Dynamic Planning and Control Methodology
for Large-Scale Concurrent Construction Projects

by

Moonseo Park

B.S., Architectural Engineering, Seoul National University, 1989
M.A., Landscape Architecture, Seoul National University, 1992
M.S., Civil and Environmental Engineering, MIT, 1999

Submitted to the Department of Civil and Environmental Engineering
in Partial Fulfillment of the Requirements for the Degree of
PhD in the Field of Construction Engineering and Management

at the

Massachusetts Institute of Technology

June 2001

© 2001 Massachusetts Institute of Technology
All rights reserved

Signature of Author

Department of Civil and Environmental Engineering
May 18, 2001

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 Planning and Control Methodology for Large-Scale Concurrent Construction Projects

by

Moonseo Park

Submitted to the Department of Civil and Environmental Engineering in Partial Fulfillment of the Requirements for the Degree of PhD in the Field of Construction Engineering and Management

ABSTRACT

Concurrent construction has been widely used for modern construction projects, as a method to shorten time-to-market. Concurrent construction, however, requires a careful and systematic approach to its planning and management, since it also has greater potential to impact the construction process than the traditional more serial method. These industrial trends and challenges in concurrent construction, together with increased understanding of dynamics and complexities of construction, have increased the demand for a more efficient planning and control method. In this context, the simulation-based scheduling method that has the potential to more effectively deal with the dynamic state of construction processes has currently emerged as an alternative to the network-based method. However, despite its potential advantages over the network-based method, very few of the existing simulation tools have overcome their practical limitations and have proven their applicability to real construction processes. As an effort to address some of these challenging issues, this thesis presents Dynamic Planning and Control Methodology (DPM) that has been developed to help prepare a more robust construction plan against uncertainties and to provide policy guidelines for the planning and control of a construction project, taking into consideration the context in which the project is being developed. The use of DPM would be especially beneficial for construction projects performed concurrently and involving higher complexity and uncertainties, ensuring that those projects can be delivered in time without driving up costs.

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

This thesis is dedicated to all of my family for their love and caring. In particular, I would like to give special thanks to my mother who has never slept well due to anxiety over her son, and early every morning, has prayed for her son’s health and success at MIT. I cannot repay what I owe to her but just hope this thesis can be her small happiness.

I am really thankful to my mentor, Professor Peña-Mora for his inexhaustible support and advice on my research and thesis. I can never forget that he called me friend and that he believed in my ability, when it was a challenge for me to see it myself.

Many thanks to Professor Moavenzadeh, Professor Lyneis, and Professor Miller. Their kind and sincere advices on my thesis were invaluable to me. Without their continual guidance, this thesis could not have been completed. I would also like to thank Dr. Kwak. His humble but creative advices have been helpful in system dynamic modeling work. Also, I have to thank Professor Hyunsoo Lee at SNU, Professor Manhyoung Lee at CNU, Professor Kichan Kim at CUK and Professor Sebyung Oh at YNU. They not only helped with invaluable advices, but also gave me pleasant memories.

In addition, I would like to acknowledge the contribution to this thesis by David Ford, Assistant Professor at Texas A&M University, Joe Peck, Corporate Planning and Scheduling Manager at Modern Continental Co., Philip Helmes, Vice President at InteCap Inc., and Sanghyun Lee and Margaret Fulenwider, Graduate Students at MIT, and the support for the thesis research received from InteCap Inc., the National Science Foundation CAREER Award and the White House PECASE Award CMS-9875557.

Finally, I would like to thank my wife, Heesook, son, Chanhyuk, and daughter, Phoebe. And, I want to apologize to Chanhyuk who continuously nagged me not to go to school after dinner. I often forget but can now appreciate that my studies at MIT would be impossible without them. When I returned home late, they were asleep peacefully, hand in hand. Seeing them was enough to make me realize how happy I was.

Thank God.
TABLE of CONTENTS

EXECUTIVE SUMMARY .......................................................................................................................... 9

CHAPTER 1: RESEARCH BACKGROUND .................................................................................................. 12
  1.1 OBJECTIVES OF THE RESEARCH ......................................................................................... 15
  1.2 BENEFITS OF THE RESEARCH ......................................................................................... 15

CHAPTER 2: DYNAMICS IN CONCURRENT CONSTRUCTION .......................................................... 17
  2.1 CONCURRENT CONSTRUCTION ....................................................................................... 18
  2.2 DESIGN-DRIVEN FEEDBACKS ......................................................................................... 19
  2.3 CONSTRUCTION-DRIVEN FEEDBACKS .............................................................................. 21

CHAPTER 3: DPM AS AN ALTERNATIVE TO NETWORK-BASED TOOLS ...................................... 24
  3.1 SIMULATION APPROACHES ........................................................................................... 25
  3.2 DYNAMIC PLANNING AND CONTROL METHODOLOGY .............................................. 26

CHAPTER 4: COMPONENTS METHODOLOGIES .............................................................................. 30
  4.1 SYSTEM DYNAMICS ........................................................................................................ 31
  4.2 CONCURRENT ENGINEERING ........................................................................................ 32
  4.3 OVERLAPPING FRAMEWORK FOR CONSTRUCTION ....................................................... 34

CHAPTER 5: FUNDAMENTALS OF DPM ............................................................................................ 37
  5.1 USER-DEFINED MODELING APPROACH ........................................................................ 38
  5.2 CONSIDERATION OF FEEDBACKS .................................................................................. 39
  5.3 CAPTURING CONSTRUCTION DYNAMICS ...................................................................... 41
    5.3.1 Change as Iteration Trigger ...................................................................................... 41
    5.3.2 Dependencies ............................................................................................................ 46
      Internal Dependency ....................................................................................................... 47
  5.4 REDUCING SENSITIVITY TO CHANGES ............................................................................ 49
    5.4.1 Traditional Buffering Practice ................................................................................... 49
      Lack of Characteristics .................................................................................................... 50
      Losses on a Merging Point ............................................................................................. 50
      Inefficient Sizing ........................................................................................................... 51
    5.4.2 Reliability Buffer ....................................................................................................... 52
      Buffering Logistics ......................................................................................................... 52
      Buffer Sizing ................................................................................................................ 54
      Dynamic Buffering ......................................................................................................... 55
  5.5 SMART SYSTEM .................................................................................................................... 61

CHAPTER 6: IMPLEMENTATION ........................................................................................................... 63
  6.1 SYSTEM ARCHITECTURE .................................................................................................... 64
    6.1.1 Integrating Methodologies ........................................................................................ 64
      Strategic Planning and DSM ......................................................................................... 64
      CPM & PDM .................................................................................................................. 68
      PERT .................................................................................................................................. 68
      GERT, Q-GERT and SLAM ............................................................................................ 68
      Concurrent Engineering, Critical Chain and Overlapping Framework ....................... 69
      System Dynamics Project Management Models ........................................................... 69
# LIST of FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1-A</td>
<td>Design-Driven Feedbacks</td>
<td>20</td>
</tr>
<tr>
<td>FIGURE 1-B</td>
<td>Design-Driven Feedbacks</td>
<td>22</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>Comparison of Simulation Techniques</td>
<td>29</td>
</tr>
<tr>
<td>FIGURE 3-A</td>
<td>Causal Loop Diagram Notation</td>
<td>31</td>
</tr>
<tr>
<td>FIGURE 3-B</td>
<td>Stock and Flow Structure</td>
<td>32</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>Overlapping Framework</td>
<td>33</td>
</tr>
<tr>
<td>FIGURE 5</td>
<td>Overlapping Framework for Construction</td>
<td>35</td>
</tr>
<tr>
<td>FIGURE 6</td>
<td>User-Defined Modeling Approach</td>
<td>38</td>
</tr>
<tr>
<td>FIGURE 7</td>
<td>Feedback Processes in Project Execution</td>
<td>40</td>
</tr>
<tr>
<td>FIGURE 8</td>
<td>Changes as Iteration Trigger</td>
<td>42</td>
</tr>
<tr>
<td>FIGURE 9</td>
<td>Behaviors of Change and Rework</td>
<td>45</td>
</tr>
<tr>
<td>FIGURE 10</td>
<td>Examples of External Dependency</td>
<td>46</td>
</tr>
<tr>
<td>FIGURE 11</td>
<td>Examples of Internal Dependency</td>
<td>47</td>
</tr>
<tr>
<td>FIGURE 12</td>
<td>Contingency Buffer on a Merging Point</td>
<td>51</td>
</tr>
<tr>
<td>FIGURE 13</td>
<td>Reliability Buffer on a Merging Point</td>
<td>54</td>
</tr>
<tr>
<td>FIGURE 14-A</td>
<td>Dynamic Buffering I</td>
<td>56</td>
</tr>
<tr>
<td>FIGURE 14-B</td>
<td>Dynamic Buffering II</td>
<td>57</td>
</tr>
<tr>
<td>FIGURE 15</td>
<td>Reliability Buffering Steps</td>
<td>60</td>
</tr>
<tr>
<td>FIGURE 16</td>
<td>Smart System</td>
<td>62</td>
</tr>
<tr>
<td>FIGURE 17</td>
<td>Integrating Methodologies</td>
<td>65</td>
</tr>
<tr>
<td>FIGURE 18-A</td>
<td>Smart Cell for Activity</td>
<td>66</td>
</tr>
<tr>
<td>FIGURE 18-B</td>
<td>Smart Cell for Relationship</td>
<td>67</td>
</tr>
<tr>
<td>FIGURE 19</td>
<td>System Architecture of DPM</td>
<td>72</td>
</tr>
<tr>
<td>FIGURE 20-A</td>
<td>Collaboration Scheme</td>
<td>75</td>
</tr>
<tr>
<td>FIGURE 20-B</td>
<td>Scheme of Web-Based Collaboration</td>
<td>76</td>
</tr>
<tr>
<td>FIGURE 20-C</td>
<td>Scheme of Data Exchange in DPM Systems</td>
<td>77</td>
</tr>
<tr>
<td>FIGURE 20-D</td>
<td>Data Exchange with Primavera</td>
<td>78</td>
</tr>
<tr>
<td>FIGURE 20-E</td>
<td>DPM Working on the Web</td>
<td>79</td>
</tr>
<tr>
<td>FIGURE 21</td>
<td>Model Boundary</td>
<td>83</td>
</tr>
<tr>
<td>FIGURE 22-A</td>
<td>Change Option-Further Delay Loop</td>
<td>85</td>
</tr>
<tr>
<td>FIGURE 22-B</td>
<td>Quality Management Thoroughness-Further Delay Loop</td>
<td>86</td>
</tr>
<tr>
<td>FIGURE 22-C</td>
<td>Downstream Reprocess Iteration-Further Delay Loop</td>
<td>87</td>
</tr>
<tr>
<td>FIGURE 22-D</td>
<td>Feedback Processes in Construction Activities</td>
<td>89</td>
</tr>
<tr>
<td>FIGURE 23</td>
<td>Generic Construction Process Model Structure</td>
<td>92</td>
</tr>
<tr>
<td>FIGURE 24</td>
<td>Fraction of RFI</td>
<td>94</td>
</tr>
<tr>
<td>FIGURE 25</td>
<td>Hidden Change Co-Flow</td>
<td>99</td>
</tr>
<tr>
<td>FIGURE 26</td>
<td>Hidden Change Iteration Paths</td>
<td>101</td>
</tr>
<tr>
<td>FIGURE 27</td>
<td>Schema of Dynamic Project Model</td>
<td>104</td>
</tr>
<tr>
<td>FIGURE 28</td>
<td>Cost-Benefits Tradeoff of Fast-Tracking</td>
<td>112</td>
</tr>
<tr>
<td>FIGURE 29-A</td>
<td>Workforce (1) during Construction</td>
<td>114</td>
</tr>
<tr>
<td>FIGURE 29-B</td>
<td>Workforce (2) during Construction</td>
<td>115</td>
</tr>
<tr>
<td>FIGURE 30</td>
<td>Workforce of Excavation Work</td>
<td>117</td>
</tr>
<tr>
<td>FIGURE 31</td>
<td>Productivity Determinants of Excavation Work</td>
<td>118</td>
</tr>
<tr>
<td>FIGURE 32</td>
<td>Effect of Labor Policies</td>
<td>122</td>
</tr>
<tr>
<td>FIGURE 33</td>
<td>Effect of Changes in Time Variables</td>
<td>123</td>
</tr>
<tr>
<td>FIGURE 34</td>
<td>Cost-Benefits Tradeoffs of Alternative Policies</td>
<td>126</td>
</tr>
<tr>
<td>FIGURE 35</td>
<td>Impact of Managerial Changes</td>
<td>129</td>
</tr>
<tr>
<td>FIGURE 36</td>
<td>Basis of Unintended Change Impact</td>
<td>130</td>
</tr>
</tbody>
</table>
FIGURE 37-A: IMPACT OF UNINTENDED CHANGES (CASE I) ................................................................. 133
FIGURE 37-B: IMPACT OF UNINTENDED CHANGES (CASE II) .......................................................... 134
FIGURE 37-C: IMPACT OF UNINTENDED CHANGES (CASE III) .......................................................... 135
FIGURE 38: SIMULATION OF RELIABILITY BUFFERING .................................................................. 138
FIGURE 39: MULTIVARIATE SENSITIVITY SIMULATIONS OF DOWNSTREAM PROGRESS .............. 141
FIGURE 40: PROJECT LOCATION AND EXISTING BRIDGE ............................................................. 145
FIGURE 41: PRIMAVERA-GENERATED ACTIVITY DURATIONS ..................................................... 148
FIGURE 42: DPM-GENERATED ACTIVITY PERFORMANCE ............................................................. 149
FIGURE 43-A: FLEXIBLE LABOR POLICIES VS PROJECT PROGRESS ........................................... 152
FIGURE 43-B: FLEXIBLE LABOR POLICIES VS WORK HOURS ..................................................... 153
FIGURE 43-C: FLEXIBLE LABOR POLICIES VS CUMULATIVE WORK HOURS ............................... 154
FIGURE 44-A: OVERTIME LABOR POLICIES VS PROJECT PROGRESS .......................................... 155
FIGURE 44-B: OVERTIME LABOR POLICIES VS WORK HOURS ..................................................... 156
FIGURE 45: LABOR POLICIES VS PROJECT PERFORMANCE .......................................................... 157
FIGURE 46-A: RELIABILITY BUFFERING VS PROJECT PROGRESS ............................................... 160
FIGURE 46-B: RELIABILITY BUFFERING VS WORK HOURS .......................................................... 161
FIGURE 47-A: HIRING TIME VS PROJECT PROGRESS ................................................................. 163
FIGURE 47-B: HIRING TIME VS WORK HOURS .............................................................................. 164
FIGURE 48-A: RFI REPLY TIME VS PROJECT PROGRESS ............................................................ 165
FIGURE 48-B: RFI REPLY TIME VS WORK HOURS ....................................................................... 166
FIGURE 49: TIME DELAYS VS PROJECT PERFORMANCE ............................................................. 167
FIGURE 50-A: RECOMMENDED POLICIES VS PROJECT PROGRESS ........................................ 169
FIGURE 50-B: RECOMMENDED POLICIES VS WORK HOURS ..................................................... 170
FIGURE 51: DISTRIBUTION OF PROJECT COMPLETION TIMES VS. VARIATIONS IN WORK QUALITY ....................................................................................................................... 173
FIGURE 52: DISTRIBUTION OF PROJECT COMPLETION TIMES VS. VARIATIONS IN INPUT DURATIONS .......................................................................................................................... 174
FIGURE 53: FURTHER DEVELOPMENT OF DPM ............................................................................ 180
LIST of TABLES

TABLE 1: COMPARISON OF DPM WITH NETWORK-BASED METHODS ................................................................. 28
TABLE 2: CHARACTERISTICS OF CHANGE AND REWORK .................................................................................. 43
TABLE 3: TRADITIONAL BUFFERING PRACTICE ..................................................................................................... 50
TABLE 4: BUFFER LOCATION AND PRECEDENCE RELATIONSHIP CHANGE PATTERNS .................................... 59
TABLE 5: FUNCTIONS OF DPM ............................................................................................................................ 73
TABLE 6: COMPARISON WITH PREVIOUS PROJECT MANAGEMENT MODELS ....................................................... 106
TABLE 7: MAJOR IMPROVEMENT IN PROJECT MANAGEMENT MODELS ........................................................... 108
TABLE 8: DESCRIPTION OF BUILDING CONSTRUCTION PROJECT ........................................................................ 111
TABLE 9: EFFECT OF DEGREE OF OVERLAPPING ................................................................................................. 120
TABLE 10: EFFECT OF LABOR POLICIES .................................................................................................................. 121
TABLE 11: EFFECT OF CHANGES IN TIME VARIABLES ............................................................................................ 123
TABLE 12: COST-BENEFITS TRADEOFFS OF ALTERNATIVE POLICIES ................................................................. 125
TABLE 13: CHANGE IMPACTS AND ROLE OF RELIABILITY BUFFERING ............................................................. 131
TABLE 14: BASE SCENARIO FOR MODEL SIMULATION ........................................................................................... 137
TABLE 15: SIMULATION SETTINGS FOR SENSITIVITY TEST ..................................................................................... 139
TABLE 16: OPTIMAL BUFFER SIZES IN DIFFERENT CONDITIONS ............................................................................. 142
TABLE 17: INPUT DATA FOR THE TREBLE COVE PROJECT ................................................................................ 146
TABLE 18: EFFECT OF LABOR POLICIES ................................................................................................................ 151
TABLE 19: PROJECT COMPLETIONS ACCORDING TO BUFFERING ........................................................................ 159
TABLE 20: EFFECT OF TIME DELAYS ..................................................................................................................... 162
TABLE 21: SIMULATION OF POLICY RECOMMENDATIONS .................................................................................... 168
TABLE 22: SENSITIVE STUDIES OF DPM POLICIES ............................................................................................... 171
EXECUTIVE SUMMARY

Concurrent construction has been widely used for modern construction projects, as a method to shorten time-to-market. In fact, its time saving feature has placed it as a possible alternative to the traditional more serial method. However, concurrent construction also has greater potential to impact the project development process than the traditional more serial method [Pena-Mora and Park, 2001]. In reality, this popular method often results in unexpected costs and does not necessarily always lead to the expected shorter project duration [Fazio et al., 1988]. Despite having different explanations, researchers on concurrent construction commonly argue that increasing concurrency without a carefully planned strategy can lead to a poor management of concurrent construction and in turn disrupt construction sequences. These industrial trends and challenges in concurrent construction have increased the demand for a more efficient planning and control method together with increased understanding of dynamics in construction.

In fact, the traditional network-based scheduling methods lack a mechanism to efficiently formulate and evaluate construction plans under uncertainties and constraints, which is required to deal with the high degree of complexities involved in today’s construction projects. Problems
encountered in the planning and control of construction projects are inherently dynamic but they are treated statically in the network-based scheduling methods. In addition, learning has rarely accumulated across construction projects. This is, in part, because construction is process-based work that is performed on an unfixed place by a temporary alliance among multiple organizations [Slaughter, 1999]. However, it is also true that the lack of learning in construction is attributed to the lack of learning mechanism in the traditional planning tools. As a result, the traditional planning tools have not been so helpful in dealing with chronic managerial problems in carrying out construction projects. In this context, the simulation-based scheduling method that has the potential to effectively deal with the dynamic state of construction processes has currently emerged as an alternative to the network-based method. However, despite its potential advantages over the network-based method, few of the existing simulation tools have overcome their practical limitations and have proven their applicability to real construction processes.

As an effort to address some of these challenges, this thesis presents a simulation-based dynamic planning and control tool (called Dynamic Planning and Control Methodology, DPM). Chapter 1 describes the research background including motivations, objectives and expected benefits from the use of DPM. In Chapter 2, the dynamics of concurrent construction are analyzed, focusing on feedback processes, concluding that the planning and management of concurrent construction requires a systematic and dynamic approach. Then, Chapter 3 discusses the target functionality of DPM as an alternative to network-based methods, comparing DPM to the previous research efforts in terms of applicability, flexibility, and reality in representation of construction processes. In order to achieve the target functionality, DPM employs many component methodologies, some of which are introduced in Chapter 4. In addition, Chapter 5 presents the fundamental concepts of DPM, which are user-defined modeling approach, consideration of feedbacks, capturing construction dynamics, reducing sensitivity to changes, and smart system. All of the concepts of DPM have been derived from closer observations of construction processes and practices thus far, and they have been elaborated, taking into consideration functional requirements to achieve DPM target goals. These fundamental concepts have been materialized by systematically integrating DPM component methodologies, which is discussed in Chapter 6. Chapter 6 also presents the system architecture of DPM and describes
the system dynamics models in which all of the component methodologies are integrated, focusing on the generic construction process model. Finally, the performance of DPM as a planning and control tool and its applicability in real world settings are examined with three application examples in Chapter 7.

The proposed DPM would help to prepare a more robust construction plan against uncertainties and provide policy guidelines for the management of a construction project. In particular, DPM could help effectively deal with unplanned and/or indirect events that might occur during construction, by taking into consideration the context in which a construction project is being developed. This would be especially beneficial for construction projects performed concurrently and involving higher complexities and uncertainties. In addition, DPM could also help utilize learning from one project for others, by allowing simulation model structures to be tuned up based on information obtained from the actual project performance. This makes it possible to embed one's knowledge and learning from a project into the planning and control system.

The development of DPM would also contribute to increasing the applicability of the simulation approach in project planning and control. In fact, simulation capability has been seen as an opposite concept to applicability. Partly due to this recognition, the previous research efforts to increase the applicability of the simulation approach have mainly focused on the development of user-friendly graphic representations of simulation components. In contrast, DPM attempts to increase applicability, while keeping required reality in representation by introducing the user-defined modeling approach, in which simulation models are pre-structured and users can define the model settings. In addition, by incorporating the fundamental concepts and principles of network-based tools into the system dynamics models, DPM has the functionality of the traditional planning tools as well as having simulation capability.

Consequently, the use of DPM would help ensure that construction projects could be delivered in time without driving up costs by enhancing the planning and control capabilities.
CHAPTER 1

RESEARCH BACKGROUND

Shortening time-to-market has been one of the most critical factors to the success of businesses in many industries. As a result, companies have sought a method that can ensure a faster product development, most commonly focusing on product cycle time reduction through concurrent development. The construction industry is not an exception. The increasing preference of project owners and managers to fast-track construction proves the popularity of concurrent development in construction. In addition, many success stories of fast-tracking have demonstrated that the popularity of this delivery method is warranted [Huovila et al., 1994; Williams, 1995]. However, concurrent construction also has greater potential to impact the project development process than the traditional more serial method [Pena-Mora and Park, 2001]. In reality, this popular method often results in unexpected costs and does not necessarily always lead to the expected shorter project duration [Fazio et al., 1988].

Many explanations on these potential risks are suggested in the literature. Russell et al. [1991] attributed them to the increased level of uncertainty during construction. Laufer and Cohenca [1990] and Tigh [1991] argued that increased non-value adding iterations caused by incomplete design contributed to unexpected costs and delays. In addition, Pena-Mora and Park [2001] identified more fundamental problems such as the negative effects of feedback processes involved in fast-track construction projects. Despite having different explanations, researches on
concurrent construction commonly argue that increasing concurrency without a carefully planned strategy can lead to a poor management of concurrent construction and in turn disrupt construction sequences.

In addition, these industrial trends and challenges in concurrent construction, together with increased understanding on dynamics and complexities of construction, increased the demand for a more efficient planning and control method, which resulted in research efforts to supplement the traditional CPM (Critical Path Method, DuPont Inc and UNIVAC Division of Remington Rand, 1958). In fact, CPM has been most widely used in the planning and control of construction projects. Most projects today, however, have demonstrated that its usefulness is warranted only when construction is not heavily constrained by either time or resources under a quite dynamic environment [Hegazy, 1999]. To meet this challenge, a significant amount of research efforts have been made to supplement the CPM, mostly through adding probabilistic capabilities [Martinez et al., 1997] and more expressive power. PERT (Program Evaluation and Review Technique; Booz-Allen Hamilton and Lockheed Co., 1958) incorporates probabilities into the duration of activities. PDM (Precedence Diagramming Method; J. David, 1964) diversifies precedence relationships between activities. In addition, GERT (Graphical Evaluation and Review Technique; A. Pritsker, 1966) models ‘what-if’ conditions by incorporating probabilistic branching and loop structures into scheduling.

All of these researches have contributed to enhancing planning and control capabilities to some extents. However, despite the increased capabilities and advanced commercial software packages, the network-based scheduling methods still lack the mechanism to efficiently formulate and evaluate construction plans under uncertainties and constraints, which are required to deal with a high degree of complexities involved in today’s construction projects. Since the network-based methods assume that the attributes of activities such as duration and production rate are known at the beginning of construction and do not change during construction, they cannot represent actual construction processes realistically, which results in frequent updates to reflect the actual performance into scheduling [Martinez and Ioannou, 1997]. Regarding this, many researchers [Halpin, 1973; Paulson, 1983; Bernold, 1989; Martinez, 1996] argue
that problems the Network-based scheduling methods have can be overcome by adopting the simulation approach, which can describe and capture the dynamic state of construction, and provide an analytic tool to evaluate construction plans and find possible problems with a diagnostic capability. Also, their research results including CYCLONE (CYCLic Operations NEtwork; Halpin, 1977), INSIGHT (Interactive Simulation using GrapHics Techniques; Paulson, 1983), and STROBOSCOPE (State and Resource Based Simulation; Martinez, 1996) have demonstrated that the simulation approach can be more effective in dealing with the dynamic state of construction processes than the network-based methods and that its ability to simulate construction plans prior to physical execution can substantially enhance the effectiveness of planning [Martinez and Ioannou, 1997].

Due to its advantageous features, the simulation-based scheduling method has currently emerged as an alternative to the network-based method. However, despite its potential advantages over the network-based method, very few of the existing simulation tools have overcome their practical limitations and have proven their applicability to real construction processes. Their application is still limited to a specific construction process due to the lack of flexibility in modeling and only those who have a lot of modeling experience and knowledge can use them. In addition, excluding human factors from modeling makes simulation results less realistic since many dynamic feedbacks inherent within the construction processes are closely related to human factors e.g. the effect of workers’ fatigue and schedule pressure on productivity.

All of these things necessitate the development of a more flexible and applicable simulation-based tool for the planning and control of construction projects. As an effort to meet these industrial needs, this thesis presents a simulation-based dynamic planning and control tool (called Dynamic Planning and Control Methodology, DPM) that has been developed by incorporating innovative buffering contents and concurrent engineering principles into system dynamics models as well as schedule networking concepts of CPM, PDM, PERT, GERT, and SLAM (Simulation Language for Alternative Modeling, Pritsker, 1994).
1.1 OBJECTIVES OF THE RESEARCH

The research aimed to develop Dynamic Planning and Control Methodology (DPM) that can rigorously deal with indirect and/or unanticipated events that might occur during the construction project execution, taking into consideration the context, in which the project is being developed. In particular, DPM’s simulation-based dynamic approach to planning and control can help enhance the construction performance under highly complex and uncertain environment.

1.2 BENEFITS OF THE RESEARCH

The proposed DPM would help prepare a more robust construction plan against uncertainties and provide policy guidelines for the planning and control of construction projects. A key asset of DPM is that before controlling an activity, the methodology reduces the sensitivity to variations the activity may experience. This feature, together with the ability to formulate and evaluate construction plans ahead of time, helps dampen the effect of hard-to-control variations, while keeping control efforts minimized.

It would also contribute to increasing the applicability of the simulation-based planning and control to construction projects by providing a more flexible and applicable simulation methodology. In fact, the need for integrating two different planning and control approaches (network-based approach and simulation-based approach) has been addressed in the belief that the integration of two approaches could yield particular value, since they offer very different perspectives on project management [Rodrigues and Bowers, 1996]. As will be detailed in Chapter 6.2, DPM rigorously integrates two different approaches in project planning and control by introducing the user-defined modeling approach and the change cycle, and incorporating the fundamental concepts and principles of network-based tools into the DPM models. This could
increase the applicability of the simulation-based approach in construction planning and control, together with the web-based collaboration scheme, based on which the DPM system is structured.

All the features of DPM would be especially beneficial for construction projects performed concurrently and involving higher complexity and uncertainties. Consequently, the use of DPM would help ensure that those projects could be delivered in time without driving up costs by enhancing planning and control capabilities.
CHAPTER 2

DYNAMICS IN CONCURRENT CONSTRUCTION

Construction is inherently dynamic and involves multiple feedback processes, which produce self-correcting or self-reinforcing side effects of decisions [Sterman, 1992]. These feedback processes can become more dynamic and complex under time and resource constraints, which is normally the situation in concurrent construction. For this reason, concurrent construction usually involves more diversified and dynamic feedback processes than does sequential construction.

This chapter identifies feedback processes involved in concurrent construction, followed by a brief introduction of concurrent construction. In addition, this chapter also demonstrates how those feedback processes can impact the design and construction process. Findings in this chapter explain why the planning and management of concurrent construction requires a systematic and dynamic approach and provide a logical background for DPM to focus on feedback processes involved in construction.
2.1 CONCURRENT CONSTRUCTION

During the 1970s, technical complexity of projects, increased government regulations, spiraling inflation and political pressures have all contributed to the increased cost of construction, which resulted in a search for new and imaginative procedures to ensure faster and more economical project completions [Fazio et al. 1988]. In an effort to meet these challenges, phased construction and fast-tracking management techniques have been developed as part of the Professional Construction Management approach [Barrie and Paulson, 1984]. Since then, these accelerated project delivery methods have received considerable attention under the competitive business environment.

Both phased construction and fast-tracking approaches are part of concurrent construction but concurrent construction has a broader meaning. In the literature, fast-tracking has been most often understood simply as overlapping between design and construction. For instance, Hendrickson and Au [1989] argue that in fast-tracking, initial construction activities are begun even before the facility design is finalized. And, Project Management Institute [1987] defines fast-tracking as the starting or implementation of a project by overlapping activities, commonly entailing the overlapping of design and construction activities. As a result, an emphasis has been usually given only to the overlapping between design and construction. However, construction activities themselves are also often overlapped when they are performed under time constraints. This also has a significant influence on project performance, especially when the overlapped activities have a high criticality. For this reason, concurrent construction refers to overlapping among construction processes themselves as well as overlapping between design and construction processes.

Meanwhile, despite its promise of speed, concurrent construction also has greater potential to impact the project development process than the traditional more sequential method. In reality, this popular method often results in unexpected costs and does not necessarily lead to the expected shorter project duration [Fazio et al., 1988]. In the literature, these potential risks are usually attributed to the increased level of uncertainty [Russell et al, 1991]. As a result,
research efforts on concurrent construction have mainly focused on uncertainty reduction. However, in dealing with uncertainty, the previous studies did not explicitly address the potential effects of feedback processes involved in concurrent construction. Closer observations of the design and construction process, as will be described in the following chapter, indicate that the feedback processes make the construction process more dynamic and unstable, possibly creating negative impacts on the project performance. In particular, when a project is concurrently performed without proper planning those feedback processes can cause the disruption of the whole project development process. For these reasons, to effectively plan and control concurrent construction, and minimize its negative impact, the feedback processes involved in concurrent construction need to be identified and the dynamic behavior of construction resulting from those feedback processes needs to be dealt with in a systematic manner.

2.2 DESIGN-DRIVEN FEEDBACKS

Figure 1-a and Figure 1-b represent feedback processes possibly existing in concurrent construction, which are either positive or negative to the project performance. Specifically, the solid arrows in Figure 1-a represent the feedback processes such that overlapping between the design and construction process may or may not be beneficial for the project performance. With proper planning and management, the design and construction overlapping can shorten the project duration and reduce costs, as initially planned. However, it can also delay the schedule and increase costs for various reasons. The design and construction overlapping makes the design work usually proceed with insufficient volume and poor information. Given the uncertainty in the design process, the necessity for assumptions to be made may be increased [Tighe, 1991].

For instance, the structural engineers may be forced to anticipate loads of a facility for the foundation design before the final facility layout and detailed specifications for the facility, such as exterior finishing material, have been determined. This increased level of assumption may lead to frequent design changes. In addition, with concurrent, design
work may begin before the owner's requirements are firmly determined. In such a case, the owner's needs and the corresponding design may be altered more often than on a traditional project. Frequent design changes driven by both the increased assumption level in design and owners' requests on design changes, in turn, can produce more rework or non value-adding changes (hereinafter referred to as rework or changes) in construction. Along with the increase of design changes, the increased possibility of design changes' impact on construction can also increase the potential for construction rework or changes. With the traditional delivery method, design changes created in the design stages do not necessarily affect the construction performance, as construction may not have started. In concurrent construction, however, design changes occurring during the design stages may cause rework or changes in construction when the construction component is already underway. Increased construction changes can delay project duration, which may trigger further overlapping between design and construction.
Consequently, all of the feedback processes mentioned above can contribute to an increase in construction rework or changes, which could lead to schedule delays and cost overruns. Meanwhile, there is another phenomenon found in the concurrent design work that is not necessarily found on the traditional method: everyone in the design process often makes allowances for unknowns, to avoid possible impacts and changes [Tighe, 1991]. For instance, without knowing the required capacity of an air-handling unit (AHU) the mechanical engineers may over-size it to ensure adequate capacity. Likewise, the architects may assign additional space for that equipment and the structural engineer may increase the safety factor for its structural analysis due to the unknown loads that the AHU may impose on the structure. This over-sizing practice increases protection but at the same time may cause a substantial increase in the project costs due to inefficient use of resources.

2.3 CONSTRUCTION-DRIVEN FEEDBACKS

Concurrent construction also involves feedback processes within construction processes, as shown by the solid arrows in Figure 1-b. On a concurrent project, as a result of the design-driven feedback processes discussed in the previous chapter, more rework and changes can occur during construction, which generates more work to do. The construction schedule, however, cannot be simply extended due to time constraints. One possible control action to meet the schedule is to increase work hours, either by hiring more workers or putting them to work overtime. Increased work hours can help facilitate the construction. However, once the self-reinforcing side effects of the action become dominant over self-correcting effects, the control action can result in further delays and a rise in construction costs.

For example, when overtime continues beyond a certain threshold, workers will become fatigued, leading to lower productivity (denoted as R1 in Figure 1-b) and an increase in error rate (denoted as R3 in Figure 1-b), which, in turn, further delays the construction schedule and requires more use of overtime. Another possibility to meet the schedule may lie in running
the project more in parallel by overlapping activities, which forces workers to work on construction components, for which the upstream work may not yet be completed (denoted as R2 in Figure 1-b). As a result, the downstream work does not have a schedule buffer that can absorb the impact of any errors and changes made during upstream work, which makes the downstream work more vulnerable to the upstream errors and changes. This can also lead to an increase in construction changes and further delays, which, in turn, requires more overlapping of the construction processes. Consequently, all of these feedback processes can produce more construction changes by self-reinforcing their vicious loop effects, which results in schedule delays and cost overruns. In contrast to the feedback processes driven by the design work, these construction-driven feedback processes can continuously create vicious loop effects throughout the construction duration.
Summarizing this chapter, depending on characteristics of a project, feedback processes discussed thus far can have a significant impact on the project performance, in particular when a project is performed concurrently in a heavily constrained environment. Moreover, if the increased construction cost resulting from those feedback processes exceeds the possible economic gain through the reduced project duration, the effectiveness of concurrent construction should be questioned except in some cases where the market value of the shortened time is beyond the tradeoff between the possible economic gain and the increased cost. For these reasons, to effectively plan and manage concurrent construction, the feedback process should be identified before physical execution is undertaken and it should be carefully monitored throughout the project duration. However, the dynamic state of construction caused by those feedback processes makes it difficult to anticipate or measure the construction performance resulting from any planning and managerial actions in a linear fashion. This lack of capability to deal with the dynamic state of construction in the traditional network-based planning tools necessitates a systematic and dynamic approach to the planning and management of concurrent construction, in which indirect and/or unanticipated events that might happen during concurrent construction can be rigorously dealt with.
CHAPTER 3

DPM AS AN ALTERNATIVE TO NETWORK-BASED TOOLS

To date, a lot of research efforts on the simulation-based planning and control have been made to overcome problems that cannot be addressed in network-based planning methods. And, the research results have demonstrated that the potential capability of the simulation approach makes it possible to find possible problems before physical execution, substantially enhancing the effectiveness of planning and control [Martinez and Ioannou, 1997]. However, despite its potential advantages over network-based methods, only few of the existing simulation tools have overcome their practical and theoretical limitations, and have proven their applicability to real construction processes.

In this chapter, the previous researches on the simulation-based planning and control are reviewed together with their application examples. Then, Dynamic Planning and Control Methodology (DPM) is introduced as an alternative to network-based methods and compared to the previous research efforts in terms of applicability, flexibility, and reality in representation of construction processes.
3.1 SIMULATION APPROACHES

CYCLONE [Halpin, 1977] introduced a simulation technique into construction for the first time and INSIGHT [Paulson, 1983] has extended the modeling capabilities of CYCLONE together with interactive user interfaces [Paulson, 1983]. Both of them focus on the analysis of resource idleness resulting from the non-steadiness of construction processes and the minimization of its impact on the construction performance [Paulson et al., 1987]. One example application is a concrete placing model developed by Bernold [1989]. The Bernold model was used to analyze how changing conditions impact the construction process and performance. And it showed that the simulation approach could help to systematically identify factors that cause productivity interruptions and to increase the productivity by enhancing resource effectiveness. However, although both of CYCLONE and INSIGHT allow the flexible production rate of construction processes and the limitation of a specific resource level in transit or in queue, they do not provide a capability to flexibly control the resource level.

In addition, Carr [1979] developed MUD (Model for Uncertainty Determination) to evaluate the effectiveness of a schedule network, focusing on correlations between activity durations and work conditions such as site condition, equipment efficiency, and weather condition. Using MUD as a component, Padilla and Carr [1991] developed DYNASTRAT (DYNAmic STRATegy), which allows dynamic resource allocation during the simulation of construction process. In evaluating uncertainties involved in construction, DYNASTRAT recognizes uncertainty as either favorable or adverse factors [Wang and Demsetz, 2000]. Meanwhile, factor-based simulation tools have been developed with the introduction of PRODUF (Project Duration Forecast, Ahuja and Nandakumar, 1985) and PLATFORM [Levitt and Kunz, 1985]. PRODUF can generate more objective distributions of activity durations, while it requires extensive historical data. By applying heuristic rules into simulation, PLATFORM reduces the amount of input data required for the simulation of activity durations. However, PLATFORM treats all associated factors as having the same effect on the construction performance [Wang and Demsetz, 2000], which leads to less reliable performance projection.
These simulation-based planning methods have been further refined with STROBOSCOPE [Martinez, 1996]. STROBOSCOPE recognizes uncertainties involved in construction processes as a function of dynamic state of construction and describes activity duration and sequencing in terms of the dynamic information as the construction evolves [Martinez and Ioannou, 1997]. This modeling technique provides more flexibility and power in modeling the dynamic state of construction, making it possible to model the underlying process-level operations. In particular, one notable feature of STROBOSCOPE that can characterize and track individual resource units during simulation run provides more various options for simulating resource utilization processes. The effectiveness of this functional characteristic is well represented in the Tommelein [1998]'s pipe installation process model. To verify the usefulness of lean construction techniques in pipe installation, the model was structured to analyze the impact of coordination planning on resource management. With different input variables including production resources and duration, the model effectively simulated changes in pipe-spool buffer size, productivity of construction crew, and project duration [Tommelein, 1998]. In addition, STROBOSCOPE allows the expansion of its usage by providing the add-on function. An example is the STROBOSCOPE CPM add-on developed by Martinez and Ioannou [1997], which added probabilistic functions to the traditional CPM method. The applications of STROBOSCOPE, however, are still limited to a single construction process and to the simulation of physical unit flow in resource utilization. Detailed review of the previous simulation approaches is summarized in Appendix I.

3.2 DYNAMIC PLANNING AND CONTROL METHODOLOGY

In fact, significant advances in the simulation approach have been achieved through the past researches. However, only few of the current simulation-based methods have the flexibility and reliability necessary to be used as an alternative to network-based methods. For this reason, despite their potential advantages over network-based methods, they have not been yet widely accepted by the industry. For a simulation-based method to be accepted as an alternative to network-based methods, it needs to be as flexible and applicable as network-based methods are, as well as having capabilities to realize its potential advantages over network planning methods.
In this context, DPM to be presented in this thesis provides a more flexible and applicable simulation-based planning and control tool for construction projects.

The system dynamics modeling technique employed by DPM is well suited to dealing with the dynamic complexity in construction projects, which has been proven by some researchers [Ng et al., 1998; Pena-Mora and Park, 2001]. The basement construction process model developed by Ng et al. [1998] describes and captures the dynamic state of basement construction processes, by incorporating environmental and managerial factors into the process simulation as well as resource availability. Pena-Mora and Park [2001]'s model simulates multiple building construction processes in order to find the most appropriate overlapping degree between the design and construction. The model is structured to capture information flow in project monitoring and control as well as physical project execution processes. The modeling results show that it is possible to develop a generalized planning and control tool using system dynamics. In addition, DPM incorporates innovative buffering contents and concurrent engineering principles into system dynamics models as well as schedule networking concepts of CPM, PDM, PERT, GERT, and SLAM, which increases the applicability and flexibility of DPM.

Table 1 highlights DPM's innovative approaches by showing the differences between the network-based methods and DPM. As opposed to the previous simulation approach, DPM adopts the user-defined modeling approach, which will be discussed in Chapter 5. The user-defined modeling approach makes it possible to significantly increase the applicability of simulation approach in project management, while keeping the required simulation capabilities. Following Table 1, Figure 2 graphically represents the advantages of DPM over network-based methods and the existing simulation techniques. Detailed discussions on how DPM realizes the target functionality to be an alternative to the network-based methods will be presented in the following chapters.
<table>
<thead>
<tr>
<th>Description</th>
<th>CPM (Deterministic)</th>
<th>PDM (Deterministic)</th>
<th>PERT (Probabilistic)</th>
<th>GERT (Probabilistic)</th>
<th>DPM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong></td>
<td>Duration</td>
<td>Duration</td>
<td>Duration</td>
<td>Duration</td>
<td>Duration</td>
</tr>
<tr>
<td></td>
<td>Precedence</td>
<td>Precedence</td>
<td>Precedence</td>
<td>Precedence</td>
<td>Precedence Relationships (Internal &amp; External Dependencies)</td>
</tr>
<tr>
<td></td>
<td>Relationships (FS only)</td>
<td>FS, FF, SF, SS</td>
<td>Duration</td>
<td>Duration</td>
<td>Construction Characteristics</td>
</tr>
<tr>
<td></td>
<td>Lead/Lag</td>
<td></td>
<td>Probability</td>
<td>Probability</td>
<td>Resource (Labor, Material, Equipment)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Path Probability</td>
<td>Path Probability</td>
<td>Other Influences Profiles (e.g., Changes, Cash Flow, Safety, Environment, Seasonal Effects)</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>Estimated Completion</td>
<td>Estimated Completion</td>
<td>Probabilistic Estimate of Completion</td>
<td>Probabilistic estimate of completion</td>
<td>Performance Curves (Time, Costs, Quality, Safety, Environment)</td>
</tr>
<tr>
<td></td>
<td>Criticality</td>
<td>Criticality</td>
<td>Completion</td>
<td>Criticality</td>
<td>Criticality</td>
</tr>
<tr>
<td></td>
<td>Float</td>
<td>Splitting</td>
<td>Completion</td>
<td>Path Probability</td>
<td>Profile Probability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Float</td>
<td>Float</td>
<td>Float</td>
<td>Policy alternatives under “what-if” conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Policy Alternatives under “what-if” Conditions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Policy Guidelines (Labor Control, Overlapping Degree)</td>
</tr>
<tr>
<td><strong>Relationship</strong></td>
<td>Type</td>
<td>Linear</td>
<td>Linear &amp; Non-linear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External</td>
<td>Start &amp; Finish of Activities</td>
<td></td>
<td></td>
<td>Entire Duration of Activities</td>
<td></td>
</tr>
<tr>
<td>Internal</td>
<td>Not Considered</td>
<td></td>
<td></td>
<td>Considered in the form of internal constraints caused by physical constraints, resource availability, production rate, etc.</td>
<td></td>
</tr>
<tr>
<td>Resource Utilization</td>
<td>Resource Leveling and Allocation</td>
<td></td>
<td></td>
<td>Resource Availability and Utilization Rate considered</td>
<td></td>
</tr>
<tr>
<td>Progress</td>
<td>Fixed</td>
<td></td>
<td></td>
<td>Varied (depending on construction characteristics, productivity, schedule pressure, fatigue, etc.)</td>
<td>Analyzing cost-benefits tradeoffs of policies and tracing the causes of simulation results (e.g., resource bottleneck, productivity decrease, financial constraints)</td>
</tr>
<tr>
<td>Problem Solving Capability</td>
<td>Mainly using criticality on time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2: Comparison of Simulation Techniques (Dimensionless)
CHAPTER 4

COMPONENTS

METHODOLOGIES

In order to have the target flexibility, applicability and reality in representation described in Chapter 3, DPM has been developed based on the methodologies listed below. In this chapter, some of the component methodologies are introduced to help understand the fundamental concepts and logics of DPM to be discussed in the following chapters.

- Network scheduling: CPM, PDM and PERT
- Feedbacks and probability branching among activities: GERT, Q-GERT and SLAM
4.1 SYSTEM DYNAMICS

System dynamics was developed to apply control theory to the analysis of industrial systems in the late 1950's [Richardson, 1985]. Since then, system dynamics has been used to analyze industrial, economic, social and environmental systems of all kinds [Turek, 1995]. One of the most powerful features of system dynamics lies in its analytic capability [Kwak, 1995], which can provide an analytic solution for complex and non-linear systems like construction. Construction projects are inherently complex and dynamic, involving multiple feedback processes and non-linear relationships [Sterman, 1992]. In this context, a system dynamic modeling approach is well suited to dealing with the dynamic complexity in construction projects, which has been proven by some researchers [Ng et al., 1998; Pena-Mora and Park, 2001].

System dynamics modeling generally proceeds in the following steps [Kwak, 1995]: First, based on a modeler’s understanding on the system, conceptual model structures are described in the form of a causal loop diagram to show the dynamics of variables involved in the system. In a causal loop diagram, variables are connected by arrows that denote the causal influences between variables [Sterman, 2000]. Figure 3-a represents causal relationships between construction progress and schedule pressure. Appropriate

Figure 3-a: Causal Loop Diagram Notation
schedule pressure can increase productivity, which can facilitate the construction progress. At the same time, higher schedule pressure can also slow down the construction progress by lowering work quality. As a result, increased or decreased construction progress affects schedule pressure again, forming feedback loops.

Having a causal loop constructed, variables in the model structures come to have quantitative attributes based on the relationships built in the causal loop diagram. This step also includes the identification of stock and flow structures (see Figure 3-b), which characterize the state of the system and generate the information, upon which decisions and actions are based, by giving the system inertia and memory [Sterman, 2000]. Stocks represent stored quantities and flows control quantities flowing into and out of stocks. Once this model formation step is done, the completed model needs to be tested and validated in accordance with the purpose of the model. Finally, the validated model is applied to solving the given problems.

![Figure 3-b: Stock and Flow Structure](image)

### 4.2 CONCURRENT ENGINEERING

Concurrent engineering has been developed in other industries' product development activities to cope with competitive business environments that require the industries to develop and market products faster [Eppinger et al., 1992]. Concurrent
Engineering aims principally at reducing the duration of engineering time and costs. One of the major challenges facing concurrent engineering lies in an overlapping practice. Properly overlapped tasks can facilitate development progress, while overlapping practice without careful management may increase the development cost and worsen the product quality [Eppinger et al., 1993].

Eppinger [1997] classifies overlapping practices in terms of upstream evolution and downstream sensitivity, focusing on transferring information that is derived from design parameters. Upstream evolution describes the ability of the upstream to provide finalized information, with which a downstream task can proceed. Downstream sensitivity describes the sensitivity of the downstream to changes in an upstream task. By understanding evolution and sensitivity for sequential task pairs, an overlapping strategy can be chosen [Eppinger, 1997]. Figure 4 shows four possible overlapping practices according to the characteristics of upstream and downstream.

![Overlapping Framework](adopted from Eppinger, 1997)

Figure 4: Overlapping Framework [adopted from Eppinger, 1997]
Eppinger explains overlapping effectiveness based on information transfer patterns between upstream and downstream activities. Poor overlapping does not lead to shorter project duration due to earlier transfer of preliminary data that subsequently changes, resulting in wasteful iteration. On the other hand, effective overlapping can accelerate a task process by changing information transfer practices. Upstream task parameters and requirements should be fixed early enough for downstream task to begin prior to the completion of upstream tasks. Another alternative is that the downstream task uses preliminary data to begin early and finalizes with fast iteration. In this case, the downstream task should be flexible so that final data will not affect it. The implications of these overlapping practices provided an insight into developing an overlapping framework for construction.

### 4.3 OVERLAPPING FRAMEWORK FOR CONSTRUCTION

Eppinger' framework has been applied to construction projects by Pena-Mora and Li [2001]. The types of activity characteristics in their frameworks are different because Eppinger focuses on information transfer between overlapped activities, while Pena-Mora and Li deal with the transfer of physical production units between the activities. Pena-Mora and Li [2001] argue that task production rate, upstream production reliability, and downstream task sensitivity are the activity characteristics used to determine effective overlapping strategies in construction.

Given the task production rate, upstream production reliability, and downstream task sensitivity, it is possible to provide a framework, by which activities with certain characteristics should or should not be overlapped [Pena-Mora and Li, 2001]. The framework presented in their research focuses on the fact that overlapping practices should vary depending on the characteristics of construction activities. For instance, foundation work is insensitive to changes in the upstream excavation work, while finishing is very sensitive to any changes made in preceding processes such as
Figure 5: Overlapping Framework for Construction
partitioning and concrete pouring. Thus, different overlapping strategies should be applied to these processes.

The framework in Figure 5 deals with activities that have a sequential relationship between them such that 25% of the upstream work must be completed before the downstream work can start. Overlapping alternatives are made on the basis of upstream activity progress with a range from 25% to 175% of the upstream progress. The framework is divided largely into four quadrants, which consist of possible combinations of two different production types in upstream and downstream. In each quadrant, overlapping alternatives are made based on the upstream production reliability and downstream task sensitivity. A reliable upstream activity produces less defective work thus downstream work can begin sooner. However, with an unreliable upstream work, the degree of overlapping should be reduced to avoid the possibility of downstream changes. Similarly, a sensitive downstream is more vulnerable to the upstream changes thus the overlapping degree should be reduced. By incorporating all these characteristics into the framework, it is possible to formulate an overlapping strategy for construction activities that have precedence relationships.

The component methodologies described thus far provided conceptual foundations for the development of DPM. In addition, concepts from all of the component methodologies are imbedded in system dynamics models that have been developed to materialize DPM functions. As will be discussed in Chapter 6, they are systematically integrated so that DPM can have necessary flexibility, applicability, and reality in representation to be an alternative to the traditional network-based planning and control tools.
CHAPTER 5

FUNDAMENTALS OF DPM

This chapter presents fundamental concepts and logics of DPM, which have been developed to address some of the challenges involved in the planning and control of construction. As discussed in the previous chapters, construction processes are inherently dynamic and complex, having multiple feedbacks, in particular when they are performed concurrently. This dynamic and complex feature makes construction unstable and creates a lot of iterations among construction processes, which often lead to schedule and cost overruns. In fact, construction iterations resulting from uncontrollable outside factors are inevitable but most of non-value adding iterations can be reduced, once they are identified in advance and are managed with a well-prepared plan. For these reasons, DPM focuses on reducing the impact of non-value adding iterations during construction by identifying the cause of the iterations and assessing their impact magnitude along with impact paths. In addition, throughout planning and control, DPM utilizes the past experience on a project that tends to be wasted in the traditional network-based approach, which can increase the accuracy of performance projections and the robustness of planning. Apart from functionality, in DPM the same emphasis is given to users’ accessibility to a tool that has been an obstacle to the use of simulation tools. By allowing users to define contents of pre-structured models, DPM increases its applicability in line with having advantages of the simulation approach. The fundamentals of DPM are discussed in detail in the following sub chapters.
5.1 USER-DEFINED MODELING APPROACH

Although simulation-based planning tools have their own advantage to deal with dynamic complexities in the construction project execution, the application of existing tools has been limited to a specific set of activities on a project due to the inflexibility of the tools. In contrast, the application of DPM is not limited to a specific set of activities on projects by introducing the concept of user-defined modeling. For general use, DPM has generic parameters and structures, common to almost all construction projects, with the ability to customize for a specific project and to describe project activities. As a result, users can define contents of DPM models that have been pre-structured, which significantly increases the applicability of the simulation-based approach into construction projects in a real world setting.

![User-Defined Modeling Approach Diagram]

Figure 6: User-Defined Modeling Approach
One thing to note associated with this approach is that the success of the user-defined modeling depends on how well pre-structured models can represent construction processes and dynamics involved in a given project and how reliable the simulation output of the models is. In fact, these can be the most challenging issues in the user-defined modeling approach where the simplification of model structures is attempted. For these reasons, in DPM the most influential construction dynamics and characteristics are identified and they are converted into the generic parameters and model structures of DPM in a simplified form.

5.2 CONSIDERATION OF FEEDBACKS

DPM focuses on capturing feedback processes involved in construction projects. These feedback processes contribute to generating indirect and/or unanticipated events during the project execution and make the construction process dynamic and unstable, which cannot be captured in the traditional planning tools.

Suppose that construction processes consist of a set of steps conceptualized in Figure 7. When a certain control action is taken to reduce variations from the planned performance, the action can fix problems and enhance the construction performance but at the same time it can worsen the performance in another area due to side effects of the action. For example, when a construction project is behind schedule, one possible action to meet the original schedule is to change equipment. By replacing the current equipment with high performance equipment, it is possible to facilitate the construction process. However, it may take time for workers to get familiar to operating the new equipment or coordinating with other subsequent processes may become more difficult.

As a result of low productivity and increased coordination problems, it is also possible that changing equipment can further delay the construction schedule. DPM assists in identifying this kind of feedbacks that are caused by human response to control
actions. In addition, DPM captures vicious feedback processes that are triggered by process constraints, construction policies or work environment. Based on identified feedbacks, DPM simulates construction processes more realistically before actual resource commitment, which increases the reliability of planning and control actions for construction projects.

Figure 7: Feedback Processes in Project Execution
5.3 CAPTURING CONSTRUCTION DYNAMICS

As discussed in Chapter 5.2, feedbacks involved in construction processes make construction dynamic and unstable, resulting in non-value adding iterations among construction activities. Although many other factors can exist, in DPM, changes, work dependencies among activities, construction characteristics, and human responses to work environment and policies are considered as major factors that are associated with those feedbacks. In particular, changes are distinguished from rework, which is a unique feature found in construction, and they play an important role in making construction dynamic and unstable, and creating non-value adding construction iterations. In this chapter, explanations on how DPM captures construction dynamics are made focusing on changes and work dependencies.

5.3.1 Change as Iteration Trigger

Construction processes involve a lot of value adding and non-value adding iterations, in particular when they are performed concurrently. Some of them are inevitable, since they result from complexities and uncertainties embedded in the construction process or uncontrollable outside factors like weather conditions. However, most of non-value adding iterations can be reduced, once they are identified in advance and are managed with a well-prepared plan.

In construction, iterations are mainly triggered by changes. Normally, changes refer to work state, processes, or methods that deviate from the original plan or specification. They usually result from work quality, work conditions or scope changes. In addition, changes that have been already made (denoted as Changes as Result in Figure 8) can be the source of subsequent changes in either concurrent, succeeding or preceding tasks (denoted as Changes as Source in Figure 8). For example, changes in the design work
Managerial Change Processes: A B C

Unintended Change Processes: E F G

Figure 8: Changes as Iteration Trigger
that have been made by mistake can cause subsequent changes in construction. In this case, the design changes are a result to the designer, while they can be a need for changes to the construction crew.

Changes can be also categorized as unintended changes and managerial changes (denoted as Changes as Behavior in Figure 8). Unintended changes occur without the intervention of managerial actions. The arrows labeled $E$, $F$, and $G$ in Figure 8 illustrate the unintended change process. Meanwhile, managerial changes are made by managerial decisions during quality management or project monitoring and control. As illustrated in Figure 8, once changes occur during construction ($A$ and $B$), changes result in either subsequent changes ($C$) or rework ($D$), depending on managerial decisions.

Table 2: Characteristics of Change and Rework

<table>
<thead>
<tr>
<th>Des.</th>
<th>Purposes</th>
<th>Object</th>
<th>Scope of Work</th>
<th>Triggering Another Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managerial Changes</td>
<td>Minimizing the impact of changes that have already occurred by adopting a different method or process than in the original plan and specification.</td>
<td>Preceding or succeeding tasks</td>
<td>Vary, depending on sensitivity</td>
<td>May trigger</td>
</tr>
<tr>
<td>Rework</td>
<td>Achieving what are originally intended in the plan and specification.</td>
<td>The problematic tasks</td>
<td>Same as the scope of problematic tasks</td>
<td>None</td>
</tr>
</tbody>
</table>

Both change and rework are done in the form of either ‘adding’, ‘deleting’ or ‘replacement (deleting and adding)’. However, given the same problem, they have different behavior patterns, since change and rework have different characteristics, as summarized in Table 2. For example, in Case I on Figure 9, given the problem (a hump on the concrete surface), rework would be done by deleting the problem, while
change would be done by adding some more concrete. In addition, in Case IV where floor tiling has been finished with less than the required height, although both change and rework have the same behavior pattern (replacement) in solving the problem, the object would be the problem area in rework, while the previous work would be the object in change. Due to these differences in handling problematic work, managerial decisions on whether to change or rework are usually made based on the analysis of each option’s impact on the construction performance in terms of time, costs and quality. In construction, the change option is more general. Since construction has a physical manifestation, construction rework is usually accompanied with the demolition of what have been already built, which normally has a bigger direct impact on the construction performance than the change option.

By adopting the change option, it is possible to avoid rework on problematic tasks that may require more resources. However, as discussed above, changed tasks can also become a change source that can cause another subsequent changes, which might have more impact on the construction performance than the rework option in a certain condition. For example, the increased concrete height in Case I and Case III on Figure 9 may trigger subsequent changes in succeeding tasks, i.e., reducing the size of ventilation ducts. In addition, in Case V on Figure 9 where some of piles have not been correctly positioned, it may be possible to proceed with the superstructure without correcting the position of the piles by changing the position of columns. However, this change option may necessitate unplanned cantilever construction in order to keep the original floor layout, which needs to be evaluated as compared to re-driving the piles. Consequently, the decision on the change option needs to be carefully made based on a good understanding of how changes evolve to non-value adding iterations, which can create unplanned and indirect side effects.
<table>
<thead>
<tr>
<th>Case</th>
<th>Planned Performance</th>
<th>Change (Unintended)</th>
<th>Rework (Managerial)</th>
<th>Possible Subsequent Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case I</td>
<td>Concrete Pouring</td>
<td></td>
<td>Deletion</td>
<td>Adding</td>
</tr>
<tr>
<td>Case II</td>
<td>Work Already Done</td>
<td></td>
<td>Deletion</td>
<td>Adding</td>
</tr>
<tr>
<td>Case III</td>
<td>Floor Tiling</td>
<td></td>
<td>Replacement</td>
<td>Adding</td>
</tr>
<tr>
<td>Case IV</td>
<td>Piling</td>
<td></td>
<td></td>
<td>Adding</td>
</tr>
<tr>
<td>Case V</td>
<td></td>
<td></td>
<td></td>
<td>Keep Current State</td>
</tr>
</tbody>
</table>

Figure 9: Behaviors of Change and Rework
5.3.2 Dependencies

Dependencies involved in construction processes play the role of the main constraints to the construction progress, making the processes dynamic and unstable. Accordingly, to effectively plan a construction project, these dependencies need to be identified and modeled in detail. There exist two kinds of dependencies in the process of a construction project. That is, external dependency among activities and internal dependency within an activity.

External Dependency

The external dependency captures the dependency relationship between activities, which has been represented by the precedence relationship in the traditional network-based methods. While the precedence relationship deals with only the start and finish of an activity in a static manner the inter-activity dependency represented here can dynamically describe the dependency relationship throughout the activity duration. Figure 10 shows some examples of external dependencies in construction projects. In the graphs, the horizontal axis represents the progress of upstream work, while the vertical axis represents the progress of downstream work.

Figure 10: Examples of External Dependency [Ford and Sterman, 1997]
For example, the dependency relationship between excavation and foundation work can be represented by Graph A in Figure 10. The downstream activity, foundation work, is scheduled to start at 50% completion of the upstream activity, excavation, and thereafter foundation work can proceed in proportion to the progress of the excavation. In contrast, Graph B in Figure 10 represents the concurrence relationship such that downstream work can start only after partial or entire completion of upstream work. In this case, there is no further dependency between the overlapped processes once downstream work gets started.

**Internal Dependency**

The internal dependency represented here has not been considered in the traditional network-based methods. However, a construction process involves procedural or physical constraints that can create dependencies between tasks within an activity. For example, the concrete formwork requires forms to be installed, inspected for proper installation, and corrected if the installation is unacceptable. This kind of procedural constraint exists in most construction processes and can affect the construction progress. In addition, some activities in construction projects such as structural work have physical constraints such that lower floor work should be completed before any work on the following floor, because lower floor supports those above. In the dynamic project model, such a constraint is simulated with the internal concurrency depicted in Graph A on Figure 11. Meanwhile,

![Graph A and Graph B](image)

*Figure 11: Examples of Internal Dependency [Ford and Sterman, 1997]*
the internal concurrency depicted in Graph B on Figure 11 can be applied to the activities that do not have physical constraints such as partitioning, finishing, and external work. This internal concurrency relationship allows those works to proceed at any time given enough resources.

All of the dependencies discussed above can constrain the construction process and influence the construction performance. In particular, when a project is concurrently performed, they may have a significant impact on the project schedule and costs. In order to represent these dependencies in line with precedence relationships, which are finish-to-finish, finish-to-start, start-to-start, and start-to-finish, the dynamic project model has a sub model sector designed for each task and activity to be linked in concurrence relationships. In addition, a relationship that can describe reprocess iterations caused by the downstream work is also represented in the model structure. All of these concurrence relationships are embedded in each generic process model structure of DPM.

In addition to changes and work dependencies that have been discussed thus far, there can be many other factors that determine the behavior of the construction system. As discussed in Chapter 4.3, construction characteristics of an activity represented by activity production type, upstream reliability and downstream sensitivity contribute to creating construction dynamics. In addition, workers' response to schedule pressure, work environment, and managerial decisions also can affect the construction performance, generating unanticipated side effects, as conceptualized in Figure 3-a. In conclusion, all of these factors have impacts on the construction system, having different magnitudes and types, which makes it important to understand their roles in the planning and control of construction projects.
5.4 REDUCING SENSITIVITY TO CHANGES

Chapter 5.3 discussed that construction processes inherently involve many non-value adding iterations and they are mainly triggered by changes. In particular, when construction processes are performed concurrently, those iterations have more impact on the construction performance. DPM attempts to reduce non-value adding iterations involved in construction processes by reducing the sensitivity of a construction system to unintended changes. To do this, DPM provides an appropriately pooled, located, characterized and sized buffer, which called *Reliability Buffer*.

5.4.1 Traditional Buffering Practice

Buffering is a common practice in project planning. Traditionally, project managers or schedulers have used a time contingency to guarantee the completion time either of an activity or a project (the contingency buffer). Practices on the contingency buffer vary depending on the level of scheduling as summarized in Table 3. At the lower-level scheduling, in order to guarantee the schedule performance of individual activities, time contingencies are normally added to the most-likely estimate of activity durations, mostly by subcontractors or sub-divisions. For instance, designers may apply a pessimistic estimate to scheduling the schematic design instead of its average duration so that they can keep their promise on the delivery time in case the design work is delayed. Meanwhile, when a higher-level schedule is made based on the collected lower level schedules, it is often that contingency factors are again added to the preceding activity in order to avoid subsequent schedule disruptions in the succeeding activity. For instance, knowing or not knowing that contingency factors have been already incorporated into the activities, a project manager may put a several days contingency in the excavation work so that the foundation work can start as scheduled, even if the excavation work is delayed few days.
Table 3: Traditional Buffering Practice

<table>
<thead>
<tr>
<th>Level</th>
<th>Scheduled by</th>
<th>Buffering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower</td>
<td>Subcontractors or Subdivisions</td>
<td>Adding some contingency to individual activities to guarantee the schedule performance of each individual activity</td>
</tr>
<tr>
<td>Higher</td>
<td>General Contractors or Project Managers</td>
<td>Given precedence relationships, adding some contingency to the preceding activity to avoid subsequent schedule disruptions in the succeeding activity.</td>
</tr>
</tbody>
</table>

**Lack of Characteristics**

Based on the scheduling sequence, the contingency buffer is normally conceived as being positioned at the end of the activity duration. However, since the contingency buffer is characterized as nothing more than some more time, it is difficult to locate and utilize the buffer. For this reason, once added to the duration of an activity, the contingency buffer tends to be used as part of an activity without clear distinctions from the original duration, then, stopping being a contingency. As a result, the contingency buffer is not so helpful to protect the schedule performance, often creating a rubber band duration. This time added to the original duration only result in schedule expansion. When people realize that they have more time than the time known to complete a task, their work productivity usually goes down [Sterman, 2000], often with the task being deferred to the last minute.

**Losses on a Merging Point**

In addition, as Goldratt [1997] points out, the contingency buffer that focuses only on an individual activity is often inefficient on the merging point of a schedule network. Since an activity dependency relationship can create a bottleneck on the merging point, delays accumulate but advances are used up in the traditional buffering practice.
Suppose that the duration of each activity ‘A’, ‘B’, and ‘C’ in Figure 12 is 15 days, which contains 5 days contingency. In addition, each of the activity ‘A’ and ‘B’ has a finish-to-start dependency relationship with the activity ‘C’. Assuming that the activity ‘A’ is finished 5 days earlier and the activity ‘B’ and ‘C’ are finished on time, the total duration will be 30 days. Due to the precedence relationships involved in the activities, the activity ‘C’ cannot start until the activity ‘B’ is done even though the activity ‘A’ is done 5 days earlier than the scheduled time. As a result, the start time of the activity ‘C’ is governed by the activity ‘B’ and a schedule advance achieved in the activity ‘A’ does not add any benefit to the project. This result also implies that delays made in the activity ‘B’ will be passed on to the activity ‘C’. For example, assuming that it took 20 days to finish the activity ‘B’ the 5 days delay made in the activity ‘B’ delays the start of the activity ‘C’ for 5 days and, as a result, the total duration will be 35 days.

Inefficient Sizing

In addition, the traditional way of sizing a buffer can also contribute to making the contingency buffer ineffective. In fact, the size of a buffer is normally decided based on individual experiences and assigned in a uniform way. Furthermore, the application of such an inappropriately sized buffer is often duplicated by different project functions, as observed in Table 3. As a result, the traditional contingency buffer may not be
efficient in protecting the project schedule performance, resulting in an unnecessary resource idle time.

5.4.2 Reliability Buffer

In contrast to the traditional contingency buffer, the reliability buffer presented in this thesis aims to systematically protect the whole project schedule performance from being disrupted by failures in individual activities. To do that, the reliability buffer pools, re-locates, re-sizes, and re-characterizes the contingency buffers. In addition, to effectively control schedule deviations from the initial plan, the location and size of a reliability buffer are dynamically updated throughout the construction duration. Details on reliability buffering are discussed below.

Buffering Logistics

Excess time beyond the known average duration is normally conceived as a contingency buffer that has been put during scheduling processes. However, in reliability buffering a fraction of duration, for which people could get a lax attitude, is also seen as an implicit contingency buffer. Reliability buffering starts with taking off such contingency buffers that are fed explicitly or implicitly in individual activities. Taking off contingency buffers from individual activities can make the activities benefit from appropriate schedule pressure, overcoming ‘the last-minute syndrome’. In addition, the reliability buffer is fed in the front of the downstream activity in precedence relationships and is characterized as a time to find problems or finish up work in the upstream and ramp up resources on the downstream activity. By putting buffer at the beginning of activities instead of at the end of activities, the reliability buffer can deal with the issue of ill-defined tasks that may require time for definition. This makes it possible to focus on activities having problems before they activate a domino effect, as it might happen in the traditional buffer.
Meanwhile, by locating a buffer at the beginning of the activity duration, it is also possible to reduce losses on the merging point of a schedule network. Continuing with the contingency buffer example, 5 days contingency buffers are taken off from all the activities (this is based on the initial assumption that a 5 days contingency buffer are applied to all the activities). Then, a 5 days reliability buffer (actually, buffer size varies depending on associated activity characteristics) is fed in the front of the activity ‘C’, as depicted in Figure 13. As a result, the activity ‘A’ and ‘B’ have 10 days, and ‘C’ has 15 days for the completion of each activity. Assuming that the actual durations of the activity ‘A’ and ‘B’ are the same as in the previous example, the activity ‘B’ is now delayed 5 days. Despite the delay in the activity ‘B’, the activity ‘C’ can start with reliability buffering (as a result of dynamic buffering, the precedence relationship between the activity ‘B’ and ‘C’ has been changed from a finish-to-start to a finish-to-start with a 5-days lead. In fact, the size of the updated reliability buffer and the lead time are not necessarily 5 days in our approach. However, for explanation purpose this example has been updated with the initially planned buffer size and accordingly the precedence relationship has a 5-days lead. Details on dynamic buffering will be discussed later.)

During the buffering period, it is possible for workers in the activity ‘C’ to find upstream problems or possible technical or functional mismatches before the activity ‘C’ starts by checking the work that has been done in the activity ‘A’ and ‘B’ thus far. Once problematic work is found, they can ask upstream workers to correct the work, which would otherwise impact the activity ‘C’ in progress. In addition, the reliability buffer can also provide a time to thoroughly review and prepare the activity ‘C’, ramping up necessary resources. Consequently, the activity ‘C’ can become more reliable, which increase the possibility of finishing the activity ‘C’ within the reduced duration (10 actual working days less 5 days of the reliability buffer) together with increased productivity resulting from schedule pressure. On condition that the activity ‘C’ is complete as scheduled, the total duration will be 25 days.
Buffer Sizing

How to size a buffer is another crucial issue in buffering. The size of a buffer needs to be long enough to keep downstream activities reliable. However, an inappropriately sized buffer can also create unproductive idle time, which may delay the entire schedule. For this reason, a buffer needs to be sized in a systematic way rather than relying on individual experiences. In this regard, reliability buffering provides a systematic way in sizing a buffer based on a simulation approach to be explained later in this thesis.

Pena-Mora and Li [2001]'s research suggests that buffering practice for construction activities should vary depending on the activity production rate, the upstream reliability, and the downstream sensitivity. A backbone concept underlying their research is that the degree of overlapping should be decided in a way that enough time to discover and fix problems made in upstream can be secured before downstream starts. According to their research, more overlapping can be allowed when upstream evolves fast or when downstream evolves slowly. In other words, slow upstream or fast downstream requires more time buffer to discover and fix problematic upstream work. This is because in both cases, slow upstream and fast downstream, there exist more chances for problematic upstream work to impact downstream in progress. In addition,
upstream reliability also governs the degree of overlapping because more problematic upstream work caused by low reliability could lead to more downstream changes. Accordingly, the less reliable upstream work is, the more time buffer downstream work needs. Lastly, the degree of overlapping also depends on downstream work sensitivity because the upstream work impact varies depending on the sensitivity of downstream work to the upstream work. For this reason, more sensitive downstream work requires more time buffer.

The implications of all those buffer size determinants discussed above can be good guidelines for effective buffering, which will be validated by the dynamic project model developed in this thesis. However, effective buffer sizing is also closely related to many other construction conditions including workforce control and project monitoring policies, as well as the buffering size determinants. In addition, each determinant normally has a different effect on effective buffering and their effects also vary depending on precedence relationships. For example, when activities have a start-to-start precedence relationship, the effect of upstream production type is reduced, as the lag time increases. For these reasons, the reliability buffer adopts a simulation approach in sizing a buffer, which makes it possible to provide an activity with an appropriate buffer size in a systematic manner, given certain activity characteristics and project control policies.

**Dynamic Buffering**

In order to effectively control schedule deviations from the initial plan, initially planned buffers need to be continuously updated during construction. Normally, a construction project evolves throughout the project duration, during which the characteristics of a construction system continuously change, resulting in changes in the construction performance. As in the example on Figure 13, if the upstream work is delayed less than the buffer size (e.g., 5 days), the initially planned buffer can absorb delays made in the upstream work (e.g., up to 5 days). However, once the upstream work is further delayed and as a result, the applied buffer is used up, the delayed upstream work comes to impact the downstream work schedule performance.
With a static buffering approach, delays made in the upstream work are directly passed on to the downstream work by simply pushing forward the initially planned durations of the buffer and the downstream work, as depicted in Figure 14-a. In contrast, having a dynamic buffering applied, the impact of the upstream work schedule disruptions on the downstream work can be minimized by dynamically updating the location and size of a buffer based on the information obtained from the actual performance and the forecast of the remaining construction performance. In principle, buffering is updated in a way that the initially planned downstream work start can be protected from upstream work schedule disruptions and, if any schedule advances are made in the upstream work, the downstream work can benefit from the schedule advances. In case the actual upstream work duration is longer than its initially planned duration, this dynamic approach helps to minimize the impact of upstream work schedule disruptions.

**Figure 14-a: Dynamic Buffering I**

For example, in Figure 14-a, by applying a dynamic buffering approach, the initial upstream work duration \((D_i)\) is updated at time ‘\(t_s\)’ with the forecasted duration \((D_f)\), based on which the buffer size \((B_d)\) are also newly decided. The updated buffer size \((B_d)\) is likely bigger than the initially planned buffer \((B_i)\), since more changes are possibly involved in the upstream work than expected, which requires a longer buffering
period. In addition, the associated precedence relationship is also changed in order to protect the original downstream work start time. As a result, the initial finish-to-start relationship comes to have a lead equivalent to the delay in the upstream work \((t_3 - t_2)\). Assuming that the upstream work is actually finished at time \(t_3\), unnecessary downstream work resource idle time in the static buffering can be saved as much as \(t_5 - t_4\).

One thing to note is that although there can exist gaps between the actual duration \((D_a)\) and the forecasted duration \((D_f)\), the gaps usually narrow down, as more information is obtained in successive monitoring and control interventions, and construction progresses towards completion. On the other hand, it is also possible for the upstream to be finished earlier than the initial schedule. In this case, the dynamic approach makes the project schedule benefit from schedule advances achieved in the upstream work. For example, the activities and the reliability buffer in Figure 14-b are re-scheduled at time \(t_c\) as in the example on Figure 14-a. As a result of dynamic buffering, it is possible to save unnecessary downstream work lead time as much as \(B_d - B_s\).

![Figure 14-b: Dynamic Buffering II](image)

As summarized in Table 4, the patterns of buffer location and size changes are different depending on associated precedence relationships and the simulation result of the upstream work duration and characteristics. At each schedule update...
time, the remaining construction performance is forecasted based on the information obtained thus far. According to simulation results, appropriate buffer sizes and locations for activities are decided in a way that the downstream work can benefit most from schedule advances in the upstream work, minimizing the impact of upstream schedule disruptions. As a result, buffer sizes and locations are continuously changed throughout the project duration. In addition, associated precedence relationships may also change according to buffer size and location changes, in case the initial precedence relationship is a finish-to-start.

Summarizing this chapter, the reliability buffering aims to aggressively protect the project schedule performance by pooling, re-locating, re-sizing, and re-characterizing the contingency buffer that is fed explicitly or implicitly in individual activities. As a result, activity durations and associated precedence relationships can be also changed at each planning and control time in a way that the impact of schedule disruptions in an individual activity on the project schedule performance can be minimized. Reliability buffering steps include 1) taking off contingency buffers from individual activities and pooling them, 2) re-sizing the contingency buffer based on simulation results, given project activity characteristics and project control policies, 3) putting the re-sized buffer in between activities in precedence relationships, more precisely at the beginning of the succeeding activity duration, 4) characterizing it as a time to ramp up necessary resources for the downstream work and to find problematic upstream work that would impact the downstream work in progress, 4) using the remaining schedule contingencies in the buffer pool as a path pool buffer, in order to absorb schedule disruptions in individual activities, if any (its role is similar to Goldratt’s project buffer [1997] but the logistics of the path pool buffer is different from that of the project buffer), and 5) dynamically updating the location and size of buffers by utilizing the information obtained from the actual performance.
Table 4: Buffer Location and Precedence Relationship Change Patterns

<table>
<thead>
<tr>
<th>Descriptions</th>
<th>Without Lags/Lags</th>
<th>With Leads</th>
<th>With Lags</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conditions</td>
<td>Precedence Relationships</td>
<td>Conditions</td>
</tr>
<tr>
<td>FS</td>
<td>Initial</td>
<td>FS</td>
<td>FS</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>FS</td>
<td>FS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FS</td>
<td>FS</td>
</tr>
<tr>
<td>FF</td>
<td>Initial</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF</td>
<td>FF</td>
</tr>
<tr>
<td>SS</td>
<td>Initial</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SS</td>
<td>SS</td>
</tr>
<tr>
<td>SF</td>
<td>Initial</td>
<td>SF</td>
<td>SF</td>
</tr>
<tr>
<td></td>
<td>Update</td>
<td>SF</td>
<td>SF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SF</td>
<td>SF</td>
</tr>
</tbody>
</table>

*Note: Di: Initially Planned Duration  Bi: Initially Planned Buffer Size  BS: Initially Planned Buffer Start Time  Li: Initially Planned Lead or Lag  Df: Forecasted Duration  Bf: Forecasted Buffer Size  BSf: Forecasted Buffer Start Time*
Contingency Buffer

Activity B

Re-Sizing, Re-Locating, & Re-Characterizing

Reliability Buffer

Taking off & Pooling

Activity A

Re-Sizing, Re-Locating, & Re-Characterizing

Activity C

Path Pool Buffer

Figure 15: Reliability Buffering Steps
5.5 SMART SYSTEM

Normally, the past experience on a construction project tends not to be utilized in the traditional planning and management approach. In contrast, DPM attempts to convert the past experience to knowledge. This knowledge-based approach makes DPM smarter and more accurate, as construction proceeds. At the beginning of construction, DPM would be established with many assumed variables, but as construction advances, actual performance will replace the initial input values for a more realistic projection of the construction performance. Through the continuous feedback, the difference between assumptions and the actuals is being reduced. The actuals will be also stored in a database for future utilization by an agent on the DPM system.

This chapter presented the fundamental concepts and logics underlying DPM, which are user-defined modeling approach, consideration of feedbacks, capturing construction dynamics, reducing sensitivity to changes, and smart system. All of the concepts and logics of DPM have been derived from closer observations of construction processes and practices thus far, and they have been elaborated, taking consideration into functional requirements to achieve DPM target goals. These fundamental concepts and logics have been materialized by systematically integrating DPM's component methodologies, as will be discussed in Chapter 6.
Check whether the current data is well fit to explain the actual performance.

Figure 16: Smart System
CHAPTER 6

IMPLEMENTATION

DPM has been implemented by materializing its fundamental concepts and logics described in Chapter 5. In particular, the implementation of DPM focused on meeting functional requirements for DPM to be an alternative to the traditional network-based planning and control methods, which are to increase flexibility in simulation modeling and applicability in use, while keeping the capability to simulate construction processes and capture dynamics involved in the construction process. As for system constitution, all of the DPM component methodologies were systematically integrated into system dynamics models and a system interface with users and databases was elaborated using a Java program.

Based on the discussions thus far, this chapter illustrates how the concepts and logics of DPM have been materialized, using the component methodologies. First, Chapter 6.1 overviews the DPM system, during which the functions and system architecture of DPM are presented. Then, Chapter 6.2 describes the system dynamics models, in which all of the components methodologies are integrated, focusing on the model structure that represents the generic process of a construction project.
6.1 SYSTEM ARCHITECTURE

In this chapter, the system architecture of DPM is overviewed. First, how components methodologies have been integrated into system dynamics models is discussed. Then, the discussion continues with DPM functions and advantages over the existing planning and control methods. Finally, the system interface and collaboration scheme of DPM are presented.

6.1.1 Integrating Methodologies

DPM has been developed by incorporating reliability buffering contents, the concept of strategic planning and concurrent engineering principles into system dynamics models as well as schedule networking concepts of CPM, PDM, PERT, GERT, Q-GERT, and SLAM. The incorporated components methodologies play their roles in the DPM system as detailed below.

Strategic Planning and DSM

As conceptualized in Figure 17, DPM implements the concept of strategic planning by representing input data with DSM (Design Structure Matrix) and introducing smart cells. DSM representation and smart cells make it easier to recognize relationships between activities. In addition, they well organize input data so that input data can be effectively utilized by other DPM functions. There are two kinds of smart cells. Smart cells for an activity (Figure 18-a) contain information on activity duration and activity characteristics such as production types and reliability. Meanwhile, those for relationship (Figure 18-b) have information on the relationship of associated two activities. Relationships represented by smart cells include reprocess iteration relationships (RI) as well as precedence relationships (FS, FF, SF, SS). For example, the smart cell in Figure 17 represents the relationship such that problems made during the activity E can
Figure 17: Integrating Methodologies
Figure 18-a: Smart Cell for Activity
Figure 18-b: Smart Cell for Relationship

Probability of realizing this dependency relationship

Precedence Relationships (FF,FS,SS,SF) or Reprocess Iterations (RI)

Lag or Lead Time associated with this dependency relationship

To be used to break down activities into smaller segments. In this case, 20% of downstream work can start independent of its upstream, because that portion of work is classified as general, administrative work, which is not dependent on upstream work.

Unique to the set of UP and DN. i.e. Sensitivity of 'C' to 'A' can be different from sensitivity of 'C' to 'B'

* Impact of Reprocess Iterations: function of Dependency Probability, Upstream Sensitivity, and Downstream Reliability
cause reprocess iterations between the activity E and C. In addition, Sensitivity in the smart cell indicates that the vulnerability of the activity C to the problems made in the activity E is low.

**CPM & PDM**

DPM represents CPM and PDM by controlling work dependencies between activities. For example, the smart cell in Figure 17 indicates that associated two activities have a FS relationship with 5 days lag. Given the precedence relationship, DPM constrains the construction process by controlling the downstream work dependency to the upstream work progress. On thing to note associated work dependencies is that in DPM, non-linear dependency relationships can be represented as well as linear dependency relationships, as discussed in Chapter 5.3.2.

**PERT**

PERT is taken into consideration in DPM by classifying activity durations into ‘most-likely’, ‘pessimistic’, and ‘optimistic’. Once different types of durations are provided through a smart cell, DPM generates spread of simulated actual durations having confidence bounds. In addition, DPM can represent probabilistic branching of PERT together with other sensitivity simulations. As a result, three different value types (most-likely, pessimistic, and optimistic) can be applied to activity duration and other variables at the same time.

**GERT, Q-GERT and SLAM**

For the implementation of the concepts from GERT, Q-GERT and SLAM, reprocess iterations caused by downstream work changes are considered in DPM. Once the relationship type that indicates the possibility of those iterations (RI) is marked in smart cells, DPM creates reprocess iterations between the upstream work and the downstream work, when changes occur in the downstream work. The probability that
governs these iteration processes can be also defined by users. For example, suppose that as shown in Figure 17, the activity C has a RI relationship with the activity E, and the probability of realizing the relationship is 100%. In this case, a certain amount of changes made in the activity E can trigger the same amount of subsequent changes in the activity C.

**Concurrent Engineering, Critical Chain and Overlapping Framework**

The concepts and principles of concurrent engineering, critical chain and overlapping framework are incorporated into DPM through reliability buffering. As detailed in Chapter 5.4, reliability buffering aims to effectively deal with potential problems that might be caused by increased concurrency between activities, which is also the subject that concurrent engineering deals with. In addition, reliability buffering identifies the construction characteristics defined in overlapping framework (production types, upstream reliability, and downstream sensitivity) as the most important factors that determine the effective buffering time and size. Critical chain also contributes to finding the effective buffering size by helping to identify contingency factors imbedded in individual activity durations. By systematically integrating all of these methodologies through reliability buffering, it is possible to more effectively handle problems that are often encountered in concurrent construction.

**System Dynamics Project Management Models**

In terms of providing the possibility of using system dynamics models in project management, the previous system dynamics project models contribute to the development of DPM. However, DPM is distinguished from the previous project models, since they deal with project development under closed environment, i.e. product development or software development processes. As a result, they focus on rework cycle that has a significant impact on the performance of a project having the same repeated processes under limited constraints. In contrast, DPM focuses on iteration cycles caused by changes, which are more general in construction than those caused by rework. In addition,
as will be described in Chapter 6.2, the system dynamics models of DPM are structured for general use. As a result, model structures and variable equations are made so that all possible structures and variable relationships, some of which are unknown at the modeling time, can be represented.

6.1.2 DPM Functions

As a result of integrating all of the component methodologies, parameters in DPM include time and resource constraints, buffering, construction policies and human factors as well as activity duration and precedence relationships. Based on initial input data and control actions taken by DPM users, DPM creates a project plan, suggests project policies and simulates project performance profiles, as diagramed in Figure 19. In addition, by comparing the simulated performance with the actual performance, parameters in DPM can be calibrated for getting more reality and accuracy in projection of the project performance. All the simulation data and changes in the system are stored in its database for future utilization by an agent on the DPM system.

Suggestion of Project Plans and Policies

Given construction input data, a robust project plan and project policies are suggested based on the characteristics of construction activities and work conditions. In particular, desirable buffer size and location for activities are recommended so that the vulnerability of downstream work on upstream changes can be reduced. In addition, DPM provides the most effective construction policies that can rigorously deal with indirect and/or unanticipated events during the construction execution.

Performance Profiles Projection and Analysis

Performance profiles in terms of time, cost, quality, and safety can be projected from the data date until milestone completion. Depending on input variables
and planning purposes, the output may start with network-based methods and evolve into DPM as data for parameters are provided. Based on the projected profiles, the simulated construction performance is analyzed to identify problem areas, activity criticality, bottleneck of resources and construction sequences.

**Analysis of Effectiveness of Managerial Actions**

To examine the effectiveness of managerial actions in advance, performance consequences (trends and patterns) of decisions or policies are analyzed. The decisions and policies include delivery time and duration changes, labor control changes, buffering and other re-engineering efforts.

**Calibration of System**

Once actual data are obtained, the parameters in DPM that were used for planning and simulation can be tuned up for more accurate and reliable planning and projection, by comparing the simulated performance with the actual performance. This feedback process can help convert the past experience on a project to the knowledge that can be utilized for planning and management of the project next time.

Having the functions discussed thus far, DPM has advantages and improvements over the existing planning and management tools, some of which are summarized in Table 5.
Parameter Tables

Changing Network Logic

Breaking Down Activities

Feeding Reliability Buffers

Finding a Better Policy

Project Data Input

Duration

Dependency Structure Matrix

Data Transfer from Other Tools

P3 MS Project

Project Data Input

DPM Planner

Shortening Target Durations

Changing Network Logic

Breaking Down Activities

Feeding Reliability Buffers

Finding a Better Policy

DPM Simulation Engine

Model Validation

Influential Factors

Influence Curves

Model Calibration

Simulation Analysis

Actual Historic Data

Simulated Data

Cost

Time

Quality

Safety

Performance Profiles

RFI

Change Orders

Deficiency Level

Construction Progress

Performance Profiles

Application to Planning and Control

Feedback & Database

Planning & Simulation

Figure 19: System Architecture of DPM

72
### Table 5: Functions of DPM

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Network-Based</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 CPM (Critical Path Method)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 PERT (Program Evaluation and Review Technique)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 GERT (Graphical Evaluation and Review Technique)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 PDM (Precedence Diagramming Method)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Simulation-Based</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 CYCLONE (Cyclic Operations Network)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 MUD (Model for Uncertainty Determination)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 INSIGHT (Interactive Simulation using Graphics Techniques)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8 PLATFORM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9 PRODF (Project Duration Forecast)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 DYNASTRAT (DYNAMIC STRATEGY)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 CEV (Conditional Expected Value)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12 SLAM (Simulation Language for Alternative Modeling)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13 STROBOSCOPE (State and Resource Based Simulation)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14 NETCOR (NETWORKs under Correlated uncertainty)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Concurrent Engineering</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 DSM (Dependency Structure Matrix)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 Strategic Decision Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17 Overlapping Framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 Theory of Constraints</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 DPM (Dynamic Planning and Control Methodology)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Note

1 DuPont Inc and UNIVAC Division of Remington Rand, 1958
2 US Navy and Booz-Allen Hamilton and Lockheed Co., 1958
3 Pritsker, 1966
4 Craig, 1964
5 Halpin, 1977
6 Carr, 1979
7 Paulson, 1983
8 Levitt & Kurz, 1985
9 Ahuja & Nadakumar, 1985
10 Padilla & Carr, 1991
11 Pritsner & Russel, 1992
12 Pritsner, 1994
13 Martinez, 1996
14 Wang & Demsetz, 2000
15 Eppinger, 1994
16 Alarcin & Bastas, 1998
17 Pena-Mora & Li, 2000
18 Goldratt, 1977
19 Pena-Mora & Park, 2001
6.1.3 Collaboration Scheme

Many different types of commercialized software are currently being used in the planning and control of modern construction projects. Moreover, in some cases, even project functions working for the same project use different types of tools. Consequently, a newly developed methodology needs to be compatible with the existing tools, in order to increase its applicability. In addition, the various project functions involved in a construction project tend to be geographically distributed and in different work conditions [Pena-Mora and Dwivedi, 2000], which requires the collaboration capability of a planning and control tool. For these reasons, DPM is designed to share project data with other existing tools and to support various kinds of computing devices, which is detailed below (more discussions on the collaboration scheme of DPM can be found in Pena-Mora and Dwivedi, 2000).

Project Data Sharing

As diagramed in the system architecture on Figure 19, data input can be made through DPM input windows or by transferring data in Primavera P3 or Microsoft Project into DPM. Then, given the input data DPM can analyze and project the construction performance by simulating system dynamics models. Input data and simulation results are saved in the DPM database, which can be used for the calibration of the system dynamics models or further simulation. DPM also provides a generic interface with various analysis and planning tools, which makes it possible for project management personnel to access the analysis and planning tools on the same DPM platform. The component based system architecture of DPM makes it possible for an added application to function as another module in DPM irrespective of the operating system of the application [Pena-Mora and Dwivedi, 2000]. For instance, it is possible for project management personnel to execute DPM own functions by simulating the system dynamics models and to do network scheduling work by calling P3 scheduling engine on the DPM platform.
Web-Based Collaboration

The DPM system architecture allows project functions distributed and in different work conditions to collaborate by supporting various kinds of devices such as mobile phones, laptop computers, and palm pilots. Due to the nature of construction, construction crew do not always get access to a desktop computer in their site office, 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 [Pena-Mora and Dwivedi, 2000]. For example, DPM makes it possible for a project manager in the headquarter to collaborate with construction crew on the site and designers in their office.

Figure 20-a: Collaboration Scheme
Implementation

To implement the collaboration scheme discussed thus far, the system architecture conceptualized in Figure 20-b is adopted. DPM provides a web-based planning and control environment, in which a user can simulate DPM with Java applet in his/her side activated. After the user makes input, the applet requests simulations to the main server through Java Remote Method Invocation (RMI). Then, RMI simulates DPM and simulation results are saved in DPM database through Java Data Base Connectivity (JDBC).

Figure 20-b Scheme of Web-based Collaboration
For the web implementation of DPM, three main tools, Java programming language, Vensim, and RMI are extensively used. Java language makes DPM platform-independent. Furthermore, the utilization of Micro-Java supports hand-held devices without the limit by the operating system [Pena-Mora and Dwivedi, 2000]. In addition, Vensim, which is a powerful System Dynamics modeling tool provides a simulation engine and analytical tools. Lastly, Java RMI is used to increase distributed computing capabilities. RMI allows Java objects running on the same or separate computers to communicate with one another via remote method calls. Such method calls appear the same as those operating on object in the same program [Deitel and Deitel, 1999]. Combining three main tools make DPM to work through the web.

Meanwhile, Figure 20-c more specifically shows how distributed systems exchange data among them. User input can be transferred to DPM through Java RMI. Then, Vensim.class calls venjava.dll, and in turn venjava.dll loads the Vensim DLL file. Through these processes, the DPM models are simulated. Once simulated, DPM shows the results through Java applet and save them in Oracle database through JDBC.

![Figure 20-c: Scheme of Data Exchange in DPM Systems](image)
In addition, by allowing data importing from Primavera, which is one of the most widely used project management software, DPM further increases its applicability. As conceptualized in Figure 20-d, DPM accesses and controls the SQL Server of Primavera Enterprise through JDBC driver. Meanwhile, Java RMI is used to connect the SQL server, considering that the Primavera SQL database can be located in remote places.

![Diagram of data exchange with Primavera](image)

Figure 20-d: Data Exchange with Primavera
Figure 20-e: DPM Working on the Web
6.2 SYSTEM DYNAMICS MODEL DEVELOPMENT

All of the research component methodologies and fundamentals of DPM discussed thus far have been integrated into the system dynamics models to be described in this chapter. With consideration of the research goals, the system dynamics modeling work focused on capturing dynamic interactions among concurrent processes and examining the effectiveness of construction planning, monitoring, and controlling. In particular, the system dynamics models aim to provide activities with an appropriately sized and located buffer, given construction activity characteristics and construction control policies.

This chapter first discusses feedback processes involved in construction that may trigger non-value adding iterations, followed by the definition of some important model variables and the DPM model boundary. Then, the generic construction process model structure, which constitutes the skeleton of DPM is presented and important concepts imbedded in the generic process model structure are discussed. Finally, the DPM model is compared to the previous system dynamics models of project management and the potential improvement achieved by the DPM model is discussed.

6.2.1 Definitions

**Production Type:** The pattern of an activity work progress. In the case of *Fast Production*, productivity is initially high but decreases as construction progresses due to increased work complexity. In contrast, the productivity of *Slow Production* is initially low but increases as construction progresses due to learning effect.

**Reliability:** The degree of work quality and robustness against uncertainties. A *Reliable* activity produces less changes, while an *Unreliable* activity generates more changes.

**Sensitivity:** The degree of how much an activity is sensitive to changes made internally (*Internal Sensitivity*) or externally (*External Sensitivity*). A *Sensitive* activity is more vulnerable to changes than an *Insensitive* activity.
**Changes**: changes refer to work state, processes, or methods that deviate from the original plan or specification

**Unintended Changes**: Changes resulting from work quality, work conditions or scope changes, which can cause managerial changes, rework, or hidden changes, depending on managers’ willingness to adopt the change option and quality management thoroughness.

**Managerial changes**: Changes made by a managerial decision to avoid the direct impact of rework

**Hidden Changes**: Unintended changes that have been inspected and monitored but not found. Hidden changes are released to the downstream work together with work done correctly.

**Quality Management**: Actions taken to improve quality through monitoring or to control quality through inspection, including quality assurance by contractors and quality control by owners’ representatives

**Quality Management Thoroughness**: Thoroughness in doing quality management. In the model structures, it refers to the fraction of discovered changes in total changes that have occurred, while \((1 - \text{Quality Management Thoroughness})\) represents the fraction of hidden changes in total changes that have occurred.

---

### 6.2.2 Model Boundary

A clear definition of the model boundary is important in system dynamics modeling and needs to be done based on the modeling scope and purpose. In this context, the boundary of DPM model to be described in this chapter has been decided so that a single generic process model structure can be replicated to build a multiple-activity project to effectively capture interactions among activities. Variables in the DPM model are either *endogenous* or *exogenous*, as conceptualized in Figure 21.
The values of *exogenous* variables in the DPM model are to be set by users and do not change during the simulation. For example, users may provide activity data such as activity duration and characteristics through the DPM interface. In contrast, the values of *endogenous* variables change throughout the model simulation. For the simulation, the scope of an activity is determined in proportion to the given activity duration. Then, the model simulation is done, taking into consideration all the constraints involved in activity processes such as pre-checking, quality management, and change and rework cycle. As a result of the simulation, *endogenous* variables come to have their values. Examples of *endogenous* variables include activity and project progress, change and rework iterations, actual quality and productivity, and workforce allocation and utilization ratio. One thing to note associated with this model boundary is that the DPM model is structured so that DPM can generate the same activity duration and start time as the CPM-based tools, in case only CPM data is given to the DPM system. This results from the consideration of DPM's compatibility with existing CPM tools. However, DPM model behaviors that generate the results are dynamic. For example, actual work quality and productivity are dynamically simulated and as a result change their values throughout the simulation period, which cannot be represented in the CPM-based tools.

Meanwhile, variables that are classified as *excluded* in Figure 21 are tentatively excluded from the DPM model structure. However, once the applicability of DPM is confirmed, they need to be incorporated into the DPM model in order to increase the capability to replicate real construction processes. Examples of *excluded* variables include multiple project development sequence, the impact of scope changes, work conditions such as weather, seasonal effect and site conditions, constraints in cash flow, material and equipment, and performance profiles in terms of costs, safety, and environmental impact.
Figure 21: Model Boundary
6.2.3 Feedback Processes in Construction

Normally, construction involves feedback processes represented in the causal loop diagram on Figure 22-a, 22-b, 22-c, and 22-d. When tasks and resources are available, first, the upstream work, based on which the available tasks will be carried out, is reviewed before commissioning resources for the tasks. During the review process, problems made in the upstream work can be discovered. Once they are found, depending on managerial decisions, workers request the upstream worker to correct the problematic work. More upstream hidden changes can cause more requests for the upstream work reprocess, which results in more pending tasks (A) and schedule delays (B) in the downstream work. Otherwise, workers construct tasks not having problems in the associated upstream tasks, with given resources. Once tasks are completed, the construction performance on the tasks is periodically monitored or inspected to see whether or not the target quality is met and the intended functions are achieved. Through this quality management process, completed tasks can allow the downstream work to proceed or work needs to be done on the completed tasks.

Unintended changes resulting from low work quality, bad work conditions or frequent scope changes can cause either managerial changes (C), rework (D), or hidden changes (E), depending on managers’ willingness to adopt the change option and quality management thoroughness. The more construction is delayed the more often the change option tends to be adopted (F), in order to avoid rework, which is normally perceived to have a bigger impact on the schedule performance. However, such managerial efforts can create unplanned and/or indirect side effects. As a result of feedbacks involved in the processes (F, G, H, I, J), managerial changes can trigger further delays as well as rework. As diagramed in Figure 22-a, managerial changes trigger reprocess iterations of tasks that have been already released (refer to the definition of managerial changes in Table 2), while rework delays the construction progress by creating reprocess iterations of tasks that have not been released.
Figure 22-a: Change Option-Further Delay Loop
Figure 22-b: Quality Management Thoroughness-Further Delay Loop
Figure 22-c: Downstream Reprocess Iteration-Further Delay Loop
In addition, delays also may make quality management efforts less thorough \((K)\), which results in more hidden changes \((L)\). During the downstream review process, hidden changes released from the upstream work can be discovered. Once they are found, depending on managerial decisions, downstream workers request the upstream worker to correct the hidden changes. As a result, more hidden changes can cause more correction requests from the downstream \((M)\), which also can delay the construction progress as a result of subsequent feedback processes \((N, I, J)\) diagramed in Figure 22-b.

Furthermore, increased willingness to adopt managerial changes also can increase subsequent changes in the downstream work \((O)\), which delays the downstream work process. Consequently, reprocess requests from the downstream work are also delayed \((R)\), which again impacts the schedule performance of the activity that has originated changes \((N, I, J)\). Meanwhile, lowered quality management thoroughness creates more hidden changes \((L)\). Increased hidden changes can deteriorate the work quality of the downstream work, which creates more reprocess iterations of the downstream tasks. This also impacts the upstream schedule performance through \((R, N, I, J)\).

All of these feedback processes can constrain the construction performance, combined with resource availability, construction policies, and people' reactions to work conditions. For this reason, a good understanding on the feedback processes is crucial to the robust planning and control of a construction project under uncertainties.
Figure 22-d: Feedback Processes in Construction Activities
6.2.4 Model Description

The generic construction process model structure to be described in this chapter represents the generic process of a construction project, capturing dynamic interactions among activities during concurrent construction. In particular, the process model structure aims to capture construction iterations caused by changes, and to assess their impact on the construction performance. Having an ability to capture and quantify change impacts on the construction system, DPM examines the effectiveness of construction planning, monitoring and controlling, and policies.

Based on feedback processes and relationships among construction variables in the causal loop diagrams on Figure 22, the quantitative representation of generic construction processes has been modeled. In the model structure in Figure 23, workflow during construction is represented as tasks flowing into and through five main stocks, which are named WorkToDo (WtDo), WorkAwaitingRFIReply (WaRRep), WorkAwaitingQualityManagement (WaQM), WorkPendingduringUpchangeRP (WpURP) and WorkReleased (Wrel). Available tasks at a given time are introduced into the stock of WorkToDo through the InitialWorkIntroduceRate (iWiR). The introduced tasks are completed through the WorkRate (WR), unless changes in the upstream work, based on which the downstream work will be carried out are found. The completed tasks, then, accumulate in the stock, WorkAwaitingQualityManagement where they are waiting to be monitored or inspected. Depending on work quality, some completed tasks are either returned to the stock of WorkToDo through RPAddressRate (RPaR) or released to the downstream work through WorkReleaseRate (WrR). In addition, it is also possible for released tasks to return to the stock of WorkToDo again through RPAddressAfterReleaserate (RPaaR) for various reasons. Meanwhile, In addition, in case upstream problematic work is found during the pre-checking period, corresponding tasks flow from and to WorkToDo through RequestForInformationRate (RFIR), UpChangeAccomodateRate (UCaR), RPRquesttoUpRate (RPrUR) and
PendingWorkReleaseRate (pWR). These five stocks can be described using the differential equations listed below. For the simulation of a flexible number of construction activities, the equations in the dynamic project model are represented using two-dimensional subscripts, activity and preceding, which respectively point to the activity itself and those having concurrent relationships with the activity. Detailed explanations on equations are followed by model descriptions.

\[
\begin{align*}
(d/dt)(WtDo[i]) &= iWiR[i] + \sum_{j=1}^{n}(pWR[i,j]) + \sum_{j=1}^{n}(UCaR[i,j]) + RPaR[i] + \\
& \quad - \sum_{j=1}^{n}(RFIR[i,j]) - WrR[i] \quad [1] \\
(d/dt)(WaRRep[i,j]) &= RFIR[i,j] - UCaR[i,j] \quad [2] \\
(d/dt)(WpURP[i,j]) &= RPrUR[i,j] - pWR[i,j] \quad [3] \\
(d/dt)(WaQM[i]) &= WR[i] - WrR[i] - RPaR[i] \quad [4] \\
(d/dt)(Wrel[i]) &= WrR[i] - RPaaR[i] \quad [5] \\
\end{align*}
\]

, where \(i = \text{activity}, j = \text{preceding}, \) and \(i, j \in \{1, 2, 3, \ldots, n\}\).

Pre-Checking before Construction

When upstream changes are found during the base work, downstream workers normally ‘request for information’ (RFI) to upstream workers or project managers. If by means of RFIs, the upstream changes turn out to have occurred by mistake (unintended changes) and a managerial decision is made to correct the changes in the location of the change generation, corresponding downstream tasks are delayed until the upstream changes is reprocessed. For example, assume that before starting the floor tile work, it is found that the floor slab was constructed thicker than its specification due to inaccurate concrete pouring in the upstream. As a result, if the tile work proceeds with the current concrete slab unchanged, the facility may not have the required ceiling height. In this case, the project manager may ask the upstream concrete crew to correct the slab thickness by chipping the excess concrete. In the model structure, this process is represented as the following processes (L1). Downstream tasks corresponding to
Figure 23: Generic Construction Process Model Structure
upstream changes are moved into WorkAwaitingRFIreply, and then WorkPendingduetoUpChangeRP where they wait for the upstream changes to be reprocessed. When the upstream changes are reprocessed, pending downstream tasks are returned into the stock of WorktoDo for the base work.

However, the iteration of WorkAwaitingRFIReply-WorkPendingduetoUPChangeRP-WorktoDo does not take place in the following cases. First, when upstream changes have been released to the downstream by managerial decisions (managerial changes), they are supposed to be accommodated by changing associated downstream tasks. Continuing with the slab concrete example, it is possible to find the inaccurate concrete construction just after pouring concrete in the upstream. However, after comparing the economic impact of each option (change or rework) on the construction performance, the project manager may decide to change the specification of downstream tasks such as the thickness of mortar or the method of waterproofing instead of ordering rework on the slab. In this case, since a change option has been already adopted in the upstream work, corresponding downstream tasks are supposed to be changed after the management decision is confirmed through RFIs. Secondly, unintended upstream work changes also can be accommodated in the downstream work, when they are found during the downstream work and a change option on the changes is adopted. Going back to the concrete slab example, it is possible to find the inaccurate concrete construction during the tile work and to adopt a change option based on managerial decisions. In the model structure, both cases are represented as associated tasks in WorkAwaitingRFIreply being returned to WorktoDo through UPChangeAccomodateRate (L2).

The equations for L1 and L2 are based on how many hidden changes occurred during the upstream work are discovered and what fraction of those discovered changes are returned to the upstream work for correction. During the correction period, corresponding downstream tasks are delayed. Since the release time of pending downstream tasks vary depending on the completion time of change correction in each upstream work, it is needed to know in which upstream work hidden changes occurred so that the model
structure can locate where to request for change correction and when to release pending downstream tasks after correction work in each location of change generation is done. For these reasons, the equations of variables associated with L1 and L2 have two-dimensional subscripts, which are activity and preceding.

First, WorkRate (WR) and RequestForInformationRate (RFIR) are in proportion to work rate, which is the lesser of the available work divided by the minimum work time (MinWorkTime, minWt), and the available workforce (PotentialWorkRatefromResource, PwRR). Given a certain amount of work rate, whether tasks flow through WorkRate or RequestForInformationRate is determined by FractionofRFI (fRFI), which refers to the fraction of work requiring RFIs before execution among total available work. Multiplying total potential work rate with FractionofRFI yields RequestForInformationRate, while the amount of WorkRate can be obtained by multiplying total potential work rate with (1-FractionofRFI). FractionofRFI is determined by the function of FractionOfHiddenChangeinWorkReleased (fHCWrel) of upstream activities, and the downstream activity’s ExternalSensitivity (ES) to upstream hidden changes and QualityManagementThoroughness (QMth).

![Diagram: Fraction of RFI](image)

**Figure 24: Fraction of RFI**
More upstream hidden changes and sensitive downstream work create more RFIs. In addition, the more thorough quality management is, the more often RFIs are made. Meanwhile, the value of *PrecedenceRelationships* (PR) is used to identify an activity’s upstream activities and which precedence relationships (Finish-to-Start, Finish-to-Finish, Start-to-Start, Start-to-Finish) are involved in between activities. One thing to note associated with this split-the-flow is that *WorkRate* has one dimensional single subscript, *activity*, while *RequestForInformationRate* has two dimensional subscripts, *activity and preceding*, since it is attributed to the characteristics of upstream downstream activities as well as those of the activity itself. For this reason, *FractionofTotalRFI* (fTRFI), which represents the sum of upstream FractionofRFIs, is used in determining the amount of *WorkRate*. As a result, *WorkRate*, *RequestForInformationRate* and *FractionofRFI* can be represented by the following equations:

\[
WR[i] = (1-fRFI[i]) \times \min\left(\frac{WtDo[i]}{minWt[i]}, PwRR[i]\right) \quad [6]
\]
\[
RFIR[i,j] = fRFI[i,j] \times \min\left(\frac{WtDo[i]}{minWt[i]}, PwRR[i]\right) \quad [7]
\]
\[
fRFI[i,j] = QMth[i] \times \text{if } (PR[i,j] = \text{true}) \text{ then } (fHCWrel [j]* ES[i,j]) \text{ else } (0) \quad [8]
\]

, where \(i = \text{activity}, j = \text{preceding}, \) and \(i, j \in \{1,2,3..., n\}\).

**Quality Management**

Once gathered in *WorkAwaitingRFIreply* (WaRRep), tasks in the stock are either returned to *WorktoDo* through *UPChangeAccomodateRate* (UCaR) or moved to *WorkPendingduringUpchangeRP* (WpUIRP) through *RPRquesttoUpRate* (RPrUR). During this process, *AvgRFIreplyTime* (avgRFIt) and *ManagerialChangeRatio* (mCR) govern this split-the-flow. The average RFI reply time, would be few days or months, depending on the management’s RFI handling time and the amount of pending RFIs. Meanwhile, *ManagerialChangeRatio* refers to the normal ratio of adopting the change option. Dividing the amount of tasks in *WorkAwaitingRFIreply* by *AvgRFIreplyTime* yields the average outflow of the stock and once the average outflow is multiplied by *ManagerialChangRate*, it is possible to get the inflow into *WorktoDo* stock. Similarly,
the inflow into WorkPending during UpchangeRP can be calculated by multiplying \((1 - Managerial Change Rate)\).

\[
UCaR_{ij} = mCR[i] \times WpURP_{ij} / \text{avgRFIt} \tag{9}
\]

\[
RPRUR_{ij} = (1 - mCR[i]) \times WpURP_{ij} / \text{avgRFIt} \tag{10}
\]

where \(i = \text{activity}, j = \text{preceding}, \) and \(i, j \in \{1, 2, 3, \ldots, n\}\).

Meanwhile, the iteration loops of L1 and L2 have nontrivial impacts on the construction performance. In particular, when construction is performed concurrently, the impact of those loops becomes greater. The design and construction overlapping makes the construction work usually proceed with incomplete design drawings. Consequently, there are a lot of RFIs during construction, which can disrupt the construction sequences. Even among design activities many non-value adding iterations that can be represented by L1 or L2 occur due to insufficient volume and poor information on tasks. In fact, it is observed in the research case project, Route 3 North Project that non-value adding iterations very often occurred during the design work, mainly due to frequent scope changes and delays in the owner's decision making, which significantly delayed the whole construction processes.

Completed construction tasks are internally monitored and/or inspected by the owner's representatives. Depending on the result of quality management, completed tasks are either released to the downstream or reprocessed. The following task flows in the model structure represent the quality management process in construction. Tasks accumulated in WorkAwaitingQualityManagement are periodically monitored and inspected. In principle, tasks satisfying the target quality level and having intended functions are approved and moved to WorkReleased (L3), while changes are disapproved and pass into the stock of WorkToDo (L4) where they wait to be reprocessed (rework option). This process is governed by ActualWorkQuality (actWQ), which is a function of the reliability of an activity, upstream quality impact, schedule pressure and workers' fatigue. An unreliable activity work generates more changes than a reliable
activity work. In addition, the low quality of the upstream work can also lower the reliability of an activity work. More precisely, upstream hidden changes that have not been discovered during the downstream pre-checking (Fraction of UP Hiden Change Not Addressed in Figure 24) impact the downstream work quality. Lasting schedule pressure also can lower work quality, since workers often attempt to achieve the target schedule by cutting the corner. Lastly, when overtime continues after a certain threshold, workers will become fatigued, which possibly lowers work quality. One thing to note in association with quality management is that tasks that have not been monitored or inspected always contain some portion of undiscovered changes due to time delays involved in quality management. These undiscovered changes could impact the downstream work quality, if quality management period is long or completed tasks are released to the downstream before being checked through quality management.

During quality management, it is possible to release changes to the downstream by failing to notice them. In the model structure, the degree of overlooking changes is determined by Quality Management Thoroughness (QMth), which is normally low, when an activity work is complex or schedule pressure lasts throughout the activity work period. Overlooked changes, which are defined as hidden changes, are released to the downstream. If the downstream workers also fail to notice the hidden changes, they can deteriorate the downstream work quality, depending on the downstream sensitivity to upstream changes. The hidden change co-flow model in Figure 25 specifically describes the flow of hidden changes in order to measure the fraction of hidden changes in the total tasks released thus far, which determines the degree of upstream work impact on the downstream work quality.

In addition, it is also possible for some of discovered changes to be released to the downstream work by a managerial decision. In the model structure, this process (change option) is governed by Managerial Change Ratio (mCR). As discussed before, the project manager may release changes to the downstream work as they are, when they are not
expected to cause significant changes in preceding or succeeding tasks, and changing the
scope of other associated tasks is considered a more plausible way in terms of time and
costs than correcting discovered changes themselves. For example, assume that during
the pile work a steel pile sunk under the soil, as it could not reach a rock layer to support
the pile. In this case, the pile worker may drive another pile on the top of the previous one
instead of pulling out and then re-driving it. If failures continue, the pile worker may try
to drive piles somewhere else adjacent to the original pile location, given the approval
from the project manager, the structural engineer, and geotechnical engineer.

The equations for $L3$ and $L4$ are based on how many unintended changes occurred
during the work and what fraction of changes is discovered. In addition, the value of
variables associated with $L3$ and $L4$ is also related to the fraction of managerial changes
among discovered unintended changes. In order to calculate the amount of $WorkReleaseRate$ (WrR), first, it is needed to know how many unintended changes occur
during the work, which is determined in the model by $ActualWorkQuality$ (avgWQ). Assuming that 100 square meters of masonry work is done and $ActualWorkQuality$ is
90%, in principle, only 90 square meters of masonry work would be approved and
released to the downstream work (in the model, moving corresponding tasks from
$WorkAwaitingQualityManagement$ to $WorkReleased$). However, depending on the
thoroughness of quality management ($QualityManagementThoroughness$, QMth), the
actual amount of work release rate can be more that 90 % of the total work done. For
example, assuming that quality management thoroughness is 50%, in the above example,
5 square meters of masonry work can be additionally released by overlooking problems
on that part of work. Furthermore, as discussed before some of discovered changes can be
converted into managerial changes, which is governed in the model by $ManagerialChangeRatio$ (mCR). Supposing that $ManagerialChangeRatio$ is 50%, 2.5
square meters of masonry work out of discovered changes (5 square meters) can be added
to the actual amount of work released by a managerial decision. Consequently, 97.5 5 square meters of masonry work will be released to the downstream, while only 2.5 5 square meters of masonry work are subject to rework. Equations for $WorkReleaseRate$
and $RPAAddressRate$ are:
Figure 25: Hidden Change Co-Flow
\[ \text{WrR}[i] = \text{avgWQ}[i] + (1-\text{avgWQ}[i]) \times (1-QMth) + (1-\text{avg}[i] \times QMth \times mCR) \quad [11] \]
\[ \text{RPaR}[i] = (1-\text{avgWQ}[i]) - (1-\text{avgWQ}[i]) \times (1-QMth) - (1-\text{avgWQ}[\text{activity}]) \times QMth \times mCR \quad [12] \]

, where \( i = \text{activity}, j = \text{preceding}, \) and \( i, j \in \{1,2,3, \ldots, n \} \).

Reprocess Iterations of Work Released

Meanwhile, as discussed previously, it is possible to reprocess work that have been released for various reasons. In the model, this is represented as tasks flowing from WorkReleased to WorktoDo through RPAddressAfterReleaserate (L5). There are three variables that constitute RPAddressAfterReleaserate (RPaaR). That is, RPRequestfromDownstream (RPrfmD), RPTriggeredbyInternalManagerialChange (RPtIMC), and RPTriggeredbyExternalManagerialChange (RPtEMC).

Descriptions on the equations associated with L5 begin with RPRequestfromDownstream. RPRequestfromDownstream is initiated by hidden upstream changes. Once upstream hidden changes are found during the downstream work and it is decided to correct them in the upstream work, they are returned to the upstream work. In the model, two variables are associated with this iteration process. Corresponding tasks are returned from the downstream work through RPRequesttoUPRate, which determines the amount of RPRequestfromDownstream in the upstream work. As a result, RPRequestfromDownstream refers to the amount of hidden changes addressed by the downstream work.

In reality, as conceptualized in Figure 26, change correction requests to one upstream activity can be made in multiple downstream activities. Having multiple change correction requests from downstream activities, RPRequestfromDownstream in the upstream activity takes the biggest one among them. This is because facing the same amount of upstream hidden changes, the amount of change correction requests from downstream activities can be different, depending on downstream activities’ sensitivity,
quality management thoroughness, and the willingness to adopt the change option. For example, suppose that Upstream B in Figure 26 generate 10 hidden changes, which are released to Downstream A and B. Assuming that the sensitivities of Downstream A and B to the hidden changes are 50% and 100% respectively, only 5 hidden changes are influential to the work quality of Downstream A, while all of 10 hidden changes can impact Downstream B. As a result, a change correction request will be made on 5 hidden changes in Downstream A and on 10 hidden changes in Downstream B. Meanwhile, having different amount of change correction requests from Downstream A and B, Upstream B needs to do correction work on 10 changes, since the 10 changes caused the different correction requests. Explanations on quality management thoroughness and the willingness to adopt the change option also can be made in the same manner. In addition, in case the work scopes (ActivityScope, ASc) of associated activities are different, an adjustment based on their work scopes is needed in determining the final amount of RRequestfromDownstream. Therefore, RRequestfromDownstream can be represented by the following equation:

Figure 26: Hidden Change Iteration Paths
RPrfmD[j] = max (sum \( i=1 \ldots n \) (RPrUR\([i,j]\) * ASc\([j]\) / ASc\([i]\))) \[13\], where \( i = \text{activity}, j = \text{preceding}, \) and \( i, j \in \{1,2,3\ldots,n\}\).

While discovered changes are being reprocessed in the upstream activity, the downstream activity is delayed. Once correction work is done, the upstream activity returns the average reprocess time \((\text{AvgHiddenChangeRPTimeinUP}, \text{avgRPt})\), with which downstream activities can release pending tasks to \text{WorktoDo} through \text{PendingWorkReleaseRate}. Continuing with the previous example, the same average correction time of Upstream B is returned to both Downstream A and B according to the iteration logics discussed above. As a result, Downstream A and B apply the same time to release their tasks pending due to changes made in Upstream B. In addition, in the case of Downstream A in Figure 26, different average reprocess times, each being returned from Upstream A and B are applied to releasing their pending tasks.

Meanwhile, the impact of managerial changes on preceding tasks within the same activity is represented by \text{RPTriggeredByInternalManagerialChange} (RPtIMC). \text{PTriggeredByInternalManagerialChange} consists of two flows. That is, \text{ManagerialChangeReleaseRate} (MCRR) and \text{UPChangeAccomodateRate} (UCaR), which represent the amount of managerial changes generated either before or after work execution. In addition, the impact of managerial changes is also related to an activity's sensitivity to internally made changes (\text{InternalSensitivity}, IS) and the fraction of work released so far (\text{FractionofWorkReleased}, fWRel). This is because the more sensitive an activity is and the more work an activity has done thus far, the more managerial changes can impact. Therefore, by multiplying sum of the two flows with \text{InternalSensitivity} and \text{FractionofWorkReleased}, it is possible to calculate the amount of tasks flowing through \text{RPTriggeredByInternalIntendedChange} as follows:

\[ \text{RPtIMC}[i] = \text{MCRR}[i] + \text{UCaR}[i]) * \text{IS}[i] * f\text{WRel}[i] \] \[14\], where \( i = \text{activity} \) and \( i \in \{1,2,3\ldots,n\}\).
Finally, \( RP_{\text{Triggered by External Managerial Change}} \) (RPtEMC) determines the impact of external managerial changes. Any managerial changes made in activities having precedence relationships or reprocess iteration relationships (\( \text{TotalExternalManagerialChange} \), \( \text{totEMC} \)) can be potential impact sources. \( RP_{\text{Triggered by UP Managerial Change}} \) (RPtUPMC) represents the impact of managerial changes made in upstream activities, while \( RP_{\text{Triggered by DN Managerial Change}} \) (RPtDNMC) is the impact of downstream-managerial changes. In addition, an activity’s sensitivity to those changes (\( \text{ExternalSensitivity} \), ES) and the fraction of work released so far (\( \text{Fraction of Work Released} \), fWRel) are also related to determining the impact of externally made managerial changes. In case the work scopes (\( \text{Activity Scope} \), ASc) of activities having concurrent relationships are different from the activity itself, an adjustment based on activities’ work scopes is needed in determining the final value of \( RP_{\text{Triggered by External Managerial Change}} \). Therefore, it is possible to calculate the amount of tasks flowing through \( RP_{\text{Triggered by External Managerial Change}} \) with the following equations:

\[
\begin{align*}
\text{RPtUPMC}[i] &= \sum_{j=1}^{n} \text{if } (PR[i,j] = \text{true}) \text{ then } (\text{totEMC}[j] \ast \text{ASc}[i] / \text{ASc}[j] \ast \text{ES}[i,j] \ast \text{fWRel}[i]) \text{ else } (0)) \quad [14] \\
\text{RPtDNMC}[j] &= \sum_{i=1}^{n} \text{if } (RIR[j,i] = \text{true}) \text{ then } (\text{totEMC}[i] \ast \text{ASc}[j] / \text{ASc}[i] \ast \text{ES}[j,i] \ast \text{fWRel}[j]) \text{ else } (0)) \quad [15] \\
\text{RPtEMC}[i] &= \text{RPtUPMC}[i] + \text{RPtDNMC}[i] \quad [16]
\end{align*}
\]

where \( i = \text{activity} \), \( j = \text{preceding} \), and \( i, j \in \{1,2,3,\ldots, n\} \).

In addition to the generic process model discussed thus far, other supporting model structures for the project scope, resource acquisition and allocation, the project performance, and construction policies, which are diagramed in Figure 27 assist in examining the effectiveness of construction planning, monitoring and control of a construction project, and suggesting a robust construction plan. Descriptions on other model structures can be found in Appendix II.
Figure 27: Schema of Dynamic Project Model
6.2.5 Potential Impact of DPM Models

The DPM model structures are evolved from the previous system dynamics models in project management. For this reason, the DPM model structures need to be understood in the context of the previous work. The overall feature of the generic process model in DPM is unique by introducing change cycle, while some of the supporting model structures have their conceptual base on the previous system dynamics models, as described in Table 6. Meanwhile, new concepts and model structures have been introduced to system dynamics model based project management for the development of DPM, which are summarized in Table 7 in the context of major improvement in the previous research efforts.

This chapter illustrated how the concepts and logics of DPM have been materialized by incorporating the component methodologies into system dynamics models, together with the provision of DPM functions, collaboration scheme and system architecture. In addition, some important modeling concepts and the generic model structure that constitutes the skeleton of DPM were presented. Lastly, the potential contribution of the DPM models to system dynamics in project management was also discussed. In Chapter 7, the performance and applicability of DPM will be examined by presenting some application examples of DPM.
Table 6: Comparison with Previous Project Management Models

<table>
<thead>
<tr>
<th>Modeling Issues</th>
<th>Evolved from</th>
<th>Changes</th>
<th>Rationale for Model Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model Sector</td>
<td>Issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Overall</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Richardson and Pugh, 1981</td>
<td><strong>Augment/Newly Introduce</strong></td>
<td>Distinguishing from Rework</td>
</tr>
<tr>
<td></td>
<td>Abdel-Hamid, 1984</td>
<td>Change Cycles</td>
<td>Representing Change Order Process</td>
</tr>
<tr>
<td></td>
<td>Cooper, 1994</td>
<td>RFI Iterations during Pre-checking</td>
<td>Implementing Use-Defined Modeling</td>
</tr>
<tr>
<td></td>
<td>Ford and Sterman, 1997</td>
<td>Locating Change/Rework Source</td>
<td>Implementing Use-Defined Modeling</td>
</tr>
<tr>
<td></td>
<td>Lyneis, 1999</td>
<td>Having Multiple Ext. Dependencies</td>
<td>Having More Reality in Representing Construction Process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Progress-Based Monitoring and Quality Management</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Remove</strong></td>
<td>Coordination between activities is rare in Construction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Coordination Loop</td>
<td>Usually, RFI is answered by Project Manager</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rework Preparation Loop</td>
<td>Practically Difficult to Trace Different Work Efforts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Task Releasing Policy</td>
<td>In Particular, Ineffective When Activities are Aggregated.</td>
</tr>
<tr>
<td>Undiscovered Errors</td>
<td></td>
<td><strong>Remove</strong></td>
<td>Not Appropriate in Construction</td>
</tr>
<tr>
<td></td>
<td>Stock for Undiscovered Errors</td>
<td>Construction and Design have a Relatively Short Quality Management Cycle.</td>
<td></td>
</tr>
<tr>
<td>Interaction among Phases</td>
<td>Reichelt, 1990</td>
<td><strong>Augment/Newly Introduce</strong></td>
<td>Construction Is a Physical Manifestation.</td>
</tr>
<tr>
<td></td>
<td>Homer et al., 1993</td>
<td>Iterations Triggered by Downstream</td>
<td>Representing Feedback Caused by Downstream</td>
</tr>
<tr>
<td>Modeling Issues</td>
<td>Evolved from</td>
<td>Changes</td>
<td>Rationale for Model Changes</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>---------</td>
<td>----------------------------</td>
</tr>
<tr>
<td><strong>Scope</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Availability</td>
<td>Homer et al., 1993</td>
<td><strong>Augment/Newly Introduce</strong></td>
<td>More Effectively Representing Work Dependencies Involved in Construction</td>
</tr>
<tr>
<td></td>
<td>Ford and Sterman, 1997</td>
<td><strong>Integrating with Precedence Relationships</strong></td>
<td></td>
</tr>
<tr>
<td>Design Changes</td>
<td>Cooper, 1980</td>
<td><strong>Remove</strong></td>
<td>Not Distinguishing Design from Construction for DPM to Be a Planning and Control Tool for General Use.</td>
</tr>
<tr>
<td></td>
<td>Reichelt, 1990</td>
<td><strong>Adding Design Change to Activity Scope</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td>Homer et. Al, 1993</td>
<td><strong>Augment/Newly Introduce</strong></td>
<td>Different Productivity Patterns Involved in Construction, Which Are Changing throughout the Construction Period.</td>
</tr>
<tr>
<td>Productivity</td>
<td>Cooper, 1994</td>
<td><strong>Progress-Based Productivity Determined</strong></td>
<td></td>
</tr>
<tr>
<td>Quality</td>
<td>Hines, 1999</td>
<td>by a Function of Learning and Complexity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lyneis, 1999</td>
<td><strong>Remove</strong></td>
<td>Not Significant Compared to Progress-Based Productivity and Quality.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Experience and Quality Dilution</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Gaining OTJ Experience</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Resource</strong></td>
<td>Lyneis, 1999</td>
<td><strong>Remove</strong></td>
<td>Tentatively out of Model Scope</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Budget Constraints in Labor Control</strong></td>
<td></td>
</tr>
<tr>
<td>Authors</td>
<td>Year</td>
<td>Project Type</td>
<td>Problems addressed</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------</td>
<td>--------------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Roberts</td>
<td>1964</td>
<td>R&amp;D</td>
<td>Dynamics of R&amp;D: perceptions vs reality (1)*</td>
</tr>
<tr>
<td>Kelly</td>
<td>1970</td>
<td>R&amp;D</td>
<td>Dynamics of R&amp;D among concurrent projects</td>
</tr>
</tbody>
</table>
| Cooper              | 1980     | Design & Construction | (1)*  \
|                     |          |                    | Rework cycle: rework generation and discovery (2)*  \
|                     |          |                    | Staff hiring and allocation (3)*  \
|                     |          |                    | Interdependencies between work phases (4)*                                         |
| Richardson & Pugh   | 1981     | R&D                | (1)*  \
|                     |          |                    | Productivity & Rework generation (2)*  \
|                     |          |                    | Policy of hiring staffs (3)*                                                        |
| Jessen              | 1988     | R&D, Construction  | (1), (2), (3)*  \
|                     |          |                    | Project team motivation vs productivity                                            |
| Keloharju & Wolstenholme | 1989   | R&D                | (1), (2), (3)*  \
|                     |          |                    | Cost-time trade off                                                               |
| Abel-Hamid          | 1988, 1989, 1992 | R&D                   | (1), (2)*  \
| Abel-Hamid et al.   | 1993     |                    | Project Staffing Policies (3)  \
| Abel-Hamid & Madnick| 1991     | Software development | (1), (2), (3)*  \
|                     |          |                    | Cost & schedule estimations  \
|                     |          |                    | Quality assurance policies                                                        |
**Authors** | **Year** | **Project Type** | **Problems addressed**
---|---|---|---
Barlas & Bayraktutar | 1992 | Software development | (2)*
| | | | An interactive simulation game (5)*
Cooper | 1993, 1994 | Programs | Rework cycle: time to discover rework (2)*
Cooper & Mullen | 1993 | Defence & commercial software development | (1)*
Ford & Sterman | 1997 | Product development (chip development) | (1), (2), (3)*
| | | | Non-linear external and internal concurrency (4)
| | | | Development process prototyping (6)*
| | | | Hidden error co-flow
Park & Pena-Mora | 1999 | Design & Construction | (1), (2), (3), (4)*
| | | | Capital Game for fast-tracking construction (5)*
Park & Pena-Mora | 2001 | Design & Construction | (1), (2), (3), (4), (5), (6)*
| | | | Change Cycle: Distinction between change and rework
| | | | Managerial change vs Rework to deal with unintended change (7)*
| | | | Incorporating traditional network-based tools (CPM, PDM, PERT),
| | | | GERT, and SLAM into system dynamics models (8)*
| | | | User-defined modeling: allowing users to decide
| | | | the number of phases (9)*
| | | | Project Planning: Buffering (10)*
| | | | Web-based collaboration (11)*

* **Note: Major Improvement Areas**
  (1)* Project Monitoring & Control
  (2)* Rework cycle
  (3)* Human resource management
  (4)* Considering concurrent dependencies
  (5)* Interactive simulation
  (6)* Process prototyping
  (7)* Change Cycle
  (8)* Incorporating traditional tools
  (9)* User-defined modeling approach
  (10)* Project Planning: buffering
  (11)* Web-based Collaboration

109
CHAPTER 7

APPLICATIONS OF DPM

This chapter scrutinizes the performance of DPM as a planning and control tool, and its applicability in real world settings with three application examples. First, the performance and applicability of DPM is examined with the building construction example, which has demonstrated the possibility of the user-defined simulation modeling approach to the planning and control of construction projects. The discussion with this example focuses on DPM’s capability to analyze the cost implication of concurrent construction. Secondly, the change impact on the construction performance is analyzed by utilizing DPM model structure. Then, the effectiveness of reliability buffering that has been suggested in this thesis is examined based on the understanding the change impact, which is also validated by observing the DPM system behaviors. Lastly, the application example of reliability buffering to bridge construction projects demonstrates the applicability of DPM in real world settings, focusing on the role of reliability buffering in concurrent construction.
7.1 COST IMPACT OF FAST-TRACKING

This chapter presents the result of a case study that has been done to examine the effectiveness of fast-tracking. For the case study, DPM has been applied to an office building construction project, which is briefly described in Table 8. In reality, this case project was carried out by fast-tracking, aiming to reduce time to market in Warsaw, Poland. But it experienced a lot of problems including some of issues addressed in Chapter 2 and as a result, the project could not be completed as planned. Some results of the case study are described below (More detailed descriptions on the case study can be found in Pena-Mora and Park, 2001).

Table 8: Description of Building Construction Project

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Warsaw Daewoo Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Warsaw, Poland</td>
</tr>
<tr>
<td>Owner/Developer</td>
<td>Daewoo Corporation, Korea</td>
</tr>
<tr>
<td>Total Building Area/Floor</td>
<td>30,000 M²/40</td>
</tr>
<tr>
<td>Delivery Method</td>
<td>Fast-Tracking/Construction Management</td>
</tr>
<tr>
<td>Estimated Project Duration</td>
<td>156 weeks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project Activities</th>
<th>Schedule (weeks)</th>
<th>Turnover (M²/week/worker)</th>
<th>Hard Costs (US MIL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schematic Design</td>
<td>8</td>
<td>9.3</td>
<td></td>
</tr>
<tr>
<td>Design Development</td>
<td>12</td>
<td>6.25</td>
<td></td>
</tr>
<tr>
<td>Construction Document</td>
<td>18</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>Excavation</td>
<td>32</td>
<td>4.5</td>
<td>8</td>
</tr>
<tr>
<td>Foundation</td>
<td>24</td>
<td>3.5</td>
<td>7</td>
</tr>
<tr>
<td>Building Core Work</td>
<td>56</td>
<td>0.6</td>
<td>25</td>
</tr>
<tr>
<td>Structural Steel Work</td>
<td>56</td>
<td>0.7</td>
<td>40</td>
</tr>
<tr>
<td>Partitioning</td>
<td>56</td>
<td>0.7</td>
<td>5</td>
</tr>
<tr>
<td>Enveloping</td>
<td>44</td>
<td>0.9</td>
<td>30</td>
</tr>
<tr>
<td>MEP</td>
<td>56</td>
<td>0.7</td>
<td>40</td>
</tr>
<tr>
<td>Finishing</td>
<td>60</td>
<td>0.9</td>
<td>10</td>
</tr>
<tr>
<td>External Work</td>
<td>24</td>
<td>4.25</td>
<td>1</td>
</tr>
</tbody>
</table>

7.1.1 A Criterion for Fast-Tracking Strategies

Once DPM simulation is done with different fast-tracking scenarios, the cost-benefit tradeoff analysis presented in this chapter will be used to determine the effectiveness of fast-tracking, given each scenario. In fact, the effectiveness of fast-
tracking can not be measured solely based on economic principles because there can be many intangible benefits. For instance, there are times when market conditions will change radically and the competitive business environment requires owners to do their utmost to beat their market competitors with an earlier completion of their projects. In these cases, the market value of the shortened time is beyond the tradeoff between economic gain through the reduced project duration and the increased cost to reduce the duration.

![Diagram](https://via.placeholder.com/150)

**Figure 28: Cost-Benefits Tradeoff of Fast-tracking (Conceptualized)**

However, in the case, in which beating market conditions may not be an imperative, the cost-benefit tradeoff can be useful to compare the effectiveness of different fast-tracking strategies. This will help the project manager to establish a benchmark to compare various alternatives and find out an optimized fast-tracking strategy for successful completion of the project. In Figure 28, the cash flow of a typical building construction project during the project life cycle is conceptualized. The tradeoff of different fast-tracking strategies can be calculated by subtracting the increased cost to reduce the project duration (i.e., A-C in Figure 28) from the possible capital gain through the shortened project duration (i.e., B in Figure 28).
7.1.2 DPM System Behaviors

In this chapter, the behaviors of the DPM system will be examined with the case project. First, the case project is simulated with the base scenario. Then, simulations are done adapting the base case with various scenarios. In addition, the DPM system behaviors for the different scenarios are examined to measure the effect of project components on the project performance. Finally, the most desirable overlapping strategy between design and construction, and policies for the effective fast-tracking of the case project are suggested based on the analysis of the DPM system behaviors.

Base Case System Behaviors

The case project is simulated with flexible labor policy and no overlapping between design and construction, which will be the base case when compared to other alternatives. Some of the simulation results are as follows.

Workforce

As a result of the base simulation, the project is completed at week 200 with project costs of 197.19 U$ MIL. The graphs in Figure 29-a and Figure 29-b show the number of workers per week for each activity and the accumulated workers. As shown in the graphs, the cumulative number of workers involved in the project reaches 40,605 with an S shaped curve.

For a detailed analysis, the simulation result of required workforce for the excavation work activity is presented in Figure 30 together with that of schedule pressure and productivity. The simulation result shows that the number of required workforce varies throughout the work activity. In the DPM simulation, the actual workforce level is adjusted to the required level by the flexible labor control policy according to the model assumption. Figure 30 shows how the schedule pressure and the productivity influence the required workforce level and construction costs.
CONSTRUCTION WORKFORCE1

Figure 29-a: Workforce (1) during Construction
CONSTRUCTION WORKFORCE 2

Figure 29-b: Workforce (2) during Construction
For instance, schedule pressure significantly increases from the beginning of the activity and smoothly decreases after half of the activity has progressed. This is because in the initial stage of the activity, excavation proceeds slowly due to low productivity and the fact that construction usually commences with a relatively small number of workers. As the workers are getting familiar with the work environment, productivity continuously increases, which lowers the schedule pressure together with the already increased number of workers. As a result of the synergetic effect of relevant components, the number of required excavation worker continuously varies throughout the period. In particular, the required number of workers is getting more at its peak in week 65 as the excavation work proceeds.

**Productivity**

Going to a more detailed level, the productivity in the DPM simulation is determined by the function of schedule pressure, experience level with an activity, the effect of fatigue, and the normal productivity. Figure 31 shows the interrelationships among these factors, which influence the productivity of the excavation work. The experience level increases as the activity progresses with an S-shaped curve. There is no effect of workers' fatigue because overtime is not applied to the base case. Although workers' experience level continuously increases, the productivity drops in the later part of the activity as schedule pressure decreases. Consequently, the productivity for the excavation work continuously increases at different rates by week 68 and thereafter drops according to changes in associated variables.

To summarize, this chapter explored the basic dynamics of the case project within a sequential delivery scenario. The synergetic effect of interactions among construction components makes the construction process highly volatile to the work environment. As shown in the simulation results, workers' productivity continuously varies over time as relevant components including learning effect and schedule pressure are changed, which requires the different number of workers throughout the period. The simulation results obtained thus far will be compared to fast-tracking cases to be described in the following chapter, to measure the sensitivity of the case project's performance thus assist in effectively fast-tracking the project.
Figure 30: Workforce of Excavation Work
Figure 31: Productivity Determinants of Excavation Work
System Behaviors and Policy Implications

As discussed in Chapter 2, the effects of the feedback processes involved in construction can become greater under time and resource constraints. For this reason, the concurrent construction usually involves more diversified and dynamic feedback processes than does the sequential construction. The sensitivity study to be done in this chapter will support these arguments and provide an insight into the effective planning and management of concurrent building construction projects. For the sensitivity study, the case project is simulated with various scenarios. First, to quantify the impact of fast-tracking the project is simulated with different overlapping alternatives. Then, simulations are done to examine the effect of labor control policies on the construction. Finally, DPM performance under different construction settings such as different hiring time and quality management time are examined to measure the effect of project components on concurrent construction.

Design and Construction Overlapping

To examine the effect of the degree of overlapping between design and construction, simulations are done adapting the base case with different overlapping alternatives. The base case has a fixed headcount labor control policy and zero percent overlapping. Table 9 summarizes the result of the simulations for each case, in which their policies were changed.

The simulation results in Table 9 can be explained using the feedback processes represented in Figure 1-a and Figure 1-b. Increasing the overlapping degree between the design and construction created more changes in design and construction than those in the sequential method, which led to delays, counterbalancing the time reduction achieved by the increased overlapping. As a result, the initially expected time reduction was not achieved, which could make the vicious feedback processes in Figure 1-b dominant in the construction system. As a result, the project was completed with a relatively small amount of time reduction, compared to a significant increase in design and
construction changes (Table 9 shows that 100% overlapping resulted in 12.5% of time reduction, 91.1% of increase in design changes and 90.7% of increase in construction changes).

Table 9: Effect of Degree of Overlapping

<table>
<thead>
<tr>
<th>Degree of Overlapping</th>
<th>0%(Base)</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs Value (U$ MIL)</td>
<td>198.19</td>
<td>199.44</td>
<td>201.88</td>
<td>205.96</td>
<td>210.73</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>0.63</td>
<td>1.86</td>
<td>3.9</td>
<td>6.3</td>
</tr>
<tr>
<td>Duration Value (weeks)</td>
<td>205.2</td>
<td>199.7</td>
<td>194.5</td>
<td>188.7</td>
<td>179.5</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-2.6</td>
<td>-5.2</td>
<td>-8.0</td>
<td>-12.5</td>
</tr>
<tr>
<td>Workers Value (persons)</td>
<td>40,966</td>
<td>41,083</td>
<td>41,265</td>
<td>41,793</td>
<td>42,520</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>0.28</td>
<td>0.73</td>
<td>2.0</td>
<td>3.8</td>
</tr>
<tr>
<td>Design Changes Value (wu)</td>
<td>22,002</td>
<td>24,971</td>
<td>31,309</td>
<td>39,385</td>
<td>42,052</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>13.4</td>
<td>42.3</td>
<td>79.0</td>
<td>91.1</td>
</tr>
<tr>
<td>Construction Changes Value (man*hour)</td>
<td>35,702</td>
<td>39,670</td>
<td>47,547</td>
<td>58,850</td>
<td>68,118</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>11.1</td>
<td>33.1</td>
<td>64.8</td>
<td>90.7</td>
</tr>
</tbody>
</table>

This sensitivity study implies that more than 50% of overlapping between the design and construction may not be cost-effective and if more than 50% of overlapping is required by outer factors, more attention should be paid on reducing the cost impact.

**Labor Policies**

To examine the effect of changes in labor control policies, simulations are done adapting the base case with different assumptions for labor control policies. The base case has a fixed headcount labor control policy and zero percent overlapping. Table 10 summarizes the results of the simulations for each case.
### Table 10: Effect of Labor Policies

<table>
<thead>
<tr>
<th>Degree of Overlapping</th>
<th>Fixed HC (Base)</th>
<th>Flexible HC</th>
<th>Overtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs Value (U$ MIL)</td>
<td>198.19</td>
<td>197.19</td>
<td>208.7</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-0.5</td>
<td>5.3</td>
</tr>
<tr>
<td>Duration Value (weeks)</td>
<td>205.2</td>
<td>200.25</td>
<td>208</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-2.4</td>
<td>13.6</td>
</tr>
<tr>
<td>Workers Value (persons)</td>
<td>40,966</td>
<td>40,703</td>
<td>38,243</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-0.6</td>
<td>-6.6</td>
</tr>
<tr>
<td>Design Changes Value (wu)</td>
<td>22,002</td>
<td>21,722</td>
<td>26,714</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-1.2</td>
<td>21.4</td>
</tr>
<tr>
<td>Construction Changes Value (man*hour)</td>
<td>35,702</td>
<td>34,090</td>
<td>51,961</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-4.5</td>
<td>45.5</td>
</tr>
</tbody>
</table>

As a result of the simulation, flexible labor policy is found to be most efficient in terms of schedule and costs. In the case of overtime, there are more design and construction changes than base case, which leads to schedule and cost overrun. Overtime is typically adopted to make the schedule, however it is demonstrated here that overtime may result in more schedule overrun due to the dynamic behaviors of schedule pressure, productivity and quality. Overtime may lead to lower productivity and higher change rate when workers’ fatigue is accumulated. Thus, for this project, the overtime policy is not helpful to facilitate the project schedule. Figure 32 shows the result of the simulations normalized to compare each case when numbers for the base case are set as 1.0.

The effectiveness of labor control policies can vary depending on the nature of a project. However, the sensitivity study for overlapping alternatives implies that the construction performance in fast-tracking greatly depends on labor control polices. In particular, overtime may not be helpful to shorten the construction duration while driving up construction costs. Meanwhile, flexibility in labor control may contribute to reducing the construction duration and costs in fast-tracking.
In the previous chapter, flexible headcount was found to be the most efficient alternative for the case project. Thus, simulations are done to analyze the effect of the different time variables for key process on the case of flexible headcount and zero percent overlapping. Table 11 summarizes the results of the simulations for each case.

Generally, shortening a required time for a certain activity, such as quality management and labor hiring, is found to facilitate project duration but does not help reduce project costs. In particular, average labor hiring time and quality management time greatly affects the construction performance. Additionally, reducing quality management time decreases the number of workers and shortens project duration, while reducing the labor hiring time increases the number of workers. Figure 33 shows the result of the simulations normalized to compare each case when numbers for the base case are set to 1.0.
Table 11: Effect of Changes in Time Variables

<table>
<thead>
<tr>
<th>Degree of Overlapping</th>
<th>BASE</th>
<th>ADC=1</th>
<th>AQA=1</th>
<th>ALH=8</th>
<th>ASP=4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (U$ MIL)</td>
<td>197.19</td>
<td>197.16</td>
<td>196.13</td>
<td>196.64</td>
<td>197.83</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>0</td>
<td>-0.54</td>
<td>-0.28</td>
<td>0.32</td>
</tr>
<tr>
<td>Duration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (weeks)</td>
<td>200.25</td>
<td>200</td>
<td>189.5</td>
<td>193</td>
<td>198.25</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-0.13</td>
<td>-5.4</td>
<td>-3.6</td>
<td>-1.0</td>
</tr>
<tr>
<td>Workers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (persons)</td>
<td>40,703</td>
<td>40,575</td>
<td>39,326</td>
<td>41,777</td>
<td>41,545</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>-0.4</td>
<td>-3.3</td>
<td>2.6</td>
<td>2.0</td>
</tr>
<tr>
<td>Design Changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (wu)</td>
<td>21,722</td>
<td>24,218</td>
<td>24,708</td>
<td>22,219</td>
<td>23,657</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>11.5</td>
<td>13.7</td>
<td>2.2</td>
<td>8.9</td>
</tr>
<tr>
<td>Construction Changes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Value (man*hour)</td>
<td>34,090</td>
<td>38,170</td>
<td>39,228</td>
<td>36,799</td>
<td>37,671</td>
</tr>
<tr>
<td>Change from Base (%)</td>
<td>0</td>
<td>11.9</td>
<td>15.0</td>
<td>7.9</td>
<td>10.5</td>
</tr>
</tbody>
</table>

* Base Case: ADC= 2, AQA= 2, ALH= 16, ASP= 8  Unit: weeks
ADC: Average Design Change Time, AQA: Average Quality Approve Time
ALH: Average Labor Hiring Time, ASP: Average Schedule Perception Time

Figure 33: Effect of Changes in Time Variables
The above sensitivity study shows that reducing a required time for a certain activity helps facilitate the construction schedule in fast-tracking. This implies that to achieve effective fast-tracking, the decision-making process in design and construction should be shortened and information flow among project functions should be streamlined to assist in reducing the decision-making time.

Cost-Benefits Analysis

Findings obtained from the sensitivity studies thus far have their policy implications, narrowing downing desirable sets of the construction components. In this chapter, the effectiveness of various fast-tracking alternatives for the case project is determined among the selected construction settings. Given the different construction settings, DPM analyzes the trade-off between the increased costs to reduce project duration and the possible capital gain through shortened duration. The trade-off is very useful for establishing a benchmark to compare the effectiveness of fast-tracking strategies. With a gaming function provided by DPM, the estimation of the trade-off is possible, not only before the commencement of a project, but also during the project duration. For the case project, possible capital gain to be achieved through earlier completion is assumed US 10 MIL per year. Under this assumption, various scenarios are simulated and some of the cases are listed Table 10.

The result shows that cases with 50% of an overlapping option are favorable alternatives in terms of trade-off. Case 2 which has one week as the average quality approval time and sixteen weeks as the average labor hiring time is most efficient among them. Meanwhile, case 7 has the shortest project duration despite a negative trade-off. As shown in Table 12, all cases with 100% overlapping have a negative number of trade-off. These results imply that when a project is fast-tracked by more than 50%, fast-tracking may not be effective. If more than 50% of overlapping is required by outer factors, labor control should be flexible during the project duration and quality approval and labor hiring time should be shortened as long as possible in order to reduce the cost impact of fast-tracking.
Table 12: Cost-Benefits Tradeoffs of Alternative Policies

<table>
<thead>
<tr>
<th>Description</th>
<th>Degree of Overlapping (%)</th>
<th>Flexible Headcount</th>
<th>Avg Quality Approval Time (weeks)</th>
<th>Avg Labor Hiring Time (weeks)</th>
<th>Project Duration (weeks)</th>
<th>Project Costs (US MIL)</th>
<th>Trade-off (US MIL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base</td>
<td>0</td>
<td></td>
<td>2</td>
<td>16</td>
<td>205.2</td>
<td>198.2</td>
<td>0.00</td>
</tr>
<tr>
<td>Case 1</td>
<td>50</td>
<td></td>
<td>1</td>
<td>16</td>
<td>181.7</td>
<td>201.7</td>
<td>0.94</td>
</tr>
<tr>
<td>Case 2</td>
<td>50</td>
<td>•</td>
<td>1</td>
<td>16</td>
<td>180.7</td>
<td>199.2</td>
<td>3.69</td>
</tr>
<tr>
<td>Case 3</td>
<td>50</td>
<td>•</td>
<td>1</td>
<td>8</td>
<td>175.5</td>
<td>201.9</td>
<td>1.97</td>
</tr>
<tr>
<td>Case 4</td>
<td>50</td>
<td>•</td>
<td>1</td>
<td>4</td>
<td>174.2</td>
<td>202.5</td>
<td>1.61</td>
</tr>
<tr>
<td>Case 5</td>
<td>100</td>
<td></td>
<td>1</td>
<td>16</td>
<td>179.5</td>
<td>210.73</td>
<td>- 7.54</td>
</tr>
<tr>
<td>Case 6</td>
<td>100</td>
<td>•</td>
<td>1</td>
<td>8</td>
<td>159.0</td>
<td>212.0</td>
<td>-5.00</td>
</tr>
<tr>
<td>Case 7</td>
<td>100</td>
<td>•</td>
<td>1</td>
<td>4</td>
<td>158.0</td>
<td>212.9</td>
<td>-5.70</td>
</tr>
</tbody>
</table>
To summarize, this chapter has demonstrated how the use of DPM can help the project manager assess the time and cost consequences of overlapping the design and construction processes, and establish policies. The effectiveness of fast-tracking policies can vary depending on the nature of a project. In addition, delivery systems adopted to carry out a project are also closely related to effective fast-tracking policies. However, the sensitivity studies presented in this chapter have some general guidelines for the effective fast-tracking of building construction projects. That is: 1) the planning and management requires a systematic and dynamic approach due to the diversified and dynamic feedback processes involved in concurrent construction, 2) the synergetic effect of those feedback processes makes the construction process highly volatile to the work environment, making workers’ productivity continuously vary throughout the construction process, which requires the flexible labor control, and 3) the decision-making process in design and construction should be shortened, since time delays can magnify the ripple effects of the feedback processes under time and resource constraints.
7.2 VALIDATION OF RELIABILITY BUFFERING

This chapter examines the effectiveness of reliability buffering that has been suggested in Chapter 5.4 by utilizing DPM model structure and observing DPM system behaviors. As discussed in the previous chapters, non-value adding iterations could be caused mainly by changes and the impact of those changes on the construction performance would have different types, paths, and magnitudes, depending on the attribute of changes (intended or unintended), and the location of change generation and discovery (upstream work or downstream work). These findings are important in understanding the roles of reliability buffering in construction and also contribute to determining effective buffer size, given certain construction conditions.

In order to examine the effectiveness of reliability buffering, first, the change impact on the construction performance is analyzed based on the generic process model structure in Figure 23. During the analysis, explanations are made on logical backgrounds for reliability buffering to enhance the project schedule performance by reducing the change impact. Following this, the effectiveness of reliability buffering is scrutinized by observing the DPM system behaviors.

7.2.1 Change Impact vs. Reliability Buffering

The change impact on the construction performance can vary depending on whether changes have been made on purpose (managerial changes) or by mistake (unintended changes). The analysis of the change impact begins with managerial changes. As discussed in the previous chapters, managerial changes are made by managerial decisions during quality management or project monitoring and control. Changed tasks as a result of adopting a change option become a change source that can cause subsequent changes. In the upstream process model on Figure 35, tasks flowing through \textit{UPChangeAccomodateRate} are all managerial changes resulting from a
managerial decision and WorkReleaseRate also contains managerial changes. According to the definition of change action in Table 3, managerial changes themselves are released to WorkReleased, while other tasks in WorkReleased are moved to Worktodo through ReprocessRequest on WorkReleasedRate. The amount of tasks moved is as much as the impact magnitude of the managerial changes.

By adopting the change option, it is possible to avoid the direct impact resulting from rework (R_{up} in Figure 35). However, as shown in Figure 35, the decision on the change option in an activity can create subsequent non-value adding iterations within the activity (C_{up}) and in the downstream activity (C_{dn}). Managerial changes might have more impact on the construction performance than rework on the original changes, depending on the sensitivity of associated tasks to the managerial changes and how much work have been already done at the change impact time. In Figure 35, the impact of the change option on the upstream activity (C_{up}) is in proportion to the sensitivity of the upstream work to internal changes and the progress of the upstream work. Meanwhile, the change impact on the downstream activity (C_{dn}) can be measured by a function of their sensitivities to the upstream work change and the downstream work progress at the change impact time.

Once the reliability buffer is applied to this example case, it is possible to absorb the impact of subsequent changes in the downstream activity (C_{dn}) by systematically assigning a time buffer. Depending on the characteristics of a construction system, managerial changes have different intensity and magnitude (mostly according to production rate, reliability, and managerial tendency to adopt the change option), and the susceptibility to the changes can vary (according to internal or external sensitivity). Consequently, systematically located and sized time buffers can help reduce the domino effect of the change impact on the downstream work by effectively controlling the start time and progress of the downstream work.
Figure 35: Impact of Managerial Changes
Meanwhile, unintended changes have more complex impact patterns. As conceptualized in Figure 36, normally as the discovery time lengthens and the discovery location goes away from the location of the change generation, the impact of changes becomes greater. In addition, when changes are made on the work, based on which other work have been done already, they can create a ripple effect impacting other work as well. Whether changes occur due to the low reliability of internal work or due to hidden changes made during the upstream work can also determine the change impact. This is because the work quality of an activity is in proportion to its reliability, while the upstream work change effect on work quality is a non-linear function.

![Figure 36: Basis of Unintended Change Impact](image)

By incorporating all of the impact determinants discussed above into the generic construction process, the change impact on the downstream activity of the cases in Figure 36 are analyzed in terms of types, paths, and magnitude, as summarized in Table 13. For effective analysis, it is assumed that the rework option is adopted at each managerial decision point. However, in reality the change option is more often adopted during construction, facing unintended changes. This can be represented by combining the impact loop paths of unintended changes to be discussed below with those of managerial changes described in Figure 37.
Table 13: Change Impacts and Role of Reliability Buffering

<table>
<thead>
<tr>
<th>Cases</th>
<th>Source, Discovery Time &amp; Location</th>
<th>Impact on Downstream</th>
<th>Roles of Reliability Buffer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Types</td>
<td>Path Length</td>
</tr>
<tr>
<td>Case I</td>
<td>Upstream changes found after upstream work</td>
<td>Delay</td>
<td>Short</td>
</tr>
<tr>
<td>Case II</td>
<td>Upstream changes found before downstream work</td>
<td>Delay</td>
<td>Long</td>
</tr>
<tr>
<td>Case III</td>
<td>Upstream changes found after downstream work</td>
<td>Delay and Quality Impact</td>
<td>Very long</td>
</tr>
</tbody>
</table>

As described in Table 13, in Case I where changes are found through quality management in the upstream work, the type of impacts on the downstream work is a time delay and its magnitude is weak due to a short impact path. Since the problematic tasks are not released to the downstream activity until they are corrected, there is a time delay but no direct impact on the downstream work quality. Meanwhile, in Case II upstream work hidden changes are discovered through pre-checking in the downstream work and discovered changes are returned to the upstream work. Case II has the same impact type as Case I, but the impact magnitude is greater than that of Case I. This is because Case II has a longer impact path length, as depicted in Figure 37 and ripple effects are associated with the impact process. For example, suppose that during steel member erection, it is found that some steel members are not fit to others. In this case, the steel worker will inform the design team of the mismatch of those steel members and return the members to the steel manufacturer. Then, the design team should re-size the steel members and the new specification on the steel members will be forwarded to the steel manufacturer. In addition, if newly specified members do not fit other structural components that have physical connections with the corrected steel members, the design team may change the
specification for the connected structural components as well. During the iteration period, workers and cranes that have been working on the site have to wait for new steel members to arrive. However, once reliability buffering is applied to this case, resource idle time can be reduced and actual work can proceed without interruptions once started. During the buffer period, the steel worker can find the mismatch of the problematic steel members before bringing workers and cranes to the site. As a result, it is possible to avoid unnecessary resource idle time.

Moreover, in Case III where upstream work hidden changes are discovered after the corresponding downstream work has been done, the problems made in the upstream work impact the downstream work quality as well as delaying the process. The magnitude of the impact is also greater than other cases. This is because Case III has the longest impact path, as depicted in Figure 37. In addition, quality deterioration and ripple effects are associated with impact processes involved in Case III. For example, suppose that after pouring concrete into forms for the foundation, it is found that the strength of the poured concrete is not enough to support the dead load of the building. In this case, the resources commissioned both in the foundation and design work are squandered and the foundation work is delayed during the demolition of the problematic concrete and re-calculation of the concrete strength. This case can benefit the most from reliability buffering. Once reliability buffering is applied to this case, resource idle time can be reduced and resource use can be also economized by decreasing the possibility of the upstream impact on work quality. In the above example, by applying a reliability buffer, it is possible to thoroughly check the appropriateness of the concrete strength before pouring concrete. As a result, the problem in concrete strength can be found beforehand and concrete demolition and workers’ idle time can be avoided.
Case I

Figure 37-a: Impact of Unintended Changes (Case I)
Case II

Figure 37-b: Impact of Unintended Changes (Case II)
Case III

Figure 37-c: Impact of Unintended Changes (Case III)
7.2.2 Observation on Model Behaviors

The effectiveness of reliability buffering has been examined so far based on the structure of the generic process model. In particular, logical backgrounds for reliability buffering were discussed in line with identifying different types, paths, and magnitudes of the change impact. In this chapter, the effectiveness of reliability buffering is analyzed by observing the behaviors of the DPM system developed in this thesis. In addition, the role of construction characteristics in determining the effective buffer size is examined based on the results of sensitivity studies, which have been done by adapting the base scenario with various construction characteristics and precedence relationships. Lastly, buffering implications obtained from the DPM system behaviors are discussed.

Validation of Reliability Buffering

In order to validate the effectiveness of reliability buffering, the base scenario, which is detailed in Table 14, is simulated with the two cases, having a reliability buffer and not having a reliability buffer. The base scenario includes 50 days duration for each of the two activities and a precedence relationship such that the activity B can start only 25 days after the start of the activity A. In addition, it is assumed that 20% of the duration for both activities is a schedule contingency and 50% of the schedule contingency can be used for reliability buffering. For the buffering case, the original schedule is adjusted according to reliability buffering steps. 10-days contingency buffers are taken off from both the activity A and B. Then, a 5-days reliability buffer is fed at the beginning of the activity B duration and it is characterized as a time to ramp up necessary resources for the activity B and to find problems in the activity A. Meanwhile, during the simulation, no managerial actions to catch up delayed schedule are considered in order to more effectively measure the role of reliability buffering on the schedule performance.
Table 14: Base Scenario for Model Simulation

<table>
<thead>
<tr>
<th>Descriptions</th>
<th>Activity A</th>
<th>Activity B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Durations (days)</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Precedence Relationships</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Construction Characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production Types</td>
<td>Fast</td>
<td>Fast</td>
</tr>
<tr>
<td>Reliability Types</td>
<td>Unreliable</td>
<td>Unreliable</td>
</tr>
<tr>
<td>Value</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>Sensitivity Types</td>
<td>Sensitive</td>
<td>Sensitive</td>
</tr>
<tr>
<td>Value</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Buffering Schedule Contingency</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>Fraction of Buffering</td>
<td>NA</td>
<td>0.5</td>
</tr>
<tr>
<td>Buffer Size (days)</td>
<td>NA</td>
<td>5</td>
</tr>
<tr>
<td>Quality Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thoroughness</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Period (days)</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

As a result of the simulation, the base scenario is completed at day 90 in the no buffering case, while it is completed at day 85, in which reliability buffering is considered. For the activity B, the applied reliability buffer saved 15.4% of the original duration (65 days to 55 days). The difference in the simulated durations, given the same amount of resources and no interventions from outside of the system indicates that the reliability buffer applied to the activity B effectively absorbed change impacts from the activity A and reduced resource waste and resource idle time, as discussed in the previous chapters. Meanwhile, the simulated durations of both cases are longer than CPM-based durations, which are 75 days in the no buffering case and 70 days in the buffering case. This is because the dynamic project model simulates non-value adding iterations based on the given activity characteristics, which are not considered in the CPM-based calculation.

In Figure 38, for the early days the activity B in the buffering case is progressing at a slower rate than in the no buffering case due to a later start. In the buffering case, however, hidden changes generated in the activity A are significantly decreased during the buffered period. As a result, the activity B in the buffering case can have a higher work quality and resource utilization rate, which makes it possible to catch up with
Figure 38: Simulation of Reliability Buffering
the progress of the no buffering case sometime later. In addition, an appropriate schedule pressure created by the reduced target duration (from 50 days to 40 days) increases the productivity of both activities, which also assists in achieving an earlier completion. The simulation results showed that the reliability buffering could be beneficial for the schedule performance with no managerial actions taken. Once managerial or control actions to correct deviations from the planed schedule performance are allowed, the role of reliability buffering can be extended. Depending on driving constraints, project managers may attempt to recover the delayed schedule at the expense of costs. In this case, the reliability buffering can be beneficial for the cost performance as well by reducing resource misuse and idle time.

**Sensitivity Test**

Having examined the effectiveness of reliability buffering, this chapter discusses the role of construction characteristics in determining the effective buffer size, given different precedence relationships. The discussion begins with presenting the results of multivariate sensitivity simulations. These simulations have been performed five hundreds times by adapting the base scenario in the previous chapter with different conditions given in Table 15.

**Table 15: Simulation Settings for Sensitivity Test**

<table>
<thead>
<tr>
<th>Des.</th>
<th>Precedence Relation</th>
<th>Buffer Applied to Activity B (days)</th>
<th>Production Type (both activities)</th>
<th>Reliability (activity A)</th>
<th>Sensitivity (activity B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>Start-to-Start 25 days</td>
<td>5</td>
<td>Fast</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>Max.</td>
<td></td>
<td>5</td>
<td>Fast</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The results of the simulations show that with the same size of buffer, the completion date of the downstream, activity B ranges from 80 to 142, depending on the values of the upstream work reliability and the downstream work sensitivity. This implies that the fixed size of reliability buffer can be beneficial for the schedule only under a certain
condition and for this reason, the buffering size should vary depending on the construction characteristics of associated activities, as argued by Pena-Mora and Li [2001].

In addition, system behaviors of the dynamic project model have been explored to examine the role of each construction characteristic in buffering. In the previous chapter, it was hypothesized based on Pena-Mora and Li [2001]'s overlapping framework that a slower and unreliable upstream work, or a faster and sensitive downstream work requires a longer buffer period. In order to validate this hypothesis, the buffer size of activity B that minimizes the activity duration given a different condition was explored and then the relationships among resulting values were compared.

The overall result of the buffering optimization, given different construction characteristics supports Pena-Mora and Li [2001]'s argument on the effective overlapping. As indicated in Table 14, the faster downstream case required a longer buffering (6.3 days) than did the slower downstream case (4.2 days). In addition, the simulation results indicate that the less reliable the upstream work was and the more sensitive the downstream work was, the longer required buffering period was. However, the faster upstream work production case turned out to require a longer buffer than a slower upstream work production, which is opposite to the research hypotheses. This is due to the ripple effect considered in the dynamic project model and the precedence relationship applied to the activities. When the upstream tasks that have been once completed need to be corrected by change correction requests from the downstream work, the ripple effect takes place in tasks that are physically or functionally related to the problematic tasks. As discussed before, this ripple effect is in proportion to the upstream work progress at the impact point. Since a faster production produces more tasks during the first half of total progress, more tasks are impacted by the ripple effect during the given period. In this sensitivity study, the reliability buffering started 25 days after the start of the upstream work and thereafter the upstream work has been impacted by
Figure 39: Multivariate Sensitivity Simulations of Downstream Progress
### Table 16: Optimal Buffer Sizes in Different Conditions

<table>
<thead>
<tr>
<th>Description</th>
<th>Base Scenario</th>
<th>Alternatives</th>
<th>Changes from Base Scenario</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedence Relationships</td>
<td>SS 25</td>
<td>SS30</td>
<td>SS35</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>6.3</td>
<td>4.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Upstream Production Fast</td>
<td></td>
<td>Slow</td>
<td></td>
</tr>
<tr>
<td>Buffer Size</td>
<td>6.3</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>Downstream Production Fast</td>
<td></td>
<td>Slow</td>
<td></td>
</tr>
<tr>
<td>Buffer Size</td>
<td>6.3</td>
<td>4.2</td>
<td></td>
</tr>
<tr>
<td>Upstream Reliability Unreliable (0.85)</td>
<td>Reliable (9.0)</td>
<td>Highly Reliable (1.0)</td>
<td>↑</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>6.3</td>
<td>2.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Downstream Sensitivity Sensitive (1.0)</td>
<td>Insensitive (0.75)</td>
<td>Totally Insensitive (0)</td>
<td>↓</td>
</tr>
<tr>
<td>Buffer Size</td>
<td>6.3</td>
<td>4.9</td>
<td>0.0</td>
</tr>
</tbody>
</table>
changes discovered in the downstream work. Accordingly, a longer buffer period was required for a faster upstream work. This result implies that the role of the upstream work production type is closely related to precedence relationships, which decides the start time of the reliability buffer.

In addition, it turned out that each determinant has a different effect on effective buffering and precedence relationships involved in activities also contribute to determining the most desirable buffer size, which was hypothesized in the previous chapters. Non-linearities in the required buffer size for different cases explain different roles of each buffering determinant. In addition, as the lag time of the applied precedence relationship increased (25 days to 35 days), the required buffer size for the downstream activity decreased (6.3 days to 3 days), which demonstrates the effect of precedence relationships on effective buffering.

Summarizing this chapter, the effectiveness of reliability buffering was examined by understanding the change impact on concurrent processes and observing the DPM system behaviors. Findings in this chapter indicate that reliability buffering can help enhance the project performance by reducing change impacts and each construction characteristic has different effect on effective buffering. In addition, the non-linearity of their impacts explains why a simulation approach needs to be introduced to effectively determine buffer size. In order to gain reality and show its applicability to a real construction project, reliability buffering is applied to bridge construction projects, which will be detailed in the next chapter.
This chapter presents an application example of DPM to bridge construction planning. In order to help prepare a robust construction plan, DPM is being applied to the construction of 27 bridges, which is a part of a $400 million Design/Build/Operate/Transfer (DBOT) project awarded to Modern Continental Companies, Inc. 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. The development process is expected to span 42 months with the project completion achieved in February, 2004. The project scope includes widening the 21-mile of the state roadway and the existing 15 underpass bridges, and renovating 12 overpass bridges. In this chapter, the application of DPM to the Treble Cove Road Bridge construction is presented focusing on the role of reliability buffering in schedule planning.

This case project is one of the overpass bridge renovations and consists of 28 design and construction activities after appropriately aggregating original activities in accordance with the DPM fundamentals. In addition, for a more accurate observation on DPM system behaviors, some activities that are out of the contractor’s control such as the owner’s survey and appraisal on the road are excluded from the DPM simulation. The scope of the case project includes the demolition of the existing bridge, which is shown in Figure 40. Meanwhile, in order to get necessary data, a series of interviews with schedulers and engineers involved in the project have been made, through which the construction characteristics of the project activities summarized in Table 17 were obtained. In terms of having activities similar to other 26 bridge construction projects, this case project provides a valuable opportunity to examine the applicability of DPM and reliability buffering in a real world setting.
Figure 40: Project Location and Existing Bridge
# Table 17: Input Data for the Treble Cove Project

<table>
<thead>
<tr>
<th>Activity Code</th>
<th>Activity Name</th>
<th>Duration (days)</th>
<th>Driving Precedence Relation</th>
<th>Production</th>
<th>Reliability</th>
<th>Sensitivity</th>
<th>Effective Buffering Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sketch Plans</td>
<td>33</td>
<td>S</td>
<td>HU</td>
<td>IS</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Final Plans</td>
<td>66</td>
<td>1 ss20</td>
<td>S</td>
<td>HU</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ROW Acquisition</td>
<td>130</td>
<td>2 ss3</td>
<td>S</td>
<td>R</td>
<td>IS</td>
<td>0.25</td>
</tr>
<tr>
<td>4</td>
<td>Shop Drawing Submittals</td>
<td>35</td>
<td>2</td>
<td>F</td>
<td>R</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Shop Drawing Review/BPads</td>
<td>30</td>
<td>4</td>
<td>S</td>
<td>U</td>
<td>IS</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>Shop Drawing Review/Structural Steel</td>
<td>30</td>
<td>4</td>
<td>S</td>
<td>U</td>
<td>IS</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>Shop Drawing Review/Rebar</td>
<td>30</td>
<td>4</td>
<td>S</td>
<td>U</td>
<td>IS</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>Shop Drawing Review/SOE Plans</td>
<td>30</td>
<td>4</td>
<td>S</td>
<td>U</td>
<td>IS</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>Steel Fabrication/Rebar</td>
<td>60</td>
<td>7 ss 5</td>
<td>S</td>
<td>N</td>
<td>S</td>
<td>0.75</td>
</tr>
<tr>
<td>10</td>
<td>Steel Fabrication/BPads</td>
<td>120</td>
<td>5 ss 5</td>
<td>S</td>
<td>N</td>
<td>S</td>
<td>0.75</td>
</tr>
<tr>
<td>11</td>
<td>Steel Fabrication/Structural Steel</td>
<td>120</td>
<td>6 ss 5</td>
<td>S</td>
<td>N</td>
<td>S</td>
<td>0.75</td>
</tr>
<tr>
<td>12</td>
<td>Steel Fabrication/Sheet &amp; Brace</td>
<td>45</td>
<td>8 ss 5</td>
<td>S</td>
<td>N</td>
<td>S</td>
<td>0.75</td>
</tr>
<tr>
<td>13</td>
<td>Prepare Site for Abutment E/W</td>
<td>33</td>
<td>8</td>
<td>F</td>
<td>R</td>
<td>IS</td>
<td>0.25</td>
</tr>
<tr>
<td>14</td>
<td>Prepare Site for Center Pier</td>
<td>13</td>
<td>12</td>
<td>S</td>
<td>R</td>
<td>IS</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>Construct Abutment E/W</td>
<td>30</td>
<td>13 fs 2</td>
<td>S</td>
<td>N</td>
<td>S</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>Construct Center Pier</td>
<td>15</td>
<td>15</td>
<td>S</td>
<td>N</td>
<td>IS</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>Set BPads and Girders</td>
<td>5</td>
<td>10</td>
<td>S</td>
<td>N</td>
<td>IS</td>
<td>0.5</td>
</tr>
<tr>
<td>18</td>
<td>Construct Superstructure</td>
<td>20</td>
<td>17</td>
<td>S</td>
<td>N</td>
<td>IS</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>Bell Telephone Cable</td>
<td>80</td>
<td>17 ss 0</td>
<td>S</td>
<td>U</td>
<td>IS</td>
<td>0.75</td>
</tr>
<tr>
<td>20</td>
<td>Relocate Gas Line</td>
<td>15</td>
<td>18</td>
<td>S</td>
<td>U</td>
<td>S</td>
<td>0.5</td>
</tr>
<tr>
<td>21</td>
<td>Relocate Water Line</td>
<td>15</td>
<td>20</td>
<td>S</td>
<td>U</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>Install Telephone DB</td>
<td>15</td>
<td>21</td>
<td>S</td>
<td>U</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>Realign Treble Cove Rd</td>
<td>10</td>
<td>22</td>
<td>S</td>
<td>R</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>Realign Rte 3 NB Ramps</td>
<td>20</td>
<td>23</td>
<td>F</td>
<td>R</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>Realign Rte 3 SB Ramps</td>
<td>20</td>
<td>24</td>
<td>F</td>
<td>R</td>
<td>S</td>
<td>0.75</td>
</tr>
<tr>
<td>26</td>
<td>Demolish Existing Ctr Span</td>
<td>10</td>
<td>25</td>
<td>S</td>
<td>R</td>
<td>IS</td>
<td>0.75</td>
</tr>
<tr>
<td>27</td>
<td>Demolish Existing EAbut</td>
<td>10</td>
<td>26</td>
<td>S</td>
<td>R</td>
<td>IS</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>Demolish Existing WAbut</td>
<td>10</td>
<td>27</td>
<td>S</td>
<td>R</td>
<td>IS</td>
<td>0</td>
</tr>
</tbody>
</table>

*Note

1. Default Precedence Relationship: FSO
2. General Convention for Precedence Relationship: preceding activity - type - lead/lag
3. Production Type: F(Fast), S(Slow)
4. Reliability: R( Reliable), N(Normal), U(Unreliable), HU(Highly Unreliable)
5. Sensitivity: S(Sensitive), N(Normal), IS(Insensitive)
6. Effective Buffering Ratio: The buffering ratio of individual activities that can create the best schedule for the case project
The planning of the case project will be processed in the following steps. First, the case project is simulated with 100% flexible headcount policy and no buffering, which will be the base scenario, when compared to other scenarios. Then, in order to measure the effect of reliability buffering and construction policies on the project performance, simulations are done adapting the base case with various scenarios. Finally, the most desirable set of construction policies for the case project are suggested based on the analysis of the DPM system behaviors.

7.3.1 Base Case System Behaviors

The simulated actual duration of the base case where 100% flexible labor control and no buffering are applied is 559 working days. This is 168 days longer than the CPM-based duration of the base case, which is 391 working days. The difference in the completion time implies that there were a lot of non-value adding iterations during the simulation of the case project and they affected construction sequences. Actually, the construction team is working to address such issues that the design development of the Treble Cove Road Bridge project (approximately 16.3% complete) was already shown significant delay as of Feb 1, 2001 and construction has not been yet started. Some of these issues are due to the fact that this project was awarded to the contractor before the detailed scope of the project has been established. As a result, changes on the design work were frequently requested from the owner side during sketch plan (activity 1), final plan (activity 2), and shop drawing submittal (activity 4), which resulted in a lot of design iterations. In addition, this case project was the first design/build contract for the members of development team in the owner side, expected level of coordination among the owner, designer and constructor has not been met to date and design iterations encountered were difficult to handle. Based on interviews with the design and construction team, these challenges in the design development were represented as ‘Highly Unreliable’ in DPM and DPM-generated actual durations for those activities show how much non-value adding iterations caused by changes can affect the project progress in a quantitative manner.
Figure 41: Primavera-Generated Activity Durations
Figure 42: DPM-Generated Activity Performance
7.3.2 DPM System Behaviors and Policy Implications

The previous chapters discussed that DPM would help prepare a more robust construction plan against uncertainties and provide policy guidelines for the planning and control of construction projects. The sensitivity study to be done in this chapter will support these arguments and provide an insight into the effective planning and management of concurrent construction projects. For the sensitivity study, the case project is simulated with various scenarios. First, simulations are done to examine the effect of labor control policies on the case project. Then, to find out the most desirable reliability buffering ratio for activities, the case project is simulated with different buffering alternatives. Finally, DPM performance under different construction settings such as different labor hiring and RFI reply time is examined to measure the effect of time components on the project performance.

**Labor Policies**

To examine the effect of changes in labor control policies, simulations are done adapting the base case with different assumptions for labor control policies including flexible headcount (Case 1 to 4) and overtime (Case 5 to 8). The base case has a 100% flexible labor control policy and no buffering. Table 18 summarizes the results of the simulations for each case.

As a result of the simulation, 100% flexible labor policy (Case 1) is found to be most efficient in terms of schedule and cost reduction. In particular, as the willingness to adopt the policy increases, the simulated actual project duration is getting shorter and project costs are getting less. Meanwhile, overtime contributes to facilitating the project schedule to some extent but its effectiveness is questioned, once increased project costs are considered. As demonstrated in the simulation results, applied overtime lowered productivity and increased change rate, as workers’ fatigue was accumulated. Figure 45
shows the result of the simulations normalized to compare each case when numbers for the base case are set as 1.0.

Table 18: Effect of Labor Policies

<table>
<thead>
<tr>
<th>Cases</th>
<th>Labor Control Policies</th>
<th>Completion Time (Days)</th>
<th>Labor Hours (worker*hour)</th>
<th>Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100% Flexible</td>
<td>559</td>
<td>1.305M</td>
<td>100%FH</td>
</tr>
<tr>
<td>2</td>
<td>75% Flexible</td>
<td>565</td>
<td>1.328M</td>
<td>75%FH</td>
</tr>
<tr>
<td>3</td>
<td>50% Flexible</td>
<td>573</td>
<td>1.369M</td>
<td>50%FH</td>
</tr>
<tr>
<td>4</td>
<td>25% Flexible</td>
<td>594</td>
<td>1.451M</td>
<td>25%FH</td>
</tr>
<tr>
<td>5</td>
<td>100% Overtime</td>
<td>583</td>
<td>1.381M</td>
<td>100%OT</td>
</tr>
<tr>
<td>6</td>
<td>75% Overtime</td>
<td>586</td>
<td>1.362M</td>
<td>75%OT</td>
</tr>
<tr>
<td>7</td>
<td>50% Overtime</td>
<td>589</td>
<td>1.345M</td>
<td>50%OT</td>
</tr>
<tr>
<td>8</td>
<td>25% Overtime</td>
<td>597</td>
<td>1.322M</td>
<td>20%OT</td>
</tr>
</tbody>
</table>

The effectiveness of labor control policies can vary depending on the nature of a project. However, the sensitivity study for labor control policies presented thus far implies that the performance of concurrent construction greatly depends on labor control polices. In particular, overtime may be helpful to shorten the construction duration but once considering the time and cost tradeoff, it may not be an effective way in concurrent construction. Meanwhile, flexibility in labor control contributes to reducing the concurrent construction duration and costs by assigning workforce in a timely manner.
Figure 43-a: Flexible Labor Policies vs Project Progress
Figure 43-b: Flexible Labor Policies vs Work Hours
Figure 43-c: Flexible Labor Policies vs Cumulative Work Hours
Figure 44-a: Overtime Labor Policies vs Project Progress
Graph for CumulativeLaborHoursPJ

Figure 44-b: Overtime Labor Policies vs Work Hours
Figure 45: Labor Policies vs Project Performance
**Reliability Buffering**

In order to examine the role of reliability buffering, the case project has been simulated with various buffering scenarios; not having buffer, having uniform buffer, and having buffer based on activities’ characteristics. In addition, 100% flexible labor control policy was commonly applied to all the buffering cases and known contingency factors of activities are set as 20% of their original durations based on the interviews with engineers and schedulers. Having simulated the scenarios with DPM, the results summarized in Table 19 were obtained.

The simulated actual duration of the no buffering case (Case 1), which was the base case in the previous chapter, is 559 working days. Meanwhile, as indicated in Table 19, the buffering cases have shorter simulated actual durations (477, 463, 452, 451, and 445 in Case 2, 3, 4, 5, and 6 respectively). In the buffering cases, applied reliability buffers contributed to reducing the upstream change impact and non-value adding iterations. As a result, the resource idle time and waste were reduced, which made it possible to more effectively utilize given workforce. In particular, Case 6, where reliability buffering was applied based on activities’ characteristics, turned out to most effectively enhance the schedule and cost performance as hypothesized earlier in this thesis.

**Time Variable**

As discussed in the previous chapters, construction inherently involves time delays, which greatly contribute to determining construction performance. In this chapter, simulations are done to analyze the effect of the different time variables for key process on the case of no buffering and 100% flexible headcount, which was found to be the most efficient labor policy for the case project in the previous chapter. Table 20 summarizes the results of the simulations for each case.
Table 19: Project Completions According to Buffering

<table>
<thead>
<tr>
<th>Cases</th>
<th>Buffering (Buffer Size)</th>
<th>Completion Time (Days)</th>
<th>Deviation from Case 1 Days</th>
<th>Deviation from Case 1 %</th>
<th>Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>None</td>
<td>559</td>
<td></td>
<td></td>
<td>base</td>
</tr>
<tr>
<td>2</td>
<td>Uniform (25%)</td>
<td>477</td>
<td>-82</td>
<td>-14.6</td>
<td>rb25</td>
</tr>
<tr>
<td>3</td>
<td>Uniform (50%)</td>
<td>463</td>
<td>-96</td>
<td>-17.1</td>
<td>rb50</td>
</tr>
<tr>
<td>4</td>
<td>Uniform (75%)</td>
<td>452</td>
<td>-107</td>
<td>-19.1</td>
<td>rb75</td>
</tr>
<tr>
<td>5</td>
<td>Uniform (100%)</td>
<td>451</td>
<td>-108</td>
<td>-19.3</td>
<td>rb100</td>
</tr>
<tr>
<td>6</td>
<td>Activity 1 0%</td>
<td>445</td>
<td>-114</td>
<td>-20.3</td>
<td>rb_individual</td>
</tr>
<tr>
<td></td>
<td>Activity 2 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 3 25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 4 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 5 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 6 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 7 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 8 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 9 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 10 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 11 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 12 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 13 25%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 14 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 15 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 16 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 17 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 18 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 19 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 20 50%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 21 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 22 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 23 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 24 100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 25 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 26 75%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 27 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Activity 28 0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note 1. Buffer Size: Fraction of Taken-off Contingency Buffer (20% of Activity Original Duration)
2. 100% Flexible Labor Control Policy Applied to All Cases*
Figure 46-a: Reliability Buffering vs Project Progress
Figure 46-b: Reliability Buffering vs Work Hours
Table 20: Effect of Time Delays

<table>
<thead>
<tr>
<th>Cases</th>
<th>Time to Increase Workforce (days)</th>
<th>Time to Reply RFI*</th>
<th>Completion Time (Days)</th>
<th>Labor Hours (worker*hour)</th>
<th>Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>3</td>
<td>559</td>
<td>1.305M</td>
<td>base</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>3</td>
<td>577</td>
<td>1.309M</td>
<td>tFH14</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>3</td>
<td>588</td>
<td>1.314M</td>
<td>tFH21</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>548</td>
<td>1.300M</td>
<td>tFH3</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>6</td>
<td>495</td>
<td>1.181M</td>
<td>RFI6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>9</td>
<td>478</td>
<td>1.147M</td>
<td>RFI9</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1.5</td>
<td>722</td>
<td>1.631M</td>
<td>RFI1.5</td>
</tr>
</tbody>
</table>

* Note: Divider of activity original duration to get the average time to reply RFI. For example, assuming that the activity A has 50 days duration and the divider is 5, time to reply RFI will be 10 (50/5).

The simulation results demonstrate that shortening a required time for labor hiring and RFI reply contributes to enhancing the project schedule and cost performance. In particular, RFI reply time greatly affects the project performance. As shown in Figure 49, shortening RFI reply time by half could facilitate the project progress by 12% and reduce the project costs by 10%. In contrast, once RFI reply time is doubled, duration and costs are increased by 29% and 24% respectively. These simulation results imply that for this case project, coordination among the project functions is crucial to the success of the project. Consequently, the decision-making process in design and construction should be shortened and information flow among project functions should be streamlined to assist in reducing the decision-making time.
Figure 47-a: Hiring Time vs Project Progress
Figure 47-b: Hiring Time vs Work Hours
Graph for Project Progress

Figure 48-a: RFI Reply Time vs Project Progress
Figure 48-b: RFI Reply Time vs Work Hours
Time Variables vs Project Performance

Figure 49: Time Delays vs Project Performance
Policy Recommendations

Findings obtained from the sensitivity studies thus far have their policy implications, narrowing downing desirable sets of the project components. First, 100% flexible labor control policy was found to be most efficient in terms of schedule reduction. In addition, it was possible to find the most desirable buffering ratios for the activities of the case project, which are summarized in Table 21. The simulation results also demonstrated that shortening workforce control time and RFI reply time could contribute to significant schedule reduction in carrying out the case project. Having obtained the desirable project settings, this chapter examines their effectiveness when they are combined, by simulating them in a comprehensive manner. Table 21 summarizes the project settings, with which simulations have been done and compares the simulation results with the base case.

Table 21: Simulation of Policy Recommendations

<table>
<thead>
<tr>
<th>Cases</th>
<th>Flexibility in labor Control</th>
<th>Reliability Buffering*</th>
<th>Time Delays</th>
<th>Completion Time (Days)</th>
<th>Labor Hours (worker*hour)</th>
<th>Output Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Time to Increase Workforce (days)</td>
<td>Time to Reply RFI*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>N/A</td>
<td>None</td>
<td>N/A</td>
<td>391</td>
<td>N/A</td>
<td>CPM*</td>
</tr>
<tr>
<td></td>
<td>100%</td>
<td></td>
<td>7</td>
<td>3</td>
<td>559</td>
<td>1.305M</td>
</tr>
<tr>
<td>2</td>
<td>100%</td>
<td>Subject to Individual Activity Characteristics (see Table 00)</td>
<td>3</td>
<td>9</td>
<td>388</td>
<td>1.055M</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>RCMD</td>
</tr>
</tbody>
</table>

* Note 1. Buffer Size: Fraction of Taken-off Contingency Buffer (20% of Activity Original Duration)
2. Divider of activity original duration to get the average time to reply RFI.
3. Based on CPM-related data only
Figure 50-a: Recommended Policies vs Project Progress
Figure 50-b: Recommended Policies vs Work Hours
The simulation results demonstrate that applying the desirable project settings to the case project could significantly enhance the project schedule and cost performance (35% schedule reduction and 30% cost down compared to the base case). The simulated duration is also shorter than CPM-based duration of the project. Of course, the simulation results have been obtained, assuming that a significant time reduction in worker hiring and RFI reply was achieved. In practice, it is not easy to achieve such amount of time reduction, since there are many other factors that govern the processes. However, the important thing is that by utilizing DPM-generated results it is possible to find which activity will be the bottleneck of a project and where to focus during the project development. In fact, as discussed before, this case project experienced a lot of design changes, which delayed the whole development process. Consequently, once RFI reply time can be decreased by increasing the level of coordination among the project functions, it is possible to avoid or significantly reduce the impact of those changes, as quantified by DPM simulations.

In addition, the sensitivity studies where the recommended policies have been simulated under uncertain conditions imply that DPM-generated policies are robust against uncertainties. The simulation results presented in Figure 51 and Figure 52 have been obtained by simulating the case project two hundreds time with the conditions given in Table 22.

<table>
<thead>
<tr>
<th>Test #</th>
<th>Simulation Settings</th>
<th>Simulation Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Flexibility in labor Control</td>
<td>Time to Reply RFI*</td>
</tr>
<tr>
<td>1</td>
<td>Applied 100%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Policy Not Applied</td>
<td>Not Applied</td>
</tr>
<tr>
<td>2</td>
<td>Policy Applied 100%</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>Policy Not Applied</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* Note 1. Divider of activity original duration to get the average time to reply RFI.
2. Random Number Distribution: Standard Normal Distribution
3. Standard Deviation: 0.05
4. Mean: 1 in both Test 1 and Test 2
5. Number of Simulation: 200
6. Simulation settings that are not specified are the same as the base case.
The first sensitivity test has been done to examine the policy's robustness to changes in work quality. During the simulation, a variation factor is randomly generated ranging from 75% to 125%, which is weighted to the initially given work quality of activities. As a result of the multiple simulations, it turned out that DPM-generated policies could help reduce variations in the estimated project duration under uncertain work quality conditions. As shown in Figure 51, by applying reliability buffering, it was possible to reduce the variation of project duration in 95% confidence boundary from 100 days to 40 days as well as enhancing the project schedule and cost performance (in Figure 51, the distribution of project duration frequency in the buffering case is narrower and taller, compared to that in the no buffering case).

In addition, the robustness of the DPM-generated policies have been also tested varying input durations, which is normal in PERT. Having simulated the case project with a range of input durations, the simulation results were compared to those calculated using PERT. This sensitivity test implies that DPM-generated policies could also help make the project completion less sensitive to variations in input durations by reducing the variation of project duration in 95% confidence boundary from 100 days to 80 days. In addition, as shown in Figure 52, when the DPM-generated policies are applied the distribution of project duration frequency becomes skewed to the left, which suggests there is a higher possibility to achieve an early finish with DPM plans than in PERT.

In conclusion, although the obtained simulation results can vary depending on project settings, they well demonstrate how DPM can contribute to enhancing the project performance in a real world setting by providing robust plans and policy guidelines. Additionally, the simulation results also imply that DPM-based construction policies including reliability buffering can be more effective, when combined with other managerial efforts such as reducing a process time and increasing the level of coordination among project functions.
R. Buffering Applied

- Mean: 402 days
- CPM duration: 391 days
- Duration Variation in 95% Confidence Boundary: 391 - 400 days

R. Buffering Not Applied

- Mean: 495 days
- CPM duration: 400 days
- Duration Variation in 95% Confidence Boundary: 400 - 409 days

* Note 1. RFI time reduction and 100% flexible labor control are applied to both cases
2. Reliability of project activities ranging 0.75 to 1.25 is randomly generated with 5% of standard deviation.

Figure 51: Distribution of Project Completion Times vs. Variations in Work Quality
Duration Variations in PERT

- Mean: 391 days
- 95% Confidence Boundary: 100 days

Duration Variations in DPM

- Mean: 401 days
- 95% Confidence Boundary: 80 days

* Note 1. RFI time reduction and 100% flexible labor control are applied to both cases
2. Variation factor for activity duration ranges from 0.75 to 1.25.

Figure 52: Distribution of Project Completion Times vs. Variations in Input Durations
This chapter scrutinized the performance of DPM as a planning and control tool, and its applicability in real world settings with three application examples. The building construction example demonstrated that the use of DPM could help the project manager establish construction policies, focusing on DPM’s capability to assess the time and cost consequences of fast-tracking. In the second application example, the effectiveness of reliability buffering that has been suggested in this thesis was examined based on the understanding the change impact. In addition, the DPM system behaviors showed that reliability buffering could help enhance the project performance by reducing change impacts and each construction characteristic had different effect on effective buffering. Lastly, the application example of DPM to bridge construction projects demonstrated the applicability of DPM in real world settings, focusing on the role of reliability buffering in concurrent construction and the provision of project policies.
CHAPTER 8

CONCLUSIONS

Concurrent construction has been widely used for modern construction projects, as a method to shorten time-to-market. In fact, its time saving feature has placed it as a possible alternative to the traditional more sequential method. Along with its potential benefits, however, concurrent construction also has greater potential to impact the project development process than the traditional method. Moreover, as construction projects continue to increase in size and complexity, the planning and control of concurrent construction becomes more difficult. These industrial trends and challenges in concurrent construction, together with increased understanding on dynamics and complexities of construction, increased the demand for a more efficient planning and control method.

As an effort to address some of these challenging issues, this thesis presented Dynamic Planning and Control Methodology (DPM). All of the concepts and logics of DPM, which are user-defined modeling approach, consideration of feedbacks, capturing construction dynamics, reducing sensitivity to changes, and smart system have been derived from closer observations of construction processes and practices thus far, and they have been elaborated, taking consideration into functional requirements to achieve DPM target goals. These fundamental concepts and logics have been materialized by incorporating reliability buffering contents and concurrent engineering principles into
system dynamics models as well as schedule networking concepts of CPM, PDM, PERT, GERT, and SLAM. Then, the performance of DPM as a planning and control tool and its applicability in real world settings were examined with three application examples.

8.1 POTENTIAL IMPACT

The research results obtained thus far demonstrated that DPM would help prepare a more robust construction plan against uncertainties and provide policy guidelines for the control of construction activities. Problems encountered in the planning and control of construction projects are fundamentally dynamic. However, they have been treated statically with a partial view on a project [Lyneis, 1999]. When problems occur during construction, managers tend to pay attention to the problems themselves rather than scrutinizing the construction system structure, in which they occur. As a result, chronic managerial problems persist in carrying out construction projects and construction schedule is continuously updated with a time delay in a monotonic way. In this context, DPM could help effectively deal with unplanned and/or indirect events that might occur during construction, by taking into consideration the context in which a construction project is being developed.

In addition, DPM could also help utilize learning from one project for others. In fact, learning has rarely accumulated across construction projects. This is, in part, because construction is process-based work that is performed on an unfixed place by a temporary alliance among multiple organizations [Slaughter, 1999]. However, it is also true that the lack of learning in construction is attributed to the lack of learning mechanism in the traditional network-based planning tools. The traditional tools are based on the deterministic approach, which is not suitable to controlling the dynamic state of construction and they do not have a mechanism to formalize information obtained from construction monitoring or elicit learning of construction crew. In order to address these challenges, DPM allows the model structures to be tuned up based on information
obtained from the actual project performance, which makes it possible to embed one’s knowledge and learning from a project into the planning and control system.

All of these features of DPM would be beneficial for the planning and control of construction projects, in particular when they are performed concurrently and involve higher complexity and uncertainties.

### 8.2 APPLICABILITY

The development of DPM would also contribute to increasing the applicability of the simulation approach in project planning and control. In fact, simulation capability has been seen as an opposite concept to applicability. Partly due to this recognition, the previous research efforts to increase the applicability of the simulation approach have mainly focused on the development of user-friendly graphic representations of simulation components. For example, SLAM [Pritsker, 1994] and STROBOSCOPE [Martinez, 1996] provided an integrated simulation environment, in which users can model project development processes using graphic representations of simulation components. However, the use of those tools still requires a lot of modeling experience, which makes it difficult for users (presumably, construction managers or engineers) without having modeling skills to apply them to construction.

In contrast, DPM attempts to increase applicability, while keeping required reality in representation by introducing the user-defined modeling approach, in which simulation models are pre-structured and users can define the model settings, when they use DPM. In fact, the success of the user-defined modeling depends on how well pre-structured models can represent construction processes and dynamics, and how reliable the simulation output of the models is. In order to deal with these challenging issues, DPM identified the most influential construction dynamics and characteristics, which were converted into the generic parameters and structures in system dynamics models. In
addition, by incorporating the fundamental concepts and principles of network-based
tools into the system dynamics models, DPM has the functionality of the traditional
planning tools as well as having simulation capability. As a result, depending on input
variables, and planning and control purposes, the simulation output may start with CPM
and evolve into DPM as data for parameters are provided.

In order to examine its applicability, DPM has been applied to the bridge
construction planning of Route 3 North Project in Massachusetts. The results of the case
studies demonstrated that DPM could contribute to enhancing the project performance in
a real world setting by providing robust plans and policy guidelines. However, it turned
out during the case studies that in order to appreciate construction dynamics, activities
need to be appropriately aggregated. Since DPM models are pre-structured with the
consideration of time delays involved in construction processes, DPM generates less
meaningful simulation results, when activities have too short durations (i.e., one or two
days) or an inappropriate aggregation of activities offsets the time-based construction
dynamics. In addition, there were some activity types that DPM could not effectively
represent. Examples of such activity types include issuing permit, noticing completion to
owner, and right of way acquisition. These types of activities have totally different
characteristics from those of the generic process model in DPM, which focuses on change
and rework iterations caused by work quality. For this reason, they are treated as an
exceptional type in the current DPM system, not having any strategic implications.
However, more fundamentally, the current DPM model needs to be further refined to
effectively represent those types of activities as well.

8.3 FURTHER DEVELOPMENT

Although the research work presented thus far hold a good promise by showing that
the use of DPM can help enhance planning and control capabilities, the current DPM
needs to be further refined and developed in order to get its target functionality.
Figure 53: Further Development of DPM

- Excluded
  - Resource Constraints (Cash Flow, Material & Equipment)
  - Work Conditions (Weather, Seasonal Effect & Site Conditions)

- Exogenous
  - Multiple Project Development Sequence
  - Stage Changes
  - Performance Profiles (Cost, Safety & Environmental Impact)

- Endogenous
  - Activity Characteristics
  - Project Progress
  - Actual Productivity
  - Actual Time Delays in Processes
  - Workforce Allocation & Utilization
  - Actual Work Quality

- Network-based Tools
  - Applicability
  - Flexibility
  - Reality in Representation
  - Ability to Deal with Dynamic Complexities

- Modeler-defined Simulation

- DPM

- Reality in Representation
  - Ability to Deal with Dynamic Complexities
First, the modeling factors that have been tentatively excluded in the current DPM model boundary need to be incorporated into the DPM system. For example, while the current DPM generates activity performance profiles in terms of productivity and quality, the further developed DPM needs to address some other aspects of activity performance such as cost, safety, and environmental impact. The further research also needs to focus on resource constraints caused by cash flow, and the availability of material and construction equipment, since they also play an important role in determining construction performance as well as workforce resource. Secondly, as discussed in Chapter 8.2, there are some activity types that the current DPM cannot effectively represent. For this reason, the generic process model structure needs to be further developed in a way that more diversified construction dynamics can be addressed. To do this, the inherent feedback processes and influential performance determinants in those activity types need to be identified based on closer observations. Thirdly, the functionality of reliability buffering can be further extended. Reliability buffering in the current DPM focuses on absorbing the upstream impact by utilizing implicitly or explicitly imbedded contingency factors. Once the functionality is extended to determining an overlapping degree given certain conditions, it would be able to more aggressively increase the concurrency of activities, keeping the side effect minimized. Lastly, in the current system, due to inflexibility in controlling activity relationships from outside the system, the DPM models are pre-structured in the way that activities can have all possible relationships, resulting in slowness in simulation. However, once increasing the role of application languages in simulation, it would be possible to significantly shorten the simulation time.
REFERENCE

Eppinger et al. (1993), A model-based framework to overlap product development activities, MIT Sloan School of Management, Working thesis 3635-93.


Project Management Institute (1987), Project Management Body of Knowledge of the Project Management Institute, Drexll Hill.


APPENDIX I
SUMMARY OF LITERATURE REVIEW
To deal with a matching problem that typifies fast-track process-plant projects, the author developed a model of material-management process for pipe-spool installation, using the STROBOSCOPE computer system (a discrete-event simulation modeling tool, Martinez 1996). The model is also used to verify the use of a lean construction technique known as pull-driven scheduling.

Lean Construction

Matching problems mainly caused by delays and process uncertainties may hamper construction productivity. To handle those problems, the lean production concept has been introduced into construction. The main objective of the lean production is to maximize the value of a product, while minimizing losses during production. With respect to the application of the lean concept to construction, Howell et al. (1993) showed that the dependencies and worker idle time could be alleviated by increasing buffers of materials. Ballard and Howell (1997) introduced the Last Planner to shield construction workers from process uncertainties.

Pull-Driven Process Management

The pull-driven approach originates from the same vein as lean production. The traditional, ‘push-driven’ approach aims at adhering to the resulting schedule, assuming that all resources needed to perform an activity are available at the activity’s early-start time. Resource leveling or allocation algorithms adopted in the traditional planning tools such as CPM and PERT ‘may yield some adjustments to the early start schedule, but
upon project execution, activities are expected to start their earliest possible date so as not to delay succeeding activities or the project as a whole.’ Such approach to construction scheduling may not be effective in dealing with process uncertainties, leading to less-than-optimal project performance.

In the meantime, a ‘pull-driven’ approach aims at ‘producing finished products as optimally as possible in terms of quality, time, and cost, so as to satisfy customer demand.’ ‘The pull means that resources must be selectively drawn from queues.’ To implement a pull-driven approach, resources have to be controlled selectively, for which resources will get priority over others in the same queue. This way will make the downstream resources not unduly await their match and be in process for any time longer than needed.

**Process Modeling**

1. **STROBOSCOPE**

   One major feature of the STROBOSCOPE computer system is that ‘resources can be characterized and individually tracked as they reside in various network nodes during a simulation run’. For resource allocation simulation, the STROBOSCOPE has the criteria applied in selecting resources for withdrawal from the queue such as 1) First-in first-out or last-in first-out, 2) First-in-order based on a property of resources in a single queue, 3) Best match based on properties of resources in multiple queues, all preceding a single activity, and 4) Random.

2. **Pipe-Spool Process Model**

   Using the STROBOSCOPE, the pipe-spool installation process is modeled in order to describe and experiment with alternative planning sequences. Table 1
summarizes the functionality of the STROBOSCOPE symbols which are used in the modeling work. Figure 1 depicts a model structure for a typical pipe-spool installation process. This model structure aims at analyzing the impact of coordination planning on resource management and benefits of 'pull' over 'push' under uncertainty.

Table 1: STROBOSCOPE Symbols

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>NAME</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue</td>
<td>Transport</td>
<td>Describes a certain type of work to be done, or a delay, of a known (probabilistic) duration from start to finish. May require a single resource or no resource at all.</td>
</tr>
<tr>
<td>Combi</td>
<td>Fabricate</td>
<td>Like a normal, describes a certain type of work to be done, or a delay, of a known (probabilistic) duration from start to finish. Unlike a normal, requires several resources in combination for its performance and draws what is needed from the queue(s) that precede it.</td>
</tr>
<tr>
<td>Normal</td>
<td>Compilation</td>
<td>Acts as a counter up to n (n is an integer value specified with the node); after n resources have been released into the consolidator, the consolidated set will be released from it.</td>
</tr>
<tr>
<td>Normal</td>
<td>Processing</td>
<td>Shows flow logic. Should be labeled to meaningfully describe the resources that flow through it. If the link emanates from a queue, a DRAWORDER may be specified to sequence resources being drawn from the queue.</td>
</tr>
<tr>
<td>Normal</td>
<td>Assembler</td>
<td>Shows that 2 or more resources are being assembled into a single unit resource which is of the compound (a special kind of characterized) resource type.</td>
</tr>
</tbody>
</table>
3. Implementation and Simulation

Based on the model structure depicted in Figure 1, one deterministic and three probabilistic models were implemented. While the deterministic model illustrates the project progress when every system element is assumed perfectly coordinated and synchronized, the probabilistic models simulate the project progress with consideration of all uncertainties in duration and likelihood of rework. With different input
variables including production resources and duration, the models simulate changes in pipe-spool buffer size, productivity of construction crew, and project duration. In particular, simulation of each model focuses on monitoring how buffer sizes and progress of activities vary.

Conclusions

The research results obtained from this paper imply that the lean production ‘pull’ technique can be useful in improving construction performance, in particular, of fast-tracked projects, which are prone to uncertainties. In addition, ‘the pull technique suggests that real-time feedback from construction be used to drive the sequencing of off-site work and vice versa’. As for simulation approach to scheduling, the author argues that ‘it can help the decision maker understand the system’s behavior and gauge the impact pull links may have and using the simulated data, a cost-benefit analysis can be performed prior to establishment of those link.’

Review

Throughout the paper, needs for simulation approach to project planning and control are logically represented. The STROBOSCOPE system has its strength such that modeling units that flow into or through queue (stock in system dynamics) can be characterized and individually tracked during a simulation run. For this reason, it can be suitable for the resource management of the project that requires assembly of unique parts. However, it is questioned how effectively the following issues, which need to be considered in project control and planning can be handled using the STROBOSCOPE system.

1) Human factors in production rate
2) Information flow in monitoring, report, planning, and other decision-making processes
3) Non-linear dependency relationships between activities
Simulation-based scheduling methods have currently emerged as an alternative to the traditional CPM method due to their modeling versatility and effectiveness. ‘Very few techniques, however, allow activity duration and sequencing to be defined in terms of the dynamic information that become available as a project evolves.’ The author ‘presents a program for probabilistic CPM scheduling designed as an add-on to the STROBOSCOPE simulation system.’

Why Simulation-Based method?

Although a CPM has been most widely used to plan and control construction project to date, it is not considered as an accurate representation of actual construction activities. This is mainly because the CPM is based on the assumption such that ‘activities have a fixed duration and the duration is known at the beginning of the project’. For this reason, the authors point out, in reality the CPM network has to be often updated to reflect the actuals. ‘The effectiveness of planning at the project level can be substantially enhanced if the plan can be formulated and evaluated more realistically ahead of time.’ To this, the uncertainty associated with activities needs to be recognized as a function of dynamic state of the project.

To date, a substantial amount of research has been made to meet these challenges. Although the results of those research efforts including PERT, GERT (Philips & Hog 1986), VERT (Moeller & Digman 1981), MUD (Carr 1971) and DYNASTRA (Morua Padilla 1986) have enhanced planning and control capabilities to some extents, the authors argue, ‘none of them have the necessary flexibility and power to model uncertainty in the duration of activities as a true function of the state of the project,'
nor can they model the underlying process-level operations through concurrent simulation.'

**STROBOSCOPE**

The STROBOSCOPE (State and Resource Based Simulation of Construction ProcEsses) is a 'simulation programming language based on activity cycle diagram and was developed especially for modeling construction operations.' Meanwhile, 'a STROBOSCOPE add-on is a 32-bit MS Windows Dynamic Link Library that extends the STROBOSCOPE language with new statements, functions, and variables.'

**CPM Add-On**

The CPM add-on provides a good example on how the function and usage of the STROBOSCOPE can be extended. As represented in Table 1, the CPM add-on adds probabilistic functions to the traditional CPM method.

Table 1: Statement Registered by the CPM Add-on

<table>
<thead>
<tr>
<th>Statements</th>
<th>Arguments</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPMACTIVITY</td>
<td>1) Activity Name 2) Duration Expression</td>
<td>Defines a CPM activity with the provided name and duration sampled using the provided expression.</td>
</tr>
<tr>
<td>PRECEDENCE</td>
<td>1) Predecessor Name 2) Successor Name</td>
<td>Indicates that the successor cannot start until the predecessor is finished.</td>
</tr>
<tr>
<td>CPMREPLICATE</td>
<td>1) Number of Replications</td>
<td>Simulates the CPM network the number of times indicated and produces a report as described in the CPMREPORT statement.</td>
</tr>
<tr>
<td>DOCPM</td>
<td></td>
<td>Performs a single forward and backward CPM pass</td>
</tr>
<tr>
<td>CPMREPORT</td>
<td></td>
<td>Prints the 90% Confidence Interval on project duration and a report showing the average duration, average early and late dates, average floats and criticality for each CPM activity.</td>
</tr>
</tbody>
</table>
For example, in the Table 1, the statement ‘CPM ACTIVITY’ has a multiple variable, which can be categorized into the argument 1) Activity name and the argument 2) Duration expression. The below expression is interpreted by the STROBOSCOPE as follows:

\[ CPMACTIVITY \text{ ExcSt14To17 Pertpg}[0.9,1,1,1.4]+\text{RainfallLst48hrs}*0.3 \]

‘The duration of the ExcSt14To17 activity defined above would be determined by sampling from a Beta distribution with 5\textsuperscript{th} percentile 0.9 days, Mode 1.1 days, and 95\textsuperscript{th} percentile 1.4 days; 0.3 days for each centimeter of rainfall accumulated during 48 hours prior to start of the activity would be added to the sampled value.’

**Conclusions**

The CPM add-on is expected to be a starting point for more powerful ‘project-level’ analysis tool with increased flexibility in incorporating probabilities into construction duration and providing dynamism in the state of a project.

**Review**

The usefulness of the STROBOSCOPE add-on function including CPM add-on lies in enriching the STROBOSCOPE’s built-in functions and extending the application of STROBOSCOPE to specific area. For example, it appears to be suitable as a tool for validating a heuristic approach to optimization. However, despite the extended application area achieved through the add-on function, the STROBOSCOPE is still not flexible enough to be applied for general purposes such as process-level scheduling.
Difficulties in the planning and control of construction process are mainly attributed to the non-steadiness of construction. For this reason, to enhance the productivity and resource efficiency of construction operations, the impact of various factors involved in construction needs to be analyzed and considered in the planning and control of construction. In this paper, the author presents a framework for the identification of sources of process delays, focusing on the relationship between storage limitations of flow units and productivity.

Digital Simulation Technique

In an effort to deal with the non-steady feature of construction processes that are constantly interrupted by breaks and delays, digital simulation techniques have been introduced into construction processes. Among them are the CYCLONE (CYCLi Operations NEtwork, Halpin 1973) and INSIGHT (Interactive Simulation using GrapHics Techniques, Paulson et al. 1984). They focus on the analysis of resource idleness of unbalanced systems, which cause work slowdowns or even stoppages of work tasks and eventually impact the entire production processes.

Construction Process As System

Construction process can be seen as a system (Blanchard et al. 1985) having input comprised of material, tools, equipment, labor, management, time, and conditions, and output. Meanwhile, the effectiveness of a system mainly depends on the efficient utilization of input. But, the traditional planning tools lack capabilities to deal with probabilistic elements. Moreover, the production rate of complex systems
cannot be properly described as a function, since the relationship between input variables and output rate is usually unknown.' Based on this recognition, the author argues that simulation techniques are most suitable for handling non-steady construction processes that are a complex system. With respect to this system view, the importance of the systematic monitoring, in particular, of resource effectiveness has been emphasized by several researchers (Adrian 1976, Carr 1974, Paulson 1985). Their researches commonly focus on 'increasing the productivity of repetitive processes by eliminating idle times of resources.'

**Productivity Transients of Systems**

In a system, the productivity pattern is generally divided into three phases: 1) the start-up phase during system configuration, 2) the transient phase during system stabilization, and 3) the termination phase. In particular, transients are caused mainly by resource availability and changing conditions in environment, construction site, work place, and experience. With respect to changing conditions, the author argues that 'most often they do not have an equal impact on work tasks of a process; some of the work tasks are more sensitive than others to a certain aspect of the changed condition.'

![Productivity Measurement Curve](image)

*Figure 1: Productivity Measurement Curve*
Modeling Non-Steady Processes

1. CYCLONE

The CYCLONE has the following four modeling elements (Halpin 1976):
- NORMAL: a work task element that models a logically unconstrained activity state
- COMBI: a work task that requires more than one resource to perform
- QUEUE: waiting or storage area for flow units
- ARROW: direction of flowing units within the system

![NORMAL COMBI QUEUE ARROW](image)

Figure 1: Productivity Measurement Curve

The four elements above are used to build a stock and flow structure of a model. The CYCLONE can constraint the number of units that are in transit (flow) or in queue (stock). However, although this method allows the limitation of a specific level of units in stock, it does not provide the capability to flexibly control a queue. The control of a queue is done only when the actual level reaches a specific level of units in stock. For example, when the stock level drops to the minimally required level, new units flow into the queue and when the stock level reaches its maximum, the queue releases units accumulated in the queue.
2. Concrete Placing Model

Using the CYCLONE, concrete placing process is modeled in order to describe and experiment with alternative planning sequences. The sensitivity study focuses on the analysis of the impact of changes in concrete truck waiting time, vibrator deflection rate, and concrete pile queue on the productivity of the system.

Figure 2: CYCLONE Model of Concrete Placing
Conclusions

The CYCLONE-based simulation helps systematically identify factors that cause transient productivity interruptions in construction and shows the possibility to incorporate the randomness of construction on a higher level project planning and control. However, as argued by various researchers (Paulson 1984), the unavailability of statistical process data is still challenging the simulation approach to the project planning and control.

Review

Throughout the paper needs for the simulation approach to the construction planning and control are well addressed. Also, a closer observation on construction processes provides a solid theoretical basis for the simulation approach. However, in this paper, the application of the CYCLONE technique is limited to the analysis of productivity transients. Moreover, it is questioned that the CYCLONE technique can effectively incorporate all of determinant factors into a model. In fact, the concrete placing model presented in this paper does not consider the impact of factors related to productivity transients.
One of the principal benefits of simulation approach to the project planning and control is that one can compare the effectiveness of alternative construction methods before physical execution. Furthermore, the authors argue, ‘the efficiency and effectiveness of such comparison can be greatly improved by the use of ‘matched pairs’, a variance reduction technique based on dedicated and fully synchronized random number stream.’ This ‘matched pair’ technique can improve statistical efficiency in finding a favorable alternative based on simulation results, by reducing the number of necessary simulation runs on a more logical and equitable basis. In this paper, the authors apply this technique to selecting a cheaper tunneling method between the conventional method and the New Austrian Tunneling Method (NATM) based on the simulation results of a STROBOSCOPE model.

**Matched Pairs**

To evaluate and compare the performance of alternatives with a simulation approach, the following procedures are common: Building a simulation model, Conducting a number of simulation runs, and Comparing alternatives based on the resulting average of their performance. However, this approach, the authors argue, may have a problem. Although the alternatives should be simulated and compared under the same simulation conditions for accuracy, the simulation runs for one alternative are often done under different conditions from those for other competing alternatives.

The key to meet this challenge is ‘to ensure that the random numbers used by each alternative follow similar patterns because all uncertainty in a simulation model is determined by the random numbers.’ While the simplest way to do this is ‘to
start the corresponding simulation replications for all alternative methods using the same set of random number seeds,' such approach does not always work well. In other words, even if the simulations for alternative A and B have been started with the same random number, the simulation results can be totally different depending on the model structures. To settle this potential problem, which can produce unreasonable simulation results, a ‘matched pairs’ technique has been suggested by the authors. The backbone concept of the method is ‘to dedicate a stream of random numbers to each uncertain variable that is common to all alternatives.’

Detailed guidelines for the implementation of the method are presented together with an application example.

**STROBOSCOPE Simulation Model**

Using the STROBOSCOPE, tunneling processes are modeled. Figure 1 shows construction activities and their relationships for both the conventional method and NATM. The STROBOSCOPE network illustrated in Figure 1 has similar appearance and functions to those in CYCLONE (Ioannou 1989). In addition to the basic model structure illustrated in Figure 1, the underlying logic including resource attributes, link attributes, and other modeling entities is described using the graphical user interface.

![STROBOSCOPE Network for Tunnel Construction](image_url)

Figure 1: STROBOSCOPE Network for Tunnel Construction
Simulation Experiments

Using the tunneling process model, simulation experiments are made to test the effectiveness of the ‘matched pairs’ method. The simulation results show that the method significantly increases the probability of identifying and selecting a more favorable tunneling method based on a single run from 55% to 96% and ‘the 95% confidence interval for the true cost difference given by 4,000 independent runs can be obtained by performing only seven replications using the matched pairs.’

Conclusions

‘To perform a meaningful comparison of construction alternatives, it is imperative to know when to continue and when to stop performing additional simulation runs.’ In this regard, the matched pairs method provides a very useful tool from both statistical and logical point of view. Using the method makes it possible to conduct simulation experiments so that ‘chance impacts all alternatives’ in a similar and equitable manner. In addition, it is also possible to reduce the number of necessary simulation runs significantly and to make alternatives be compared under the same conditions.

Review

The authors’ arguments with respect to the effectiveness of the matched pairs provide a good insight to those who try to evaluate and compare alternatives using the simulation methods. Narrowing down alternatives with less simulation runs and making the alternatives be compared under the same probabilistic conditions will be beneficial, especially when either the number or the order of variables in a simulation model may vary depending on simulation scenarios.
Assuming that most simulation models in construction are stochastic models, the authors emphasize the importance of appropriate statistical tools that can properly structure input data, analyze simulation results, and validate the results and model structures. The provision of statistical techniques, which are applicable to repetitive construction processes supports their arguments.

**Construction Process Simulation**

While simulation can be divided largely into deterministic and stochastic depending on its uncertainty content (Wilson 1984, Kelton 1986), stochastic simulation is more often used to model construction operations due to the non-steady nature of construction. Meanwhile, ‘the majority of problem in simulation, in the context of construction, stems from the lack of a consistent approach that clearly defines all the steps and aspects of simulation.’ This roots in the complexity of simulation techniques. Although progress has been made in introducing simulation methodologies suitable for construction, there still remains deficiency in the area of simulation experimentation. For the simulation experiment to be fruitful, the authors argue, the following needs to be ensured: ‘1) proper input in the form of statistical models for work task duration, 2) proper analysis of output, 3) validation and verification of the model, and 4) reduction of the number of necessary runs through variance-reduction techniques.’

**Input Date Modeling**

To identify appropriate distribution, techniques found to be useful in construction applications are presented. Estimate techniques are also discussed for properly
estimating parameters of given distribution. With respect to estimation method, the authors suggest that the simulator use ‘all fitting methods available within the software being used and select the parameters that produce the best fit.’ Once having estimates the parameters of a distribution, one needs to test the goodness of fit by comparing the fitted distribution to the empirical distribution. For this, the authors present various techniques and recommend that the simulator use a flexible family of distribution provided that the simulation software supports various generations from such families.

Analysis of Simulation Output

A typical approach to the analysis of simulation result begins with determining whether the simulation is deterministic or stochastic. When the simulation is deterministic, one simulation run is sufficient. However, in the case of a stochastic simulation, whether the simulation reflects a static, transient, or steady state needs to be defined, because the basis for decision-making can vary depending on the state of simulation. Most construction operations may be categorized into a transient simulation, based on the following definition of transient simulation by Wilson (1984): ‘a simulation is transient if the modeling objective is to estimate parameters of a time-dependent output distribution over some portion of a finite time horizon for a given set of initial conditions.’

Example Application

A simulation of an earth-moving operation developed by Cottrell (1989) is used as an example application of the input data modeling and output analysis. The CYCLONE model structure is illustrated in Figure 1.

Model Validation

Model validation is most often done by comparing the simulation result with the following: ‘1) results obtained from theoretical models, 2) results obtained
from analytical techniques, and 3) historical or published data.’ In particular, the simulation of construction process is ‘often repetitive and cyclic in nature, and construction is carried out with a relatively long life span.’ Thus, it is possible to validate the model by comparing the simulation results with the actual data obtained from a closer observation of construction processes.

Figure 1: CYCLONE Model of Earth-Moving Operation

Conclusions and Review

Techniques and modeling procedures reviewed in this paper can be useful in a stochastic simulation experiment. And, the authors’ approach to the simulation method gives a lot of implications. However, the techniques and modeling procedures do not appear to provide a comprehensive tool that enables the simulator to answer to ‘what-if’ questions.

205
This paper describes the background of a computer simulation approach to the construction planning and control and the system structure developed by the authors. The system structure consists of three parts: videotape data collection, computer-based simulation modeling, and stochastic analysis of the simulated data.

**Why Simulation?**

Construction processes involve complex interactions. 'Imbalances and interference among these entities cause inefficiencies and uncertainties that can adversely affect the costs and schedules of the operations'. In this regard, great improvement can be made if the construction process can be simulated and based on the simulation the performance can be optimized, before committing actual resources to the processes. And, an analytic tool that can help find and correct problems with a diagnostic capability would be beneficial.

**Simulation Methods**

<table>
<thead>
<tr>
<th>Description</th>
<th>CYCLONE (CYCLI Operations Network)</th>
<th>MICROCYCLONE</th>
<th>INSIGHT (Interactive Simulation using Graphics Techniques)</th>
</tr>
</thead>
</table>
| Key Features | - First simulation tool for construction  
- Stock & flow  
- With mainframe computer  
- ‘Batch Processing’ | - For large, complex model  
- For Microcomputer  
- ‘Batch Processing’ | - Interactive with easier user interface  
- Extended Modeling capability (resource costs optimization, easy data transfer etc.)  
- For Microcomputer |
Reasons for Lack of Popularity So Far

The applications of the existing simulation method have been limited for the following reasons: 1) the practical and theoretical limitation of the simulation methods; 2) the difficulties in obtaining quantitative production data; and 3) the lack of convenience of and economical access to computer.

A INSIGHT Model

A concrete-placing model illustrated in Figure 1 is used to show the application of a INSIGHT model. Using a built-in interactive editor (text mode), one can input 'general project information such as activity descriptions and logical relationships, resource amount, descriptions, costs, and initial positions, and probabilistic activity duration data.' Based on the simulation results, one can get a output report including 1) behavior of queues preceding activities such as the average number of resources in the queues, the percent of time that queues were occupied, and the average wait time for resources in queues, 2) production costs per hour and per shift, and 3) behaviors of activities within the model such as the percent of time the activity are busy and their duration statistics.

Conclusions and Review

In terms of its ability to input project data interactively using a built-in editor, the INSIGHT appears to have advantage over many of simulation methods available before the INSIGHT was developed. However, it is not believed that the INSIGHT runs interactively, which is argued by the authors, because it does not have a function to actively control the simulation during a simulation run. In addition, the system structure suggested by the authors shows a direction of future research on the project planning and control using project information integration.
Figure 1: INSIGHT Model of Concrete Placing
The efficient coordination and management of construction processes is one of the most critical factors to determining the success of a construction project. However, it is more important in the case of plant piping construction. And, the efficient coordination and management of construction processes begins with the proper definition of work packages. This paper highlights major decision variables that greatly influence the work-packaging process in petrochemical plant construction and presents a work-packaging process model using the data flow diagram (DFD).

**Work Packaging**

Work packaging is not just to break a construction job into a set of smaller work units. 'It is a planning activity that demands a substantial understanding about the job and the conditions in which the work is to be performed.' Thus, for work packages to be an effective units of managing a job, a project planner needs to 'understand, evaluate, and resolve various conditions that are either present or anticipated for a construction job.'

**DFD (Data Flow Diagram)**

The DFD models the interactions of functional processes and data flows among them. It has been used as modeling tools before automating any process due to its functions to graphically identify possible conflicts and redundancies in information flow. The DFD has the following modeling elements (Yourdon 1989):

- Circle: transformation of data from the input stream to the out stream (Process)
- Arrow: flow of data (Flow)
- Opened Rectangular: data storage (Store)
- Rectangular: source or destination of data (Terminator)
The modeling process is first divided into two phases. That is, long term and short term. Each produces level 1 piping plan for construction. These level 1 plans are further detailed with piping work-complexity evaluation process (level 3) and piping work-density evaluation process (level 3).

The work-complexity evaluation process provides a formal data-collection form, with which planners can evaluate the piping work complexity. In the complexity evaluation process, pipelines of selected piping block are identified and quantified by referencing drawings and related documents, which in turn forms complexity lists (refer to Figure 2). The lists include the necessary information for complexity evaluation, such as pipe size, class, referenced P&ID number, material type etc. Finally, 'work complexity is represented as the total worker hours required to build the selected block.'

Figure 2: Example of Piping-Complexity Lists
The work-density evaluation process is followed by the complexity evaluation process. Figure 3 illustrates the overall structure of the piping work-density evaluation process. The density evaluation process begins with developing and sequencing work groups based on complexity lists. A work group consists of pipelines of a similar kind e.g. material types, sizes, and fabrication types. ‘A work group eventually becomes, after density analysis, a long-term work package’, which consists of work with a certain degree of homogeneity. The provision of piping work heuristics also assists in creating more efficient work groups and work packages.

Figure 3: Piping Work-Density Evaluation Process
Conclusions and Review

This paper provides a systematic means of modeling a complex work-packaging process. To develop and sequence work packages, this paper focuses on identifying information flow among work processes. The research results show that such approach is especially useful for reducing redundancies involved in the current practices of the piping work packaging. Although the DFD and DSM (the Design Structure Matrix) commonly focus on information flow in identifying information dependencies among work activities and sequencing work packages, the DFD method adopted in the paper appears to represent the information flow more dynamically than DSM.
The WorkPlan developed in this paper aims at helping systematically develop weekly work plans. The fundamental concept of scheduling adopted by the WorkPlan is to learn from the failures of the past work. The WorkPlan incorporates learning from the past into new planning work. The authors argue that the systematic approach adopted by WorkPlan can help implement lean construction by providing a tool to identify constraints in resources and check constraint satisfaction.

Job-Shop Scheduling and WorkPlan

Although the ability to get work done at construction greatly depends on resource availability, there has been lack of tools to check resource availability prior to starting the work. CPM has a function to allocate resources to activities, but it does not make it easy to specify and check prerequisite resources. As a result, difficulties in handling discrepancies between anticipated, actually needed and actual resources necessitate an easy tool to verify resource assignment.

In addition, while CPM focuses on a definite start and finish data, making it possible to compute activity float and then level resource histograms, ‘job-shop scheduling focuses on the continuous flow of work (jobs), where jobs have due dates but their execution may be interwoven so that there is no clear start or finish for the shop’s operation as a whole’. Shop capacity tends to be a given, thus it is critical to maintain a steady flow of work for all resources, which can be achieved by creating a workable backlog. For this, the WorkPlan supports the flow of information by making all constraints explicit in the planning phase.
Structure of WorkPlan

Traditionally, a work package has been defined just as ‘a sub-element of a construction project, on which both cost and time data are collected for project status reporting.’ In order to systematically identify constraints involved in construction process, however, a work package needs to be more clearly defined. In this paper, it is defined as ‘a definite amount of similar work to be done (or a set of tasks) often in a well-defined area, using specific design information, material, labor, and equipment, and with prerequisite work completed.’ According to this definition, great improvement of construction processes is possible by allowing continuous flow of resources. In this regard, the WorkPlan aims at the development of a constraint-based database for work package scheduling that can allow close monitoring of work packages.

In the WorkPlan, constraints are assumed specific to each work package and are tracked as part of the work package information. The constraints are categorized into the following five: Contract, engineering, material, labor and equipment, and prerequisite work and site conditions. The work package that does not meet any of these constraints should not be started otherwise its execution will be slowed or interrupted during construction.

Figure 1 illustrates the structure of work packages as implemented the WorkPlan. In the WorkPlan, constraints enumerated on the left side in Figure 1 and resources at the top in Figure 1 are considered when preparing work plans. At the bottom is accounting-related data. Cost-to-Date calculates costs incurred up to the reporting data, while ‘Forecast-to-Complete describes in monetary terms the number of hours of labor and equipment that is expected to need to complete the work.’ The forecast is not based on the allocated budget or numerical average of the past cost, but it ‘reflects only remaining work to be done given the actual site conditions anticipated.’ As a result, planners will have timesheets, costs report, and PPC report, which is used to estimate the reliability of the planning system.
Figure 1: Structure of Work Packages in WorkPlan
Conclusions and Review

Above all, the WorkPlan differs from the existing commercial packages that can track project information in that 'it adopts a work package view, that is, all record keeping is structured in function of what may constrain the execution of a work package. In addition, the WorkPlan incorporates some lean concepts into construction planning, some of which are as follows:

- To avoid making low-quality assignment by systematically checking all constraints before releasing work packages to the downstream.
- To allow each person to know what others do and understand the implications of the quality of their own work on the quality of the process output, by clearly documenting, updating, and constantly reporting the status of all process flows.
- To reduce wait time for people or machines by synchronizing and physically aligning all steps in the production process.

However, the WorkPlan implemented in this paper does not consider constraints arisen from material. In addition, it does not capture the effects of interactions among activities. In this regard, it is questioned that one of the fundamental concepts underlying the WorkPlan, learning from the past, can be really achieved, since to identify the causes of the past failures, it is required to capture and understand interactions among activities.
Based on the recognition that efficient coordination of activities can greatly improve construction performance in terms of time and costs, the authors develop a system based on simulation techniques. The system especially aims at promoting concurrent engineering approaches to geotechnical construction work and providing a tool, with which one can select the most appropriate design for construction.

Advantages of Simulation Approaches

Simulation approaches have an advantage over the exiting tools in terms of that they can address ‘what is needed and when’ and the result can be compared with the original plans for verification. Consequently, they help construction resources to be optimally distributed.

System Architecture

The system represented in Figure 1 consists of two primary parts: Design modules and Activity modules. Design modules contain knowledge bases that need to be considered before construction, while Activity modules contain knowledge bases related to construction activities. Specifically, within the system Design modules define ‘what has to be done’, Activity modules determine ‘how to do it’, and finally construction simulation module performs a simulation answering to ‘when to do it’.

The system is designed to compare the effectiveness of alternative construction methods for geotechnical construction and provide the most appropriate solution. The first task of the system begins with verifying whether site improvement is needed for the
site. Then, the system determines problems related to excavation under the current foundation design. Finally, it analyzes the constructivity of the foundation members. After finishing these procedures, construction phase can start. New information is acquired from the site so that the system can define the necessary site preparation. The site preparation provides a solution for possible conflicts arisen from local site conditions. Once all detected conflicts are settled, the system proceeds construction. Otherwise, unsolved conflicts are asked to the user or sent to the design team. Activity module simulates the necessary activities to perform construction, considering the mutual interactions of the activities, logical constraints and resource constraints. Finally, the system produces a feasible construction schedule, which will be compared with the original plan.

Figure 1: System Architecture
Construction Simulation

The construction simulator consists of two part: simulation modules and a simulator engine. The simulation models assign construction activities, split them into component activities, allocate labor and equipment, and define unit production rates. The simulator engine then simulates construction processes. During the simulation each activity draws the necessary workforce and equipment from the resources available at the construction site. The results of the simulation are provided in the form of percent completion of work and degree of resource utilization at each time step.

1) Simulation Modules

Expert system rules are developed to decompose construction tasks, allocate resources, and determine unit production rates. They are applied for earthmoving, pre-loading, and pile installation.

2) Simulator Engine

The simulator is used to model construction processes. ‘Simulation embodies the principle of “learning by doing”. The simulator also makes it possible for a project manager to stop the simulation, change variable, and verify the influence of changes during construction processes. Various libraries containing a list of standard operations and activities are also provided, which can be added and modified. They allow the user to interact with a ‘living’ model of the activities during planning and controlling construction projects.

3) Functional Block

‘A functional block is a structure representing an operation’, which is composed of a set of sub-elements including activation functions, links, attributes and logical
relationships. The roles of each sub-element are summarized in Table 1 and graphically represented in Figure 2.

Table 1: Elements of Function Block

<table>
<thead>
<tr>
<th>Elements</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrow</td>
<td>Logical or functional link to other blocks</td>
</tr>
<tr>
<td>Entering link</td>
<td>Results of preceding operation</td>
</tr>
<tr>
<td>Activation Function</td>
<td>Interact with exterior computing results, create objects, allocate</td>
</tr>
<tr>
<td></td>
<td>resources, interact with queues and database</td>
</tr>
<tr>
<td>Existing Link</td>
<td>Communicate the completion of operations</td>
</tr>
<tr>
<td>Logical Relationship</td>
<td>Determine activation state of blocks</td>
</tr>
</tbody>
</table>

Figure 2: Function Block
Review

This paper presents an ambitious scheme for the implementation of the expert system in construction. The system scheme can be good guidelines for the development of a planning tool based on simulation approaches. In particular, this paper involves a lot of modeling implications by providing detailed technical descriptions for constructing the system architecture of the construction simulator and by showing the connectivity of a simulation method with either existing planning tools or future applications of the expert system. However, the paper does not provide any result of the system’s applications into real projects thus it is questioned that the system schemes provided in the paper have been properly validated.
Heuristic rules have been widely used for resource allocation and leveling in project management. But, they cannot guarantee optimum solutions. In this paper, the Genetic Algorithms (GAs) are adopted to find near-optimum solution. In addition, the introduction of various generic algorithms assists in understanding of the optimization principles and procedures for construction resources.

Resource Allocation and Leveling Heuristics

In an effort to overcome the limitation of CPM and PERT in dealing with project resources, two types of techniques have been mainly used: resource allocation and resource leveling. While the former ‘attempts to reschedule the project tasks so that a limited number of resources can be efficiently utilized while keeping the unavoidable extension of the project to a minimum’, the latter attempts to reduce the sharp variations among the peaks and valleys in the resource demand histogram while maintaining the original project duration (Moselhi and Lorterpong 1993). However, these techniques only can be applied to a project ‘one after another rather than simultaneously’. For this reason, they do not always produce a project schedule that minimizes the project time and costs (Karshenas and Haber, 1990).

During the past decades various approaches have been made to handle resource allocation problems as well as heuristic approaches. Among them are integer programming, branch-and-bound, and dynamic programming (Gavish et al. 1991). None of them, however, are practical for real-life problem size due to computational limit. On this front, heuristic approaches have been introduced as an alternative to these optimization approaches. The heuristic rules ensure the project activities to be
scheduled, keeping the logical relationships and resource amount constraints. However, ‘this comes on the expense of the total project duration, which often exceeds the duration determined by the original CPM analysis.’ Moreover, the performance of heuristic rules varies when used on different networks and there have been no solid theoretical bases for selecting the best rule for a given network.

On the other hand, resource leveling problems have been most often handled based on mixed integer program formulation (Shah et al, 1993) but such formulations are only applicable to small-sized construction projects. To overcome this practical limitation, heuristic algorithms have been introduced and the minimum moment algorithm has been most widely used to date (Harris, 1978). ‘Despite the simple nature of resource leveling heuristics and their wide implementation on commercial project management software, they can only produce good feasible solutions and by no means guarantee an optimum solutions.’

Improving Resource Allocation Heuristics

One common practice to select an optimum heuristic rule is to apply various heuristic rules and then select the schedule making the duration minimized. However, this approach has little diversity due to the small number of applicable rules. To overcome this, an approach of forcing random activity priorities is presented to help select a near-optimum schedule. This approach introduces some bias into some activities e.g. giving the highest priority to a certain activity, while others are set to the lowest and consequently monitor the impact on the schedule. Since it is not possible to figure out which activities should have higher priorities than others, a iterative procedure may be needed. However, this approach cannot ‘identify an optimum set of activities’ priorities.’

Improving Resource Leveling Heuristics

The minimum moment algorithm has been generally used to deal with resource leveling problems in construction project. This method focuses on measuring the
fluctuations in daily resource demands. While this method can be used to compare alternative resource histograms in terms of resource fluctuation, it does not consider the resource utilization. To solve this problem, the author presents a mathematical formula and argues that ‘having the moment calculations, a project manager may use them as modified heuristics according to his resource management objectives.’

**Optimization Search**

A search technique based on artificial intelligence, Genetic Algorithms, is introduced to find a near optimum set of activities’ priorities that can minimize the total project duration and costs, given the modified heuristics. ‘GAs are, in essence, optimization search procedures inspired by the biological systems’ improved fitness through evolution.’ GAs find optimal solutions based on a ‘random-yet-directed’ search method. Based on the GA implementation procedures, an GA algorithm is developed to solve resource allocation and leveling problems.

**Conclusions and Review**

The research results show that with GA search method, one can arrive at solutions by searching only a small fraction of the total search space. As a result, the computational limitation encountered with a mathematical optimization method can be overcome in finding optimized resource allocation and leveling. In addition, the research shows the possibility of GAs’ further extensions to other objects, e.g. finding the optimal early completion date of construction project that makes the time-cost tradeoff maximized.
Robust Design

Robust Design is defined as ‘an engineering methodology for improving productivity during research and development so that high-quality products can be produced quickly and at low cost.’ Since developed in the early 1960s by G. Taguchi, it has been widely used in electronics, automotive products, photography, and many other industries. ‘Robust Design draws on many ideas from statistical experimental design to plan experiments for obtaining dependable information about variables involved in making engineering decisions.’ Among statistical experimental methods, Robust Design has most often employed the orthogonal arrays, whose use for planning experiments was first proposed by C.R. Rao. Robust Design also adds a new dimension to statistical experimental design, which addresses the following concerns facing product and process designers.

- How to reduce economically the variation of a product’s function in the customer’s environment.
- How to ensure that decisions found to be optimum during laboratory experiments will prove to be so in manufacturing and in customer environments.

Quality

Robust Design not only considers quality in manufacturing processes, but also aims at achieving the target quality under all intended operating conditions and throughout the product’s lifecycle. The pursued quality in Robust Design is well explained by the following Taguchi’s notion: ‘we measure the quality of a product in terms of the total loss to society due to functional variation and harmful side effects.’
Fundamental Principles

The fundamental principle of Robust Design is ‘to improve the quality of a product by minimizing the effect of the causes of variation without eliminating the causes.’ This approach aims at minimizing control efforts, while achieving the target quality of a product. This can be achieved by ‘optimizing the product and process designs to make the performance minimally sensitive to the various causes of variation’, which is called ‘parameter design’. One parameter to determine the success of Robust Design is to measure whether the benefits of improved quality can justify the added product cost.

Tools Used in Robust Design

A significant engineering effort is devoted in generating information about how different design parameters affect performance under different usage conditions. Robust Design attempts to reduce the engineering/experimental time and efforts to get information necessary for decision-making in production or process designs by mainly focusing on the following tasks:

- Measurement of quality during design/development: A leading indicator of quality is required, which can evaluate the effect of changing a particular design parameter on the product’s performance.

- Effective experiment to find dependable information about the design parameters: It is critical to obtain dependable information about the design parameters with minimum time and resources so that design changes during manufacturing and customer use can be avoided.

Once having the estimate of design parameters’ effect on product quality, ‘they must be valid even when other parameters are changed during the subsequent design effort or when designs of related subsystems change.’ To achieve this, Robust Design employs the signal-to-noise (S/N) ratio and orthogonal arrays.
Applications and Benefits

The application of Robust Design is not limited to engineering objectives. Robust Design has been proven to be useful in profit planning in business, cash-flow optimization in banking, government policymaking, and other area. This method has been also applied to ‘determining optimum work force mix for jobs where the demand is random, and improving the runway utilization at an airport.’

Principles of Orthogonal Arrays

One of the benefits of using orthogonal arrays is the simplicity of data analysis. A matrix experiment is conducted with a set of experiments where the setting of the various product or process parameters are changed. The data resulting from the experiments are then analyzed to determine the effects of the various parameters on the performance of the product or the process. Orthogonal arrays allow such analysis to be done efficiently by computing simple average and an approach that has an intuitive appeal.

The following example, a matrix experiment for a CVC process, shows one application of orthogonal arrays in determining the effect of various parameters. The experiment aims at determining the effect of process parameters on the formation of certain surface defects in a chemical vapor deposition (CVD) process and finding the optimized setting for each parameter that can minimize the surface defect. The process parameters are temperature (A), pressure (B), settling time (C), and cleaning method (D). The starting levels (levels before conducting experiment) for the four parameters are identified by underscore in Table 1, together with other alternative levels. Without using orthogonal arrays, matrix experiment needs to be done eighty one times to determine the effects of all possible combinations of factor levels: 3*3*3*3. The matrix experiment listed in Table 2 makes it possible to obtain the almost same results with only nine-times experiments. The benefits of using orthogonal arrays dramatically increase when the number of factors or factor levels are increased.
Table 1: Factors and Their Levels

<table>
<thead>
<tr>
<th>Factors</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>A. Temperature</td>
<td>$T_0 - 25$</td>
</tr>
<tr>
<td>B. Pressure</td>
<td>$P_0 - 200$</td>
</tr>
<tr>
<td>C. Settling Time</td>
<td>$t_0$</td>
</tr>
<tr>
<td>D. Cleaning Method</td>
<td>None</td>
</tr>
</tbody>
</table>

Table 2: Matrix Experiment

<table>
<thead>
<tr>
<th>Expt. No.</th>
<th>Column Numbers and Factor Assigned</th>
<th>Observation (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-1-1-1</td>
<td>-20</td>
</tr>
<tr>
<td>2</td>
<td>1-2-2-2</td>
<td>-10</td>
</tr>
<tr>
<td>3</td>
<td>1-3-3-3</td>
<td>-30</td>
</tr>
<tr>
<td>4</td>
<td>2-1-2-3</td>
<td>-25</td>
</tr>
<tr>
<td>5</td>
<td>2-2-3-1</td>
<td>-45</td>
</tr>
<tr>
<td>6</td>
<td>2-3-1-2</td>
<td>-65</td>
</tr>
<tr>
<td>7</td>
<td>3-1-3-2</td>
<td>-45</td>
</tr>
<tr>
<td>8</td>
<td>3-2-1-3</td>
<td>-65</td>
</tr>
<tr>
<td>9</td>
<td>3-3-2-1</td>
<td>-70</td>
</tr>
</tbody>
</table>

The matrix experiment in Table 2 is the standard orthogonal array L$_9$ provided by Taguchi. The matrix experiment is made in the way that for any pair of columns, all combinations of factor levels can occur and they can occur an equal number of times, which is called the ‘balancing property’ and it implies orthogonality. Many researchers have investigated constructing orthogonal arrays. In particular, Taguchi provides a graphic tool called ‘linear graphs’, which represents interactions between pairs of columns in an orthogonal array. He also developed a set of standard orthogonal arrays.
This paper demonstrates the usefulness of system dynamics for the construction planning and control. A system dynamics model was developed and applied to monitor and control project baselines of the Bugis Junction Basement construction, which is the largest basement construction in Singapore.

**Why System Dynamics?**

Although many researchers have demonstrated the benefits of simulation in construction project operation, the simulation methods is not widely used in the construction field yet. In fact, most of the current project management tools are static for monitoring, controlling, and assessing the impact of construction changes on the control baselines (Pritsker et al 1989). In this regard, the authors argue, the system dynamics can satisfy the need in project planning and control by providing a dynamic simulation model for the process-level network.

**Modeling**

The modeling work has been done through most commonly accepted procedures; model conceptualization (causal loop diagram), model structure development, model formation, and model valuation. The model conceptualization begins with observation and understanding of basement construction processes. Then, root causes of potential problems and all of the variable that have a significant, direct effect on the process are identified. The model structure consists of three subsystems, each of which represents spoil removal, lean concreting, and cast concrete activities respectively.
The key determinants of production rate for activities are categorized mainly into:

1) physical constraints such as area availability, soil type, accessibility to the site, and weather,
2) resource availability,
3) management decisions.

During the model formation, 'the model has to be structured so that it can reproduce the actual field data when tested.'

**Figure 1:** Integrated Flow Diagrams of Three Models

**Sensitivity Analysis**

Once the model has been performed adequately 'in a form of reasonable representation of the actual construction process', sensitivity analysis is conducted to identify relative importance and impact of variables on the project performance. Five variables are chosen and then sensitivity of each variable is analyzed. Sensitive variable identified as a result of the sensitivity study would enable the project executives to know where to focus their attention on.
Conclusions and Review

The system dynamics modeling makes it possible to codify person’s tacit knowledge and thought. It is believed that this paper has demonstrated the applicability of system dynamics into construction process planning for the first time. The system dynamics can be useful for the strategic management of construction as well as process-level management. In addition, although the modeling scope in this paper is limited to a specific process, the model functions and boundary can be extended depending on modeling purpose and usage.

One handicap of the system dynamics is that it requires user’s ability ‘to handle non-numerical data’. In other words, a user should be able to quantify based on non-numerical data the impact of a certain variable on the system. In this regard, we cannot expect the potential user of the system dynamics in the industry eg. project manager or site engineers to acquire such skill and knowledge. Such difficulties in the modeling necessitate the development of an easier tool that can interface between system dynamics and users on construction site.
In practice, to identify feedback loop dominance has been one of the most challenging issues in system dynamics modeling. Identifying dominant loops has been traditionally done with experimental model exploration, model reduction, or both with understanding of the behavior patterns [Richardson, 1991]. However, 1) these informal approaches can lead to errors due to potential vulnerability to variation and bias resulting from people’s expectations and model circumstances. In addition, basing the polarity during model exploration can also produce errors. This is due to the fact that ‘feedback loops do not generate unique behavior patterns and they are only loosely coupled to specific behavior patterns.’ This argument is supported by some atypical behaviors such that positive and negative loops can produce linear behavior when systems are in equilibrium. Another potential disadvantage of informal approaches is that 2) they are tacit and uncodified.

To overcome these, research efforts have been recently made to develop a formal method, focusing on the structural aspect rather than on behavioral aspect in identifying feedback loops dominance. However, ‘these structural approaches address only a portion of the possible feedback structures, are difficult to apply, or are impractical for models of significant size.’ In addition, to rigorously analyze loop dominance, system dynamists need the following things: 1) automated analysis tool and 2) a clear understanding of loop dominance. On this front, this paper presents a practical tool and an analytic procedure to identify dominant feedback loop among multiple feedback loops in a model structure based on systems behaviors, with which an analysis procedure can be formalized. In conclusion, the research results are expected to help ease the difficulties in explaining how structure drives behavior and increase the reliability of system dynamics modeling results.
Today’s industry systems tend to increasingly depend on knowledge intensive processes whose management and operations require interdisciplinary team approach. Under a rapidly changing business environment, which requires faster and cheaper delivery of a product, industries have more widely adopted methods such as concurrent development and co-located cross-functional teams. Such methods require ‘multiple knowledge-driven processes.’ On the other hand, ‘knowledge intensive processes are often driven and constrained by the mental models of experts acting as direct participants or managers.’

One problem facing modelers who attempt to elicit and represent the knowledge of experts is that the knowledge is not explicit so it is very difficult to describe and incorporate their knowledge into modeling. In fact, many knowledge elicitation methods have been developed to date. Their application, however, is limited to the early stages of modeling such as ‘problem articulation, boundary selection, identification of variables, and qualitative casual mapping.’ On this front, this paper presents a knowledge elicitation method using ‘formal modeling and three description format transformations to help experts explicate their tacit knowledge.’

Background of Expert Knowledge Elicitation

The most methods being used in system dynamics aim at developing conceptual designs and models. For example, the ‘metaphor-analogy-model method’ cannot produce explicit and specific enough to be used in formal modeling, while effective in qualitative model conceptualization. Formal modeling should be based on detailed description of relations at ‘an operational level’ and requires more precision than conceptual modeling.
To this, formal modeling should be represented as of 'stock and flow structure, functional forms, and numerical estimates of parameters and behavioral relationships.' The motivation of the approach presented in this paper comes from the diversity of information characteristics. Forrester [1994] points out that written or numeric knowledge has its strength of being codified and more widely accessible than mental model knowledge, while written knowledge can be limited by the richness it can describe, the inability of modelers to understand it beyond text, and being filtered and biased during codification. However, he clearly recognizes the value of written knowledge as well and seeks a method to overcome its drawbacks. Meanwhile, the formal modeling method in this paper is developed based on the hypothesis that ‘pushing experts to describe relationships at the simulation model level helps them to clarify and specify their knowledge more than they would at a more abstract level using tools such as causal loop diagrams.’

Knowledge Elicitation Method

The knowledge elicitation method consists three sequential phases: positioning, description, and discussion. Details are as follows.

<table>
<thead>
<tr>
<th>Positioning: establish a context and goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establish context</td>
</tr>
<tr>
<td>Create an environment, in which the elicitation will occur by 1) describing the model purpose, 2) major subsystems and their interactions, 3) the roles and structures of the subsystems, and 4) the relationship to be characterized</td>
</tr>
<tr>
<td>Focus on one relationship at a time</td>
</tr>
<tr>
<td>Describe the relationship between two activities by identifying and defining 1) input and output variables that the relationship describes with units of measure, 2) where the relationship is used in the model, 3) why the relationship is important and 4) what other parts of the system and model the relationship affects.</td>
</tr>
<tr>
<td>Illustrate the method</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td><strong>Description</strong>: develop relationships</td>
</tr>
<tr>
<td><strong>Visual description</strong></td>
</tr>
<tr>
<td><strong>Verbal description</strong></td>
</tr>
<tr>
<td><strong>Textual description</strong></td>
</tr>
<tr>
<td><strong>Graphic description</strong></td>
</tr>
<tr>
<td><strong>Discussion</strong>: Test, understand, and improve the descriptions of different experts</td>
</tr>
<tr>
<td><strong>Examine individual descriptions</strong></td>
</tr>
<tr>
<td><strong>Compare descriptions</strong></td>
</tr>
</tbody>
</table>
Conclusions

This knowledge elicitation method has been applied to a new semiconductor development project and based on the case project the authors assess the method as follows:

1) Multiple formats are more efficient in capturing experts’ tacit knowledge that may be lost with a single-step method.

2) Four description formats improve information quality through triangulation both within individuals and across experts.

3) Multiple steps in generating graphic representation reduce the cognitive processing, provide more time to reflect and revise, and as a result improve knowledge elicitation.

4) Honoring the full range of participants’ expertise improve knowledge elicitation.
The performance of construction projects has been traditionally measured based on variance analysis of planned and actual cost and schedule. And, control actions to minimize the variances have been usually done by forcing subcontractors to accelerate their work. However, ‘the causes of the variance are not necessarily the basis of these actions and traditional measurements do not provide predictive impact assessment of performance variance.’ On this front, the authors attempt to develop an integrated, computer aided 4D visual performance measurement system (4D VPMS) that can realize dynamic project control through visualization of reliability and flow. Their approach is based on the lean techniques, which focuses on improving the reliability and flow of processes, thus improving predictability of the project performance. In addition, ‘the 4D VPMS introduces the concept of Future Path Method (FPM), which is the collaboration of prediction, performance measurement, and 4D animation technologies.’

4D VPMS

Figure 1 illustrates the system architecture of 4D VPMS. The main structure of the system consists of an object oriented 3D CAD model, the schedule simulation module, performance measurement module, and a set of databases. Schedule is dynamically simulated based on underlying data in a central database and design data provided by 3D CAD design model, and using simulation software. Performance measurement module monitors reliability and flow of processes based on their underlying data, and provides information necessary for dynamic control of the processes. Using dynamic input and continuously monitored underlying data, changes to schedule and conflict points are identified and predicted. 4D VPMS under development will have the following benefits and capabilities:
- Measure the reliability of construction processes
- Reduce uncertainties involved in design and technical specifications
- Reduce uncertainties involved in executive construction planning
- Enhance visual control
- Provide a tool for shielding production: ‘Future Path Method’
- Create conditions for simple management of a construction project.
- Increase process transparency by developing web-based 4D VPMS

**Performance Measurement**

This research attempts to provide a tool for dynamic analysis of simulation data during the simulation of schedule. The dynamic analysis tool determines the reliability of processes and assists in understanding the resource bottlenecks, correcting them, and
making effective decisions. As an example of dynamic control processes, the process of RFI management is presented. In the example, the control system helps a project manager to avoid potential delay by monitoring RFI logs, expected deadlines, and actual dates.

**Future Path Method**

The FPM aims at shielding production based on the lean production concepts. 'Shielding is accomplished by making quality assignments thereby increasing the reliability of commitment plans, such as weekly look-ahead plans (Ballard and Howell, 1998).’ The concept of making quality assignment is to be achieved by increasing the reliability of the look-ahead plan.

**Conclusions and Review**

The research results imply that efficient performance measurement can help identify the causes of failures and increasing transparency of construction processes can help improve information flow and shield construction processes from uncertainties. In addition, the visualization of the future path of schedule can help the project manager develop a commitment plan. However, the paper does not present backup theories to support their system nor applications. Moreover, as for the system architecture, the paper does not show detailed descriptions with a solid logical background.
This paper introduces two construction research projects being conducted by the collaboration between the industry and universities, addressing process improvement and the supporting role of IT in design team integration. The research projects, the CONCUR and Time-Comp, have the same objectives: changing the construction industry culture, rationalizing project processes, managing risk, pruning process trees down to the value-adding branches, and integrating design with construction. However, their approaches are different: the CONCURE focuses on concurrent engineering, while the Time-Comp focuses on time compression.

Concurrency and Time Compression

The concept of concurrency and time compression can be realized by the following activities:

<table>
<thead>
<tr>
<th>Activity</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>consecutive Sequential activities. Valid if there is a restriction on labour, space, erection sequence etc.</td>
</tr>
<tr>
<td>b</td>
<td>concurrent Wholly independent activities having no resource or space restrictions.</td>
</tr>
<tr>
<td>c</td>
<td>partition An activity reorganised into sub-tasks with sub-results allowing a dependent activity to be started earlier.</td>
</tr>
<tr>
<td>d</td>
<td>shorten Putting in more resource to do an activity faster or doing a task in an improved way.</td>
</tr>
<tr>
<td>e</td>
<td>change Obtaining a result in an entirely different way. Perhaps factory fabrication rather than site construction.</td>
</tr>
<tr>
<td></td>
<td>eliminate Eliminating an unnecessary activity such as materials storage by just-in-time delivery.</td>
</tr>
</tbody>
</table>

Figure 1: Concepts of Concurrency and Time Compression
Implementation

1) Model Integration

The integration of the individual models developed in different disciplines can be integrated by focusing on the cyclic nature of work in a project, while the models are in different styles. ‘Essentially the same tasks are undertaken at every stage in project evolution, but the degree of detail, the accuracy, the methods, and the tools change’. This cyclic nature of work makes it possible to keep consistency between discipline activities in the process model. For the process modeling, IDEFO (Integration Definition for Function Modeling) has been adopted. In addition, the common framework employed by different individual models consists of four primary elements: input, control, mechanism, and output.

\[
\begin{array}{c}
\text{Control} \\
\downarrow \\
\text{Input} \\
\text{Activity} \\
\uparrow \\
\text{Mechanism:} \\
- \text{Person}, \\
- \text{Machine}, \\
- \text{Computer program} \\
\text{Output}
\end{array}
\]

2) Complexity Modeling

The IDEFO handles the complexity problem that one can face when complex processes are represented in a single diagram, by using hierarchical decomposition. The hierarchical decomposition comes from a common 'engineering approach of first
looking at the big picture and then into progressively more levels of detail.’ As conceptualized in Figure 2, a higher level diagram can show a clear, broad view of overall processes but it does not have detailed representations of actual tasks. To see more detailed information on the actual tasks, another IDEF0 diagram containing information on their sub-activities needs to be opened. This procedure continues until the desired level of detail is achieved.

**Figure 2: Hierarchical Structure of IDEF0**

**Review**

This paper describes only a little portion of the two projects. Comprehensively, their approaches can be seen as product modeling, which attempts to integrate all information produced by each project function, mainly by standardizing product data and utilizing information technologies. One thing interesting is that they incorporate concurrent engineering concepts into the their system, although the details are not known. In addition, their approach of hierarchical decomposition can be useful in any simulation modeling work facing complexity problems.
Two main objectives of this thesis are to find 1) the proper ordering of multiple development processes and 2) the appropriate concurrency between coupled tasks. To this, a mathematical model is developed based on ‘characterizing the information exchanged between tasks using the probability of change and impact.’ The research results will help estimate 1) the probability of the process completion at a given time and 2) ‘the average completion time and cost of the development effort for a given degree of concurrency and task sequencing’. The following compares differences in modeling approaches between this research project and the proposed dynamic planning and control methodology.

<table>
<thead>
<tr>
<th>Description</th>
<th>Information Flow Model (Carrascosa M., 1999)</th>
<th>Dynamic Planning and Control Methodology (Proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectives</td>
<td>To find the proper ordering of tasks</td>
<td>To provide policy guidelines for the strategic planning and management of LSI</td>
</tr>
<tr>
<td></td>
<td>To find the appropriate degree of overlapping of coupled tasks</td>
<td>Performance Profiles (e.g. time, cost, quality, safety, environment)</td>
</tr>
<tr>
<td>Output</td>
<td>Lead time, costs, overlapping degree</td>
<td>Performance Profiles (e.g. time, cost, quality, safety, environment)</td>
</tr>
<tr>
<td>Type of Coupled Tasks</td>
<td>Exchanged between tasks</td>
<td>Information (tasks having relationships with design), Physical progress (between construction tasks)</td>
</tr>
<tr>
<td></td>
<td>Information</td>
<td></td>
</tr>
<tr>
<td><strong>Type of info./progress flow</strong></td>
<td>Unidirectional, Bi-directional</td>
<td>Bi-directional</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong># of tasks</strong></td>
<td>Multi</td>
<td>Multi (determined by modeling purpose and targeted accuracy)</td>
</tr>
<tr>
<td><strong>Determinants of dynamics of iteration process</strong></td>
<td>Information communicated between tasks</td>
<td>Information, physical progress, constraints (e.g. resource, working conditions)</td>
</tr>
<tr>
<td><strong>Task Characterization</strong></td>
<td>Duration, costs, starting time</td>
<td>Duration, costs, starting time, production type (fast and slow) and rate</td>
</tr>
<tr>
<td><strong>Key Determinants of overlapping effectiveness</strong></td>
<td>Probability of change: Likelihood of changes happening as tasks progress (2 types: concave-up and convex-up)</td>
<td>Design changes: Probabilistic distribution of a given amount of changes</td>
</tr>
<tr>
<td></td>
<td>Probability of impact: Likelihood of rework caused by changes (2 types: constant and linear)</td>
<td>Construction reliability: a function of the size of errors made during construction over time (depends on site conditions, constant)</td>
</tr>
<tr>
<td></td>
<td>* Comparison with Krishnan et al. Model (1997)</td>
<td></td>
</tr>
<tr>
<td><strong>Carrascosa</strong></td>
<td>Stochastic (probability of change and impact)</td>
<td>Construction sensitivity: Likelihood of non-value adding changes caused by design changes and errors, and construction errors (subject to change according to upstream, constant)</td>
</tr>
<tr>
<td></td>
<td>The probability is determined by timing of changes and a function of the state of completion of a task.</td>
<td></td>
</tr>
<tr>
<td><strong>Krishnan et al.</strong></td>
<td>Deterministic (evolution and sensitivity)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evolution and sensitivity are determined based on size of changes.</td>
<td>* Adoption of Carrascosa’s concepts will be reviewed and discussed</td>
</tr>
<tr>
<td>Key Modeling Assumptions</td>
<td>Changes vs. Rework</td>
<td>Probabilities of change</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Changes with the highest impact determines rework</td>
<td>Independent of each other</td>
<td>The overall model development process is streamlined and appropriate ordering of tasks are decided, using a DSM. Then, remaining coupled tasks are analyzed using the model.</td>
</tr>
<tr>
<td>Non-value adding changes in a task is the sum of the non-value adding changes caused by different changes. (The assumption of Carrascosa’s model is more reasonable, but it makes the modeling complicated and the difference is minor)</td>
<td>Correlated</td>
<td>Determined by a number of parameters</td>
</tr>
<tr>
<td>Effect of learning curve</td>
<td>Not considered</td>
<td>Considered</td>
</tr>
<tr>
<td>Role of Control Policies</td>
<td>Not allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td>Interaction between tasks</td>
<td>Limited to two parameters</td>
<td></td>
</tr>
<tr>
<td>Modeling</td>
<td>Mathematical Representation (Markov Chain)</td>
<td>System Dynamics</td>
</tr>
</tbody>
</table>
APPENDIX II

SYSTEM DYNAMICS MODEL STRUCTURES
Construction Process Model
Work in Process Model
Dependency Control Model
Work Quality Model
Buffering & Time Control Model
Resource Control Model
OverTime & Fatigue Model
Schedule Pressure Model
Workforce Monitoring Model

<ActivityReallyFinished>

WorkBeingDone

pjLabor

HoursPerDay

WorkhoursPerDay

<EffectiveWorkForce>

CumulativeLaborHoursPJ

<ProjectFinished>

WorkforceUtilizationRatio

<WorkRate>

<PotentialWorkRatefromResource>
Progress-based Parameter Model
RFI Model
Hidden Change Co-flow Model
Impact on Work Released Model
APPENDIX III
SYSTEM DYNAMICS MODEL EQUATIONS
Unit: Day
Description: Average time for quality management

DesiredWorker[activity] = IF THEN ELSE(ActivityStarted[activity] = 0, 0.5, 1) *
RequiredWorkRate[activity] / PerceivedProductivity[activity]
Unit: worker
Description: Ramping up workforce normally starts when tasks become available.
However, once reliability buffering is applied, it starts with buffering making
required workforce ready in a timely manner. for this reason, instead of
'ResourceCommisionStart', 'ActivityStarted' is used in the equation.

IntendedChangeRatioPJ = 0
Unit: Dimensionless

ProductionType[activity] = IF THEN ELSE(FastEvolution[activity] = 1, 0, 1)
Unit: Dimensionless
Description: The pattern of productivity resulting from the function of work
complexity and learning effect.

\[ g(\text{IntendedChangeRatioPJ}) = \text{GAME}(\text{IntendedChangeRatioPJ}) \]
Unit: Dimensionless

IsInstantaneousActivity?[activity] = 0
Unit: Dimensionless
Description: Referring to an activity that does not require quality control actions
and create any feedbacks. So, the activity needs to be finished independent of
external conditions and within the scheduled duration. Some examples of this kind
activity include 'demonstration' and 'notice to a public institution'.

ActivityScope[activity] = OriginalDuration[activity] * ConverterforScope&Duration
Unit: wu
Description: It is used to create atomic work units flowing through the model
structure, given duration.

ProgressInfluentialtoQuality[activity] = MIN(1, WorkOnceProcessed[activity] /
ActivityScope[activity])
Unit: Dimensionless
Description: Work progress that influences the work quality. So, it is made of
work that have been once processed either with or without resource
commissioning.

TableForRippleEffectonWorkQuality([[0,0]-
(1,1)], (0,0), (0.02, 0.02), (0.03, 0.1), (0.05, 0.2), (0.25, 0.6), (0.5, 0.8), (1,1))
Unit: Dimensionless
Description: It refers the degree of ripple effect on the downstream work quality, when hidden upstream changes impact the downstream work. In case of quality deterioration, the impact has a non-linear pattern.

ActualProductivity[activity]=ProgressBasedProductivity[activity]*SKDLEffectOnPDY[activity]
Unit: wu/(worker*Day)

ActualWorkQuality[activity]=IF THEN ELSE(ResourceCommisionStart[activity]=0, 1, ProgressBasedWorkQuality[activity]*(1-TableForRippleEffectonWorkQuality(FractionofUPHidenChangeNotAddressed[activity])*FatigueEffectonQuality[activity]))
Unit: Dimensionless
Description: The fraction that determines how much work will be done correctly.

RPTriggeredbyDNIntendedChange[activity]=SUM(IF THEN ELSE (ReprocessIterationRelationships[activity,preceding!]>-1, 1, 0)*TotalExternalIntendedChange[preceding!] *ActivityScope[activity]/ActivityScope[preceding!] *ExternalSensitivity[activity,preceding!] *FractionofWorkReleased[activity])
Unit: wu/Day
Description: Sum of needs for reprocess triggered by downstream activities. Actual amount is adjusted by work scope of associated activities, the activity's external sensitivity, and work released so far.

IntendedChangeRatioforActivity[activity]=0.25
Unit: Dimensionless

RPTriggeredbyInternalIntendedChange[activity]=TotalInternalIntendedChange[activity]*InternalSensitivity[activity]*FractionofWorkReleased[activity]
Unit: wu/Day
Description: The amount of work released that needs to be reprocessed due to internal intended changes.

RPTriggeredbyUPIntendedChange[activity]=SUM(IF THEN ELSE (PrecedenceRelationships[activity,preceding!]>-1, 1, 0)*TotalExternalIntendedChange[preceding!] *ActivityScope[activity]/ActivityScope[preceding!] *ExternalSensitivity[activity,preceding!] *FractionofWorkReleased[activity])
Unit: wu/Day
Description: Sum of needs for reprocess triggered by upstream activities. Actual amount is adjusted by work scope of associated activities, the activity's external sensitivity, and work released so far.
AvgTimeToPerceiveSchedulePressure[activity]=MAX( AdjustedTargetDuration [activity] /4* TableforProgressEffectonQM(PerceivedProgress [activity]),1)  
Unit: Day  
Description: Normally time to perceive schedule pressure is dependent on activity duration.

SKDLEffectOnPDY[activity]=TableForScheduleEffectOnPDY  
(SchedulePressure[activity])  
Unit: dimensionless  
Description: Referring to the effect of schedule pressure on work productivity.

TotalInternalIntendedChange[activity]=IntendedChangeReleaseRate[activity]+  
SUM(UPChangeAccomodateRate[activity,preceding!])  
Unit: wu/Day  
Description: Sum of intended changes made either before or after executing work.

ConverterforScope&Duration=1000  
Unit: wu/Day  

IntendedChangeRatio[activity]=IF THEN ELSE(gIntendedChangeRatioPJ>0,  
gIntendedChangeRatioPJ, IntendedChangeRatioforActivity[activity])  
Unit: Dimensionless  
Description: The ratio of adopting the change option facing problems either  
before or after executing work.

HiddenChangeAddressRate[preceding]=MIN(HiddenChange[preceding]/TIME STEP,  
VMAX(RPRequesttoUPRate[activity!,preceding] *ActivityScope [preceding]/ActivityScope[activity!]))  
Unit: wu/Day  
Description: Amount of hidden changes addressed by the downstream.  
Considering that different amount of change corrections on the same problem can  
be requested from the downstream, the actual amount of hidden change address  
rate is adjusted using work scope of associated activities. In addition, the biggest  
one is selected among multiple requests, since the amount of requests from the  
downstream can be different depending on downstream activities' management  
throughness, sensitivity, and scope.

RPAddressafterReleaseRate[activity]=MIN(WorkReleased[activity]/TIME STEP,  
(RPTriggeredbyExternalIntendedChange[activity]+RPTriggeredbyInternalIntende  
dChange[activity]+RPRequestfromDownstream[activity])*  
RippleEffectonWorkScope[activity])  
Unit: wu/Day  
Description: The total amount of work released that needs to be reprocessed.
Depending on activity characteristics, ripple effects on other work may be associated. Overlapped impacts are ignored for simplicity.

\[
\text{RequestForInformationRate}[\text{activity, preceding}] = \text{FractionofRFI}[\text{activity, preceding}] \times \begin{cases} \text{IF} & \text{THEN ELSE} \\
0 & \text{WorktoDo[activity]}/\text{MinWorkTime[activity]} \times \text{MIN(WorktoDo[activity]/MinWorkTime[activity], PotentialWorkRatefromResource[activity]}} \\
\end{cases}
\]

Unit: wu/Day
Description: Sum of RFIs to each upstream work that caused RFI. The amount is based on WorktoDo divided by MinWorkTime during reliability buffering, while it is based on the lesser of work availability-based amount and workforce availability-based amount during actual work period.

\[
\text{WorkOnceProcessed[activity]} = \text{INTEG} (\text{WorkRate[activity]} + \text{SUM(RequestForInformationRate[activity, preceding])}, 0)
\]

Unit: wu
Description: Referring to work that have been once processed. It includes pre-checking efforts but does not include work reprocesses, since experience on them is just duplicated.

Table for Learning Effect on Quality

\[
\text{TableforLearningEffectonQuality} = \begin{array}{c} 0((0,0)-(1,1.5)),(0,1.5),(0.25,1.3),(0.5,1), \\
(0.75,0.6),(1,0.1) \end{array}
\]

Unit: Dimensionless

Ripple Effect on Work Scope

\[
\text{RippleEffectonWorkScope[activity]} = 1
\]

Unit: Dimensionless
Description: Referring to the degree of ripple effect on work scope that is caused by changes.

Perceived Productivity

\[
\text{PerceivedProductivity[activity]} = \text{INTEG} ((\text{ActualProductivity[activity]} - \text{PerceivedProductivity[activity]})/\text{AvgQualityManagementTime[activity]} \times \text{ResourceCommisionStart[activity]}, \text{NormalProductivity[activity]})
\]

Unit: wu/(worker*Day)
Description: Productivity perceived by management, which is used to measure required workforce.

Progress Based Work Quality

\[
\text{ProgressBasedWorkQuality[activity]} = \text{MAX}(0.1 - (1 - \text{Reliability[activity]})) \times \text{TableforLearningEffectonQuality(ProgressInfluentialtoQuality[activity])}
\]

Unit: Dimensionless
Description: Normally, as workers get more information and experience, work quality increases. The pattern of quality increase varies depending on the initial quality level, for which in this equation, as work progresses quality increases by decreasing change rate.

Normal Completion Rate[activity] = Workforce[activity] * PerceivedProductivity[activity]
Unit: wu/Day
Description: Completion rate based on current workforce and perceived productivity.

SchedulePressure[activity] = SMOOTHI(IF THEN ELSE(ActivityBeingActivated [activity] = 0; OR: PerceivedProgress[activity] = 0, 1, FractionOfExpectedProgress[activity] / PerceivedProgress[activity]), AvgTimeToPerceiveSchedulePressure[activity], 1)
Unit: Dimensionless
Description: Schedule pressure perceived by workers during activity duration. It is not activated during buffering period.

QualityManagementThoroughness[activity]
Unit: Dimensionless
Description: Fraction of work corresponding to hidden upstream changes that were not discovered during pre-checking and as a result, will affect downstream work quality. Since this is inherited from upstream activities, there is no need for scope adjustment by using ActivityScope.

PendingWorkReleaseRate[activity, preceding] = IF THEN ELSE
(AvgHiddenChangeRPTimeinUP[preceding] = 0, 0, WorkPendingduringUPChangeRP[activity, preceding] / AvgHiddenChangeRPTimeinUP[preceding])
Unit: wu/Day
Description: Downstream work corresponding to corrected upstream changes that are returning to the downstream from each upstream.

PathPoolBuffer = MAX(0, InitialPathPoolBuffer - SUM(IsCriticalActivity?[activity] * MAX(0, ActivityDuration[activity] - AdjustedTargetDuration[activity])))
Unit: Day
Description: As construction progresses, the initial path pool buffer is getting consumed. Delays beyond AdjustedTargetDuration, which are actual ActivityDuration less AdjustedTargetDuration, will be absorbed in the path pool buffer.
ExtConcurrenceFromFFP[activity] = VMIN (IF THEN ELSE
  (PrecedenceRelationships[activity, preceding!] >= 2000,
   IF THEN ELSE (Time >= AdjustedTargetDuration[preceding!] +
     PrecedenceRelationships[activity, preceding!] - 2000 -
     AdjustedTargetDuration[activity] + ReliabilityStartTime - Raw[preceding!],
     TableForExternalConcurrence[activity, preceding!] (UpstreamProgress
     [preceding!]), 0), 1 ))
Unit: dimensionless
Description: External concurrency caused by finish-to-finish precedence
relationships. In case it has 2000 as its value, it refers FF without lag time. Once
its value is more than 2000, it will have lag time as much as its value less 2000.
Also, it will find the driving FF relationship among multiple FFs. Even after
start time is determined by this concurrency, work availability is continuously
controlled by TableForExternalConcurrence.

ReprocessIterationRelationships[activity, preceding!] = -1
Unit: Dimensionless
Description: It refers to the relationship that can create reprocess iteration in
upstream activities. None (-1), Existing (0)

RPTriggeredbyExternalIntendedChange[activity] = RPTriggeredbyUPIntendedChange[activity] +
  RPTriggeredbyDNIntendedChange[activity]
Unit: wu/Day
Description: The amount of work released that needs to be reprocessed due to
external changes.

UPChangeAccomodateRate[activity, preceding!] = IntendedChangeRatio[activity] *
  WorkAwaitingRFIReply[activity, preceding!] / AvgRFIReplyTime
Unit: wu/Day
Description: Downstream work corresponding to discovered hidden upstream
changes that will be accommodated in the downstream.

TotalExternalIntendedChange[activity] = TotalInternalIntendedChange[activity]
Unit: wu/Day
Description: Sum of intended changes made either before or after executing work
in the upstream.

HiddenChangeReprocesRate[activity] = IF THEN ELSE (WorktoDo[activity] = 0, 0,
  WorkRate[activity] * HiddenChangeRPAddressed[activity] / WorktoDo[activity])
Unit: wu/Day
Description: Hidden change reprocess rate, which is calculated by comparing
HiddenChangeRPAddressed with WorktoDo.
Table for Learning Effect on Quality:

\[
(0,0)-(1,1.5),(0,1.5),(0.25,1.3),(0.5,1),(0.75,0.6),
(1,0.1)
\]

Unit: Dimensionless

\[
\text{RPAddressRate}[\text{activity}] = (1 - \text{ActualWorkQuality}[\text{activity}]) - (1 - \text{ActualWorkQuality}[\text{activity}]) \times (1 - \text{QualityManagementThoroughness}[\text{activity}]) - (1 - \text{ActualWorkQuality}[\text{activity}]) \times \text{QualityManagementThoroughness}[\text{activity}] \times \text{IntendedChangeRatio}[\text{activity}] \times \text{WorkAwaitingQualityManagement}[\text{activity}] / \text{AvgQualityManagementTime}[\text{activity}]
\]

Unit: wu/Day

Description: Amount of work reprocessed rate through quality management, which is work done wrong less 'work done wrong but not discovered', and 'work done wrong and discovered but decided to release as it is by taking the change option'.

\[
\text{WorkOnceReleased}[\text{activity}] = \text{INTEG}(\text{WorkReleaseRate}[\text{activity}], 0)
\]

Unit: wu

Description: Amount of work that have been once released. It is distinguished from WorkReleased that can be decreased.

\[
\text{ReliabilityBuffer}[\text{activity}] = \text{MIN}(\text{ReductioninDuration}[\text{activity}], \text{INTEGER}(\text{IF THEN ELSE}(\text{VMAX}(\text{PotentialBufferingSize}[\text{activity, preceding!}]) > 0: \text{AND:} \text{VMAX}(\text{PotentialBufferingSize}[\text{activity, preceding!}]) < 1, 1, \text{VMAX}(\text{PotentialBufferingSize}[\text{activity, preceding!}])))))
\]

Unit: Day

Description: It determines the longest buffer size among candidate buffer sizes. But, any buffer size cannot exceed the duration reduction in the activity itself in order to avoid that adjusted duration becomes longer than the original one.

\[
\text{AvgHiddenChangeRPTime}[\text{activity}] = \text{IF THEN ELSE}(\text{HiddenChangeReprocesRate}[\text{activity}] > 0, \text{HiddenChangeRPAddressed}[\text{activity}] / \text{HiddenChangeReprocesRate}[\text{activity}], 0) + \text{IF THEN ELSE}(\text{ReprocessedHiddenChangeReleaseRate}[\text{activity}] > 0, \text{ReprocessedHiddenChangeReprocesed}[\text{activity}] / \text{ReprocessedHiddenChangeReleaseRate}[\text{activity}], 0)
\]

Unit: Day

Description: Average time to reprocess hidden changes, which is calculated based on residual time formula.

\[
\text{HiddenChangeReleaseRate}[\text{activity}] = \text{WorkReleaseRate}[\text{activity}] \times (1 - \text{ActualWorkQuality}[\text{activity}]) \times (1 - \text{QualityManagementThoroughness}[\text{activity}]) / (\text{ActualWorkQuality}[\text{activity}] + (1 - \text{ActualWorkQuality}[\text{activity}]) \times (1 - \text{QualityManagementThoroughness}[\text{activity}]) + (1 - \text{ActualWorkQuality}[\text{activity}])
\]
IntendedChangeRatio[activity]

Unit: wu/Day
Description: Amount of hidden change contained in work release rate


Unit: Dimensionless
Description: It refers to the fraction of discovered upstream hidden changes that can affect downstream work quality.

Unit: Day
Description: Average time that is taken in each upstream to correct the requested change correction.

Unit: wu/Day
Description: Intended change contained work release rate

FractionofTotalRFI[activity] = SUM(FractionofRFI[activity, preceding!])
Unit: Dimensionless
Description: While RequestForInformationRate has two dimensional subscripts, Description: WorkRate has one dimensional vector. Thus, it is needed to sum FractionofRFIs for each upstream.

WorktoDo[activity] = INTEG (InitialWorkIntroduceRate[activity] + SUM(PendingWorkReleaseRate[activity, preceding!]) + SUM(UPChangeAccomodateRate[activity, preceding!]) - SUM(RequestForInformationRate[activity, preceding!]) - WorkRate[activity] + RPAddressRate[activity] + RPAddressafterReleaseRate[activity], 0)
Unit: wu
Description: Work ready to be done

ActivityFinished[activity]=SAMPLE IF TRUE(FractionofWorkReleased[activity] >0.99, 0, 1)
Unit: Dimensionless
Description: Distinguished from ActivityReallyFinished. Once finished, it will not change its value. In fact, even though there can be some more work caused by reprocess iterations, the official completion time is normally decided by the time in which target scope is once achieved.

ActivityFinishTime[activity]=INTEGER(ActivityFinishTime-Raw[activity])
Unit: Day

ActivityFinishTime-Raw[activity]= INTEG (IF THEN ELSE(ActivityFinished [activity]=1, 1 , 0 ),0)
Unit: Day

ProjectProgress=SUM(WorkReleased[activity])/ProjectScope
Unit: Dimensionless

ProjectScope= SUM(ActivityScope[activity])
Unit: wu

QualityManagementThoroughness[activity]=0.5
Unit: Dimensionless
Description: Thoroughness in doing quality control and management. It refers to the fraction of discovered changes in total changes that have occurred.

AdjustedTargetDuration[activity]=OriginalDuration[activity]-ReductioninDuration[activity]
Unit: Day
Description: Adjusted duration after taking off contingency buffers

AvgRFIReplyTime[activity]=20
Unit: Day
4 weeks * 5 working days per week
Description: Referring to average time spent to answer RFIs, which vary depending on the duration of associated activities.

pjLaborHoursperDay=WorkhoursperDay*SUM(EffectiveWorkForce[activity]*WorkBeingDone[activity])
Unit: worker*hour/Day

RemainingPathPoolBufferRatio=IF THEN ELSE(InitialPathPoolBuffer= 269
0,1 ,PathPoolBuffer/InitialPathPoolBuffer )
   Unit: Dimensionless
   Description: Fraction of remaining path pool buffer at a given time

RemainingWork[activity]=ActivityScope[activity]-WorkReleased[activity]-
   WorkAwaitingQualityManagement[activity]
   Unit: wu
   Description: Remaining work to be done at a given time. It is used to estimate the
   remaining time and required workforce to meet the schedule.

ReprocessedHiddenChangeReleaseRate[activity]=IF THEN ELSE
   (WorkAwaitingQualityManagement[activity]=0,0,WorkReleaseRate
   [activity]*HiddenChangeReprocessed[activity]/
   WorkAwaitingQualityManagement[activity])
   Unit: wu/Day
   Description: Reprocessed hidden change release rate, which is calculated
   by comparing HiddenChangeReprocessed with WorkAwaitingQualityManagement.

RPRequestfromDownstream[activity]=HiddenChangeAddressRate[activity]
   Unit: wu/Day
   Description: The amount of work released that needs to be reprocessed due to
   change correction requests from downstream.

ExtConcurrenceFromSSP[activity]=VMIN(IF THEN ELSE
   (PrecedenceRelationships[activity,preceding!]>=1000:AND:
   PrecedenceRelationships[activity,preceding!]<2000,IF THEN ELSE
   (Time>=ReliabilityStartTime-Raw[preceding!]+MAX(TIME STEP,
   PrecedenceRelationships[activity,preceding!]-1000),
   TableForExternalConcurreence[activity,preceding!]
   (UpstreamProgress[preceding!]) , 0 ) ,1 ))
   Unit: dimensionless
   Description: External concurrency caused by start-to-start precedence
   relationships. In case it has 1000 as its value, it refers SS without lag time. Once
   its value is more than 1000, it will have lag time as much as its value less 1000.
   Also, it will find the driving SS relationship among multiple SSs. Even after start
   time is determined by this concurrency, work availability is continuously
   controlled by TableForExternalConcurreence.

ExternalSensitivity[activity,preceding]=1
   Unit: Dimensionless
Description: The degree of how the activity work quality and scope are sensitive to external changes. For quantification purposes, the degree of sensitivity is set assuming that the impacted activity is completely done.

\[
\text{ScheduledActivityCompletion}[\text{activity}] = \text{ActualActivityStartTime}[\text{activity}] + \text{AdjustedTargetDuration}[\text{activity}]
\]

Unit: Day

\[
\text{WorkforceUtilizationRatio} = \text{IF THEN ELSE}(\text{ProjectFinished}=0,1, \text{IF THEN ELSE}(\text{SUM}(\text{PotentialWorkRateFromResource}[\text{activity}])))
\]

Unit: Dimensionless

\[
\text{PerceivedWorkDone}[\text{activity}] = \text{WorkAwaitingQualityManagement}[\text{activity}] + \text{WorkReleased}[\text{activity}]
\]

Unit: wu

\[
\text{WorkPendingDuringUPChangeRP}[\text{activity, preceding}] = \text{INTEG} \left( \text{RPRequesttoUPRate}[\text{activity, preceding}] - \text{PendingWorkReleaseRate}[\text{activity, preceding}], 0 \right)
\]

Unit: wu

Description: Work pending during upstream change correction in the upstream.

\[
\text{WorkInProcess}[\text{activity}] = \text{SUM}(\text{WorkAwaitingRFIReply}[\text{activity, preceding}]) + \text{WorkPendingDuringUPChangeRP}[\text{activity, preceding}] + \text{WorkToDo}[\text{activity}] + \text{WorkAwaitingQualityManagement}[\text{activity}] + \text{WorkReleased}[\text{activity}]
\]

Unit: wu

Description: Work in process at a given time.

\[
\text{WorkReleaseRate}[\text{activity}] = \left(\text{ActualWorkQuality}[\text{activity}] + (1 - \text{ActualWorkQuality}[\text{activity}]) \times (1 - \text{QualityManagementThoroughness}[\text{activity}]) + (1 - \text{ActualWorkQuality}[\text{activity}]) \times \text{QualityManagementThoroughness}[\text{activity}] \times \text{IntendedChangeRatio}[\text{activity}] \right) / \text{AvgQualityManagementTime}[\text{activity}]
\]

Unit: wu/Day

Description: Amount of work release rate through quality management, which is the sum of work done correctly, work done wrong but not discovered, work done wrong and discovered but decided to release as it is by taking the change option.

\[
\text{HiddenChangeReprocessed}[\text{activity}] = \text{INTEG} \left( \text{HiddenChangeReprocesRate}[\text{activity}] - \text{ReprocessedHiddenChangeReleaseRate}[\text{activity}], 0 \right)
\]
Unit: wu
Description: Hidden changes that have been reprocessed

InternalSensitivity[activity]=1
Unit: Dimensionless
Description: Degree of how work scope is sensitive to internally made intended changes.

RPRequesttoUPRate[activity, preceding]=(1-IntendedChangeRatio[activity])*
WorkAwaitingRFIReply[activity, preceding]/AvgRFIReplyTime
Unit: wu/Day
Description: Downstream work corresponding to discovered hidden upstream changes that will be returned to upstream for correction. Requests on correction are forwarded to each upstream.

WorkReleased[activity]= INTEG (WorkReleaseRate[activity]-
RPAddressafterReleaseRate[activity],0)
Unit: wu
Description: Work released that allows the downstream work to start

UpstreamProgress[activity]=FractionofWorkReleasedforDNtoStart[activity]
Unit: Dimensionless

WorkRate[activity]=ResourceCommisionStart[activity]*(1-
FractionofTotalRFI[activity])*MIN(WorktoDo[activity]/MinWorkTime
[activity], PotentialWorkRatefromResource[activity])
Unit: wu/Day
Description: Determined by potential work rate or minimum time to do work.

WorkAwaitingRFIReply[activity, preceding]= INTEG (RequestForlnformationRate
[activity, preceding]-UPChangeAccomodateRate[activity, preceding]-
RPRequesttoUPRate[activity, preceding],0)
Unit: wu
Description: Work waiting for answers to RFI

Unit: Day

ResourceCommisionStart[activity]
Unit: worker
Description: Actually available workforce after considering flexible workforce control and overtime.

Preceding =activity
ActivityFinishTime\(_{fromPrecedesorStart}[activity]\) = V\(_\text{MAX}\) (IF THEN ELSE \\
(PrecedenceRelationships\_[activity, preceding!] > -1, ActivityFinishTime \_[activity] - ActualActivityStartTime\_[preceding!], 0)) \\
Unit: Day \\
Description: Measuring the longest duration among durations between upstream start time and downstream finish time. It is used to find local optimum buffer size.

ProgressBasedExpectedProductivity\_[activity] = NormalProductivity\_[activity] * \\
TableForProgressBasedProductivity\_[activity] (IF THEN ELSE \\
(ProductionType\_[activity] = 0, 1 - FractionOfExpectedProgress\_[activity], \\
FractionOfExpectedProgress\_[activity])) \\
Unit: wu/(Day*worker) \\
Description: Expected productivity based on progress patterns.

Unit: Day \\
Description: Referring to actual activity start time. Once reliability buffering is applied, the buffering size needs to be added.

ProgressBasedProductivity\_[activity] = NormalProductivity\_[activity] * \\
TableForProgressBasedProductivity\_[activity] (IF THEN ELSE \\
(ProductionType\_[activity] = 0, 1 - PerceivedProgress\_[activity], \\
PerceivedProgress\_[activity])) \\
Unit: wu/(Day*worker) \\
Description: Productivity based on progress patterns.

ProjectFinishTime = INTEGER(ProjectFinishTime-Raw) \\
Unit: Day \\
ProjectFinishTime-Raw = INTEG (IF THEN ELSE(ProjectFinished=1, 1 , 0 ),0) \\
Unit: Day \\
BufferingRatio\_[activity] = IF THEN ELSE \\
(gFractionOfBufferingPJ > 0, gFractionOfBufferingPJ, \\
FractionOfBuffering\_[activity]) \\
Unit: Dimensionless \\
Description: Actual fraction of buffering for an activity.

SlowEvolution\_[activity] = 0 \\
Unit: Dimensionless \\
Description: Productivity is initially low but increases as construction
progresses due to learning effect.

KnownContingencyFactorPJ=0.2
Unit: Dimensionless
Description: Contingency known to be incorporated into the duration of all activities in a project.

Reliability[activity]= 0.8
Unit: Dimensionless
Description: 1-R represents the possibility of creating changes in average.

CumulativeLaborHoursPJ= INTEG (pjLaborHoursperDay, 0)
Unit: worker*hour

IsCriticalActivity?[activity]= 1
Unit: Dimensionless
Description: Determining if an activity is on critical path

gFractionOfBufferingPJ= GAME (FractionOfBufferingPJ)
Unit: Dimensionless
Description: For gaming

ExpectedWorkRate[activity]=IF THEN ELSE
(FractionOfExpectedProgress[activity]>=1,0,
EffectiveWorkForce[activity]*ProgressBasedExpectedProductivity[activity]/(1-ContingencyToBeTakenOff[activity]))
Unit: wu/Day
Description: Expected work rate based on progress-based productivity. It is also adjusted according to whether or not contingency buffers are taken off, which increases expected work rate and eventually create schedule pressure effect for buffering. In other word, by taking off contingency buffer, an activity can benefit from schedule pressure to some extent.

ExtConcurrenceFromFSP[activity]=VMIN(IF THEN ELSE
(PrecedenceRelationships[activity, preceding!]>0:AND:
PrecedenceRelationships[activity, preceding!]<1000,IF THEN ELSE
(Time>=UpstreamFinishTimeForDownstreamtoStart[preceding!]+
PrecedenceRelationships[activity, preceding!],
TableForExtemalConcurreence[activity, preceding!](UpstreamProgress[preceding!]) , 0 ),1 ))
Unit: dimensionless
Description: External concurrency caused by finish-to-start precedence
relationships. In case it has 0 as its value, it refers FS without lag time. Once its value is more than 0, it will have lag time as much as its value. Also, it will find the driving FS relationship among multiple FSs. Even after start time is determined by this concurrency, work availability is continuously controlled by TableForExternalConcurrence.

FastEvolution[activity]=0  
Unit: Dimensionless  
Description: Productivity is initially high but decreases as construction progresses due to increased complexity.

OverTime[activity]=IF THEN ELSE(ResourceCommisionStart[activity]=0,1,IF THEN ELSE(gWillingnessToAdoptOvertime>0 , gWillingnessToAdoptOvertime  
*(TableForOverTime(RequiredWorkRate[activity]/ 
NormalCompletion Rate[activity])-1)+1,1 ))  
Unit: dimensionless  
Description: Degree of overtime to be applied

FractionOfBufferingPJ=0  
Unit: Dimensionless  
Description: Fraction of buffering for a project, which assigns to activities in a uniform way.

TableForExternalConcurrence[activity,preceding]([(0,0)-(10,1)],(0,0.1),(0.99,1),(10,1))  
Unit: dimensionless  
Description: Dependency pattern that constrain downstream work availability

PotentialBufferingSize[activity,preceding]=IF THEN ELSE  
(PrecedenceRelationships[activity,preceding]>-1, 1 , 0)*  
ContingencyToBeTakenOff[preceding]* OriginalDuration[preceding]*  
BufferingRatio[activity]  
Unit: Day  
Description: Reliability buffer size is determined by multiplying BufferingRatio of the activity with the original duration and applied contingency ratio of its upstream activities. Thus, with this equation, all possible buffer sizes associated with upstream activities in precedence relationship can be calculated.

FractionofWorkReleasedforDNtoStart[activity]=MIN(1,WorkOnceReleased[activity]/ 
ActivityScope[activity])  
Unit: Dimensionless  
Description: It is used to determine the start time of downstream activities,
being distinguished from FractionofWorkReleased. The fraction of work released that controls the downstream work start should not be decreased by work reprocess after being released.

\[ g_{\text{InitialStaffingRatio}} = \text{GAME}(\text{InitialStaffingRatio}) \]

Unit: Dimensionless
Description: For controlling InitialStaffingRatio during gaming

\[ g_{\text{ReliabilityBufferingActivated}} = \text{GAME}(\text{ReliabilityBufferingActivated}) \]

Unit: Dimensionless
Description: For gaming

\[ g_{\text{TimeToIncreaseWorkforce}} = \text{GAME}(\text{TimeToIncreaseWorkforce}) \]

Unit: Day
Description: For controlling TimeToIncreaseWorkforce during gaming

\[ g_{\text{TimeForQM}[\text{activity}]} = \text{GAME}(\text{TimeForQM}[\text{activity}]) \]

Unit: Day
Description: For controlling TimeForQM during gaming

\[ g_{\text{WillingnessToAdoptOvertime}} = \text{GAME}(\text{WillingnessToAdoptOvertime}) \]

Unit: Dimensionless
Description: For controlling WillingnessToAdoptOvertime during gaming

\[ g_{\text{WillingnessToControlHeadCount}} = \text{GAME}(\text{WillingnessToControlHeadCount}) \]

Unit: Dimensionless
Description: For controlling WillingnessToControlHeadCount during gaming

\[ \text{WorkforceAdjustRate}[\text{activity}] = \text{IF THEN ELSE}(g_{\text{WillingnessToControlHeadCount}} > 0, \text{IF THEN ELSE}(\text{DesiredWorker}[\text{activity}] > \text{Workforce}[\text{activity}], (\text{DesiredWorker}[\text{activity}] - \text{Workforce}[\text{activity}]) / g_{\text{TimeToIncreaseWorkforce}}, (\text{DesiredWorker}[\text{activity}] - \text{Workforce}[\text{activity}]) / (\text{TimetoDecreaseWorkforce})^* g_{\text{WillingnessToControlHeadCount}}, 0) \]

Unit: worker/Day
Description: Workforce adjust rate

\[ \text{WorkhoursperDay} = 8 \]

Unit: hour/Day
InitialPathPoolBuffer = \text{SUM}"IsCriticalActivity?"[activity!]*
(ReductioninDuration[activity!]-ReliabilityBuffer[activity!]))

Unit: Day
Description: Path pool buffer consists of saved durations in individual activities on the critical path, which are ReductioninDuration less ReliabilityBuffer.

InitialWorkers[activity] = IF THEN ELSE(gWillingnessToControlHeadCount=0, 1, gInitialStaffingRatio)*ActivityScope [activity]/ (NormalProductivity[activity]*OriginalDuration[activity])

Unit: worker
Description: Initially given workforce

UpstreamFinishTimeForDownstreamtoStart[activity] = SAMPLE IF TRUE(
ActivityFinished[activity]=1, Time+TIME STEP, 0)

Unit: Day

ProjectFinished = IF THEN ELSE(ProjectProgress>=0.99, 0, 1)

Unit: Dimensionless

PrecedenceRelationships[activity, preceding]=-1

Unit: Day
Description: Precedence relationships with upstream activities. None (-1), FS (0-1000), SS (1000-2000), FF (2000-)

ResourceCommisionStart[activity] = IF THEN ELSE
(Time=ReliabilityStartTime[activity]+ReliabilityBuffer
[activity]:AND:ActivityStarted[activity]=1, 1, 0)

Unit: Dimensionless
Description: The signal for resource to be commissioned. It is used to distinguish the actual activity start from buffering start, when reliability buffering is applied.

WorkAvailable[activity] = \text{MAX}(0, \text{MIN}(TotalExternalConcurrence[activity],
WorkAvailabilityFromIntConcurrency[activity]*ActivityScope[activity]-
WorkInProcess[activity]))

Unit: wu
Description: Available work at a given time, which is the lesser of available work governed by internal dependency and that by external dependency.

WorkBeingDone[activity] = ResourceCommisionStart[activity]*
ActivityReallyFinished[activity]

Unit: Dimensionless
Description: Once an activity is officially finished, there is a case in which some more work need to be done by reprocess requests. It is used in measuring applied workforce after officially finishing an activity.

\[
\text{ContingencyToBeTakenOff}[\text{activity}] = \begin{cases} 
\text{IF THEN ELSE}(\text{KnownContingencyFactorPJ} > 0, \\
\text{KnownContingencyFactorPJ}, \text{KnownContingencyFactor}[\text{activity}]) 
\end{cases} 
\times \text{gReliabilityBufferingActivated} 
\]

Unit: Dimensionless

Description: Fraction of contingency in activity duration. Once reliability buffering is activated, the contingency buffer that has been applied is taken off.

\[
\text{ReductioninDuration}[\text{activity}] = \text{OriginalDuration}[\text{activity}] \times \text{ContingencyToBeTakenOff}[\text{activity}] 
\]

Unit: Day

Description: Reduction in duration as a result of taking off contingency factor

\[
\text{AvgQualityManagementTime}[\text{activity}] = \max(\text{gTimeforQM}[\text{activity}] \times \text{TableforProgressEffectonQM}(\text{PerceivedProgress}[\text{activity}]), 1) 
\]

Unit: Day

Description: Average time for quality management by internal or external inspectors. As construction progresses, the time becomes shorter.

\[
\text{ReliabilityStartTime}[\text{activity}] = \text{INTEGER}("\text{ReliabilityStartTime-Raw}"[\text{activity}]) 
\]

Unit: Day

\[
\text{TotalExternalConcurrence}[\text{activity}] = \min(\text{ExtConcurrenceFromFFP}[\text{activity}], \min(\text{ExtConcurrenceFromFSP}[\text{activity}], \text{ExtConcurrenceFromSSP}[\text{activity}])) 
\]

Unit: Dimensionless

Description: It finds the dependency that constrains work availability of the activity among multiple upstreams having various precedence relationships.

\[
\text{TableforProgressEffectonQM} = [(0, 0) - (1, 1)], (0, 1), (0.5, 1), (0.75, 0.9), (0.9, 0.5), (1, 0.1)] 
\]

Unit: Dimensionless

Description: As construction progresses, quality management time is getting shorter.

\[
\text{Workforce}[\text{activity}] = \text{INTEG} (\text{WorkforceAdjustRate}[\text{activity}], \text{InitialWorkers}[\text{activity}]) 
\]

Unit: worker

Description: Level of workforce for an activity at a given time.
InitialStaffingRatio = 0.5  
Unit: Dimensionless  
Description: The ratio of initial staffing

Unit: Dimensionless

ExpectedProgress[activity] = INTEG (ExpectedWorkRate[activity], 0)  
Unit: wu  
Description: Expected progress of work, which is compared to actual progress in measuring schedule pressure.

TimeRemainingToCompletion[activity] = MAX(1, ScheduledActivityCompletion[activity] - Time)  
Unit: Day

FatigueEffectonQuality[activity] = IF THEN ELSE  
(ResourceCommisonStart[activity] = 0, 1, TableForEffectOfFatigueOnQuality(Fatigue[activity]))  
Unit: dimensionless

TimeToGetFatigued = 14  
Unit: Day

TableForProgressBasedProductivity[activity] = [(0,0.5)-(1,1.5)], (0.05,0.084,0.54), (0.15,0.61), (0.22,0.74), (0.284211,0.890351), (0.33,1.05), (0.39,1.20), (0.48,1.32), (0.61,1.42), (0.75,1.46), (1,1.5)]  
Unit: dimensionless

NormalProductivity[activity] = 10  
Unit: wu/(Day*worker)  
Description: It is introduced just for simulation purpose.

TableForFlexibleComplexity = [(0,0)-(1,1)], (0.49,0.89), (0.70,0.50), (1,0.25)  
Unit: dimensionless

Unit: wu/Day  
Description: Work rate based on available workforce.

TimeToDecreaseWorkforce = 7  
Unit: Day
RequiredWorkRate[activity]=RemainingWork[activity]/
TimeRemainingToCompletion[activity]
Unit: wu/Day

TableForEffectOfFatigueOnProductivity([(0,0.75)-(2,1)],(0,1),(1,1),(1.11178,0.98),
(1.23,0.96),(1.31,0.92),(1.42,0.85),(1.58,0.80),(1.80,0.75),(2,0.75),(100,0.75))
Unit: Dimensionless

Fatigue[activity]= SMOOTHI(OverTime[activity],TimeToGetFatigued,1)
Unit: dimensionless

TimetoIncreaseWorkforce=7
Unit: Day

WillingnessToControlHeadCount= 0
Unit: dimensionless
Description: Willingness to control workforce by adding or decreasing.

TableForScheduleEffectOnPDY([(0,0)-(2,2)],(0,0.5),(0.5,0.5),(0.7,0.6),(0.8,0.7),
(0.9,0.85),(1,1),(1.25,1.25),(1.5,1.35),(2,1.5))
Unit: Dimensionless

TableForEffectOfFatigueOnQuality([(1,0.75)-(2,1)],(0,1),(1,1),(1.163,0.98),(1.25,0.97),
(1.38,0.92),(1.47,0.88),(1.54,0.82),(1.54,0.82),(1.61,0.79),
(1.69,0.77),(1.83,0.75),(2,0.75))
Unit: Dimensionless

WillingnessToAdoptOvertime=0
Unit: dimensionless
Description: Willingness to control workforce by adopting overtime.

TableForOverTime([(0,0.8)-(3,2)],(0,1),(1,1),(1.81269,1.60526),(2.5,2),(100,2))
Unit: dimensionless

FractionOfHiddenChangeinWorkReleased[activity]=IF THEN ELSE
(WorkReleased[activity]=0 , 0 , HiddenChange[activity]/
WorkReleased[activity])
Unit: Dimensionless
Description: Fraction of hidden changes in work that have been released.

HiddenChangeRPAddressed[activity]= INTEG (+HiddenChangeAddressRate [activity]-
HiddenChangeReprocesRate[activity],0)
Unit: wu
Description: Hidden changes that have been addressed
WorkAvailabilityFromIntConcurrency[activity] = TableForInternalConcurrence[activity] (PerceivedProgress[activity])
Unit: Dimensionless
Description: Even if work is not released through quality management process, work done will allow the following internal work to proceed. For this reason, instead of actual progress, perceived progress is used.

TableForInternalConcurrence[activity] = \(((0,0)-(1,1)),(0,1),(0.99,1),(10,1)\)
Unit: dimensionless

HiddenChange[activity] = INTEG (HiddenChangeReleaseRate[activity] - HiddenChangeAddressRate[activity], 0)
Unit: wu
Description: Hidden changes in work that have been released. It increases by inflow of hidden change generation and it decreases by hidden change correction requests from the downstream.

KnownContingencyFactor[activity] = 0.2
Unit: Dimensionless
Description: Contingency known to be incorporated into the duration of activities.

FractionOfBuffering[activity] = 0
Unit: Dimensionless
Description: Fraction of buffering for an activity

ReliabilityStartTime-Raw[activity] = INTEG (IF THEN ELSE (ActivityStarted[activity] = 0, 1, 0), 0)
Unit: Day
Description: As an activity starts, reliability buffering can start, if activated. This equation find the buffering start time.

ReliabilityBufferingActivated = 1
Unit: Dimensionless
Description: Making reliability buffering activated

NumberOfActivity = 3
Unit: Dimensionless

ActivityStarted[activity] = SAMPLE IF TRUE(WorkAvailable[activity] > 1, 1, 0)
Unit: Dimensionless
Description: Indicating whether an activity is started.

OriginalDuration[activity] = 50
Unit: Day
Description: Initially given duration

Unit: wu/Day
Description: The reason for using TIME STEP is to make available tasks ready to work without delays. In fact, there is a delay in this process. But, compared to delays caused by work dependencies, it can be ignorable.

PerceivedProgress[activity] = PerceivedWorkDone[activity] / ActivityScope[activity]
Unit: Dimensionless
Description: Work progress perceived by workers.

ActivityReallyFinished[activity] = IF THEN ELSE
(FractionofWorkReleased[activity] >= 0.99, 0, 1)
Unit: Dimensionless
Description: Indicating that an activity is really finished.

FractionofWorkReleased[activity] = WorkReleased[activity] / ActivityScope[activity]
Unit: Dimensionless

MinWorkTime[activity] = 1
Unit: Day
Description: Minimally required time to do tasks in the activity

USErrorImpactonQuality[activity] = 1
Unit: Dimensionless

WorkAwaitingQualityManagement[activity] = INTEG (WorkRate[activity] - WorkReleaseRate[activity] - RPAddressRate[activity], 0)
Unit: wu
Description: Work done but not be checked though quality management process.

FINAL TIME = 150
Unit: Day
Description: The final time for the simulation.

INITIAL TIME = 0
Unit: Day
Description: The initial time for the simulation.

TIME STEP = 0.25
Unit: Day
Description: The time step for the simulation.