different positioning could handle ill-defined tasks by introducing a pre-checking process that can capture and correct predecessors' hidden errors before tasks are being performed. It reinforces the notion that one has to plan for eventualities and analyze decisions in the context of the whole project and not only from perspective of the activities being impacted.

On the other hand, this different positing provided by reliability buffering has another advantage. Usually, the traditional contingency buffer in construction planning tends to be used as part of an activity without clear distinction from the original duration [Horman and Kenley 1998]. 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 Parkinson's law [Parkinson, 1957] that the work expands to fill the time available for its completion.

In contrast to traditional contingency buffering, reliability buffering first attempts to take off contingency buffers from all individual activities if they have their 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. In this 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 35 days and these durations do not include the contingency buffer, 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 [Oliva, 1996].

In this research, the reliability buffering concept is extended to *reliability and stability buffering*, to incorporate both errors and changes, and splitting it is proposed so as to more effectively deal with uncertainties in concurrent design and construction.

RELIABILITY AND STABILITY BUFFER SPLIT

Though *reliability and stability buffering* can protect the schedule from unpredictable events, it may be ineffective in certain cases if they are implemented as originally developed, e.g., at dealing with the uncertainties of predecessor activities, in particular, in the case of concurrent design and construction. For instance, suppose errors and changes are more generated at Part B than Part A of Activity A as illustrated in Figure 3. 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 Part A not Part 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* (dual buffer) is introduced and is only activated in the case that two related activities considerably overlap on duration. If activities have a serial scheduling network such as finish-to-start relationship with positive lag, the *reliability and stability buffer* without split (single buffer) can deal with the uncertainty of the predecessor activity because it can cover all the predecessor activity with its finalized information. Otherwise, the dual buffer may be activated to handle excessive concurrent development. On the other hand, in terms of location, the second buffer in the dual buffer occurs at the time when the predecessor activity finishes. As the actual process is performed, the location of the second buffer may vary depending on the predecessor activity finish time.

**Hidden Error or Latent Change Generation
in the Predecessor Activity**

• **Note**
CB: Contingency Buffer
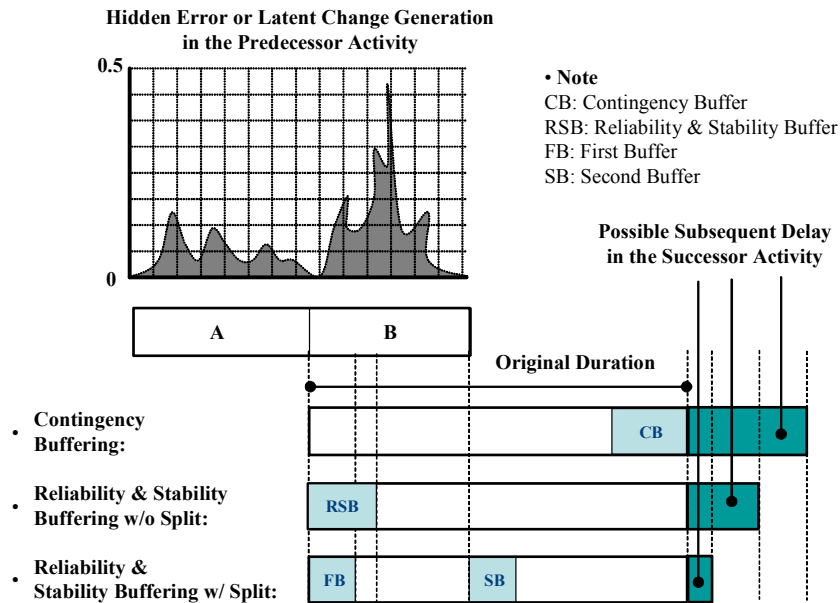RSB: Reliability & Stability Buffer
FB: First Buffer
SB: Second Buffer

Figure 3: Reliability and Stability Buffer Split

In a real construction process, if a single buffer is applied, its meaning is very similar to the pre-checking process to make sure that inherited performance from predecessor activities matches with planned performance. In the case of the dual buffer, the first buffer plays the same role as the one in the single buffer case to check if the performance of the first-half (Part A) of the predecessor activity is the same as planned. Similarly, the second buffer role is to check if the second-half (Part B) of the predecessor's performance corresponds with the planned performance as well.

THE EFFECT OF RELIABILITY AND STABILITY BUFFERING

The *reliability and stability buffer* can prevent the possible ripple effect caused by error and change iteration during the design and construction process of a project. At the *derivative activity* section, we discussed error and change impact on a project by discovery time and location. If error or change in the predecessor activity is discovered at the successor activity, in particular, after the predecessor activity finishes, it can have a strong negative impact on the successor activity as well as the predecessor activity, as seen in Figure 2. *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.

For example, suppose Activity A and Activity B are developed concurrently, having start-to-start relationship with lag 10, as illustrated in Figure 4. If an error is uncovered at the point of X, and extra work is adopted by a managerial decision, it may give a 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.
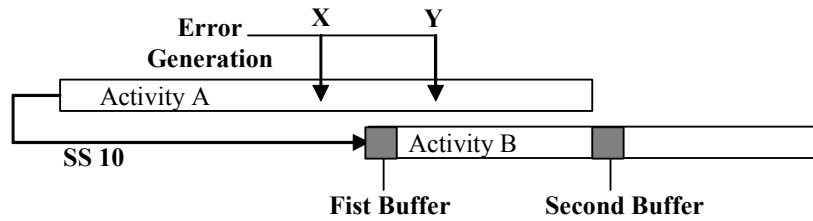

Figure 4. The Effect of Reliability and Stability Buffering Split

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 4, if 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, it may request additional work to the predecessor activity. However, the adoption of the successor activity's requests may generate additional work for already completed tasks in the successor activity. In this case, the *reliability and stability buffer* help prevent this situation, providing pre-checking time to discover *hidden errors* of the predecessor activity before actual execution of Activity B. Even if an error is generated at the point of Y and becomes a *hidden error*, it can be revealed by the second buffer, and consequently, the ripple effect on the successor activity can be reduced.

EFFECTIVENESS OF RELIABILITY AND STABILITY BUFFER

In order to examine the effectiveness of *reliability and stability buffer*, the dynamic construction project model, which has been developed for Dynamic Planning and control Methodology (DPM, Park and Peña-Mora, 2001), is adopted and simulated with the diverse cases. These cases are 'contingency buffer', 'single buffer', and 'dual buffer' case with the model variable settings detailed in Table 1.

Both Activity A and Activity B have 60-day duration having start-to-start precedence relationship with lag 30. In buffering applied cases, 20% of the activity duration is used for schedule contingency and 50% of that contingency is used as the *reliability and stability buffer*. In addition, the dual buffer takes each 50% of total reliability and stability buffer size. Therefore, if the single buffer is considered, buffer size is 6 days (60*0.2*0.5) and the dual buffer case, buffer size is 3 days (60*0.2*0.5*0.5).

As seen at the bottom of Table 1, cases applying the *reliability and stability buffer* show better ability to reduce durations than 'contingency buffer' 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 as discussed earlier.

On the other hand, all simulated durations of the three cases are longer than the duration produced by CPM, 90 days. This is because CPM does not consider error and change

iterative cycles. Therefore, the duration of CPM may be difficult to achieve accurately, if the actual conditions of the activities were materialized [Park and Peña-Mora, 2001].

Table 1. 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			
Single Buffer	Buffer Size (Days)	NA	6
Dual Buffer	First Buffer Size (Days)	NA	3
	Second Buffer Size (Days)	NA	3
Each Case Simulation Result		**Activity Duration (Days)**	
Contingency Buffer Case		80	108
Single Buffer Case		71	95
Dual Buffer Case		71	86
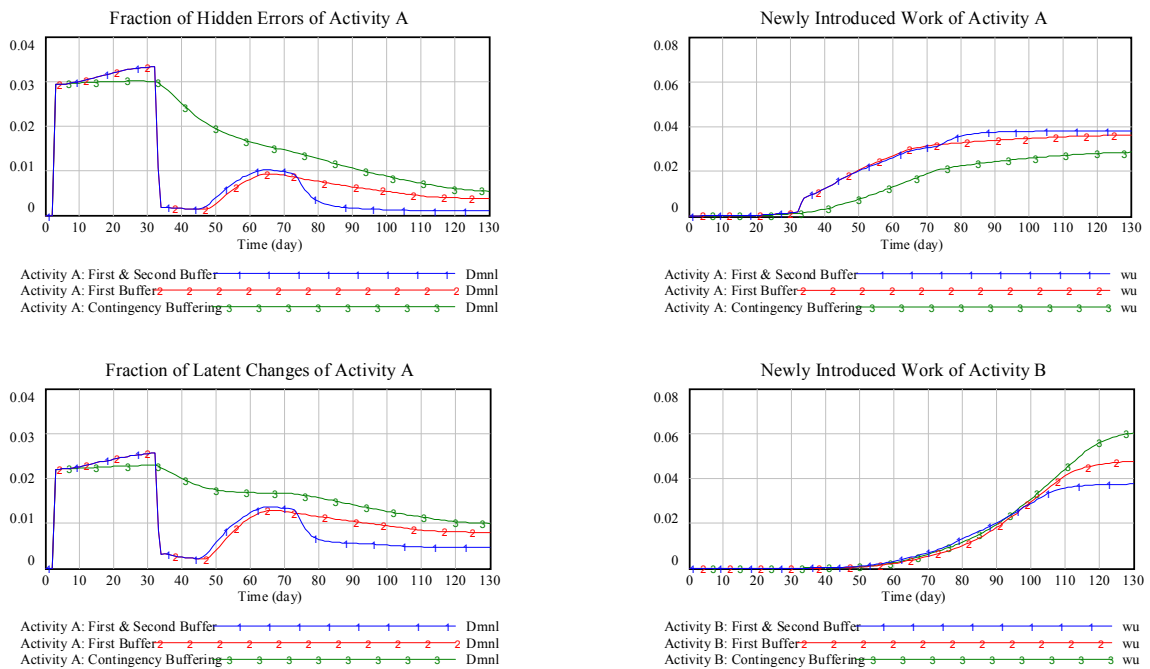
Figure 5. Simulation Result of No Buffering and Buffering Adaptation

The best result in this simulation setting is achieved from the 'dual buffer' case because the second buffer can deal with error and change impact of the later part of Activity A in this concurrent development, where the first buffer can't. The simultaneous consideration of the first and the second buffer reduces duration of Activity B as much as 22% from the contingency buffer case (108 → 86 days) and 9% from the single buffer case (95 → 86 days). In detail, *hidden errors* and *latent changes* of Activity A are reduced due to the discovery of them at the first and second buffer period, as seen in Figure 5. These reduced fractions of *hidden errors* and *latent changes* make it possible to avoid their harmful ripple effects on performance In addition, though the amount of newly introduced work for Activity A in the buffered case is much greater than the original case, early adoption enabled by *reliability and stability buffering* makes the amount of newly introduced work for Activity B less than the original case at the later stage, as seen in Figure 5. This effect contributes to avoiding the 'last minute syndrome'.

CONCLUSION

Today's wide adoption of concurrent design and construction makes projects uncertain and complex mainly due to iterative cycles caused by errors and changes. Moreover, if these errors and changes are not discovered and adopted immediately, the impact could be more hazardous than anticipated. As a mechanism to reduce these impacts of iterative, in particular, in concurrent design and construction, a simulation-based *reliability and stability buffering* is proposed with different positioning and size from the traditional contingency buffer. In addition, simulation results show that this new buffering approach can protect the planned schedule and performance well by reducing the sensitivity to errors and changes.

REFERENCE

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

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

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.

Horman, M. & Kenley R., (1998a) "Process Dynamics: Identifying a Strategy for the Deployment of Buffers in Building Projects", *International Journal of Logistics: Research & Applications,* 1, 3, pp. 221-237.

Lee, S., Peña-Mora, F, and Park, M. (2003), "Quality and Change Management Framework for Concurrent Design and Construction' *ASCE Construction Research Congress,* March 19-21, 2003, Hawaii, ASCE, Reston, Virginia.

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.

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.

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.

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.

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.

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.

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.

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.

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