HYBRID SIMULATION FOR CONSTRUCTION OPERATIONS

Hani S. H. Alzraiee

A Thesis

In the Department

of

Building, Civil, and Environmental Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Building Engineering) at

Concordia University

Montreal, Quebec, Canada

August 2013

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Hani S.H. Alzraiee**

Entitled: **Hybrid Simulation for Construction Operations**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Building Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. D. Dysart-Gale

______________________________ External Examiner
Dr. S.M. Abourizk

______________________________ External to Program
Dr. G. Gopakumar

______________________________ Examiner
Dr. Z. Zhu

______________________________ Examiner
Dr. A. Bagchi

______________________________ Thesis Co-Supervisor
Dr. T. Zayed

______________________________ Thesis Co-Supervisor
Dr. O. Moselhi

Approved by ______________________________
Dr. M. Elektorowicz, Graduate Program Director

August 12, 2013

____
Dr. C. Trueman, Interim Dean
Faculty of Engineering & Computer Science

ABSTRACT

Hybrid Simulation for Construction Operations

Hani Alzraiee, Ph.D.

Concordia University, 2013

Developing realistic and unbiased simulation models for construction operations require addressing the operational and strategic decision making levels. The dynamics and feedback processes observed in construction systems are responsible for the real behaviour of such systems and drive the needs for hybrid and integrated simulation tools. The dominant simulation methods such as discrete event simulation (DES) and system dynamics (SD) are limited individually of capturing all the significant construction operation aspects that are responsible for generating the behaviour of realistic models. Therefore, this thesis presents a hybrid simulation method for simulating construction operations by utilizing the joint powerful features of the DES and SD methods.

The proposed method provides a framework to integrate DES and SD on single computational platform. Developing a hybrid simulation model commences by decomposing the construction project into units, form which simulation models (e.g. DES or SD) are developed. A unidirectional variables interaction from DES to SD models is used. The interfacing process among simulation models is achieved by defining three variables: sender, interface, and receiver. The mechanism that controls data mapping processes between variables is outlined in a new developed synchronization method. The variables interaction protocol is described using formalism. Finally, a Hybrid Simulation Application (HiSim) is coded in VB.NET to demonstrate a sequential implementation of the developed method.

A real-world earthmoving project is modeled and simulated to test the developed hybrid simulation method. The hybrid simulation structure uses unidirectional and sequential interactions between the components of DES and SD models. The simulation is run under three scenarios, is able to predict the real project completion duration with 92% accuracy, and captures the influences of the context level variables. The findings are expected to enhance hybrid simulation applications in construction and to allow for better understanding of the impact of various internal and external factors on the project schedule and its productivity performance.

To my adorable mother Mariam, beloved grandmother Marzouka, and dear father Suliman and brother Hussein,

and

to my wife Neda, son Suliman and daughter Sama

Acknowledgment

"Humanism is the only - I would go so far as saying the final- resistance we have against the inhuman practices and injustices that disfigure human history." ― Edward W. Said

All praise and thanks are due to Allah for giving me the patience and perseverance to successfully accomplish my Ph.D. study.

This thesis is dedicated to my father Suliman, mother Mariam, grandmother Marzouka, for their endless support and encouragement throughout my life. Their unconditional love has been source of inspiration. Thanks for being the type of parents that they are, I could never achieve this goal without their endless encouragement, advises and help.

I am greatly thankful to my mentors, Professor Osama Moselhi and Professor Tarek Zayed, for their inexhaustible supports and valuable advises throughout my studies in Concordia University. I have learned a lot from them and it was a great honor to work under their supervision. Their sharp comments have always inspired me to work hard. Working with both of you is the most valuable and exciting experience in my Concordia University life.

I am also grateful to the other shining starts in my life, my wife Neda. The love and support you gave me has been essential in accomplishing this thesis. She graced our home with much love and compassion and gave me the treasure of having an adorable family. I also owe a lot to my little children Suliman and Sama for all the time I took away from them to work in this research

Many thanks to all my colleagues in the Automation and Construction Laboratory.

Hani Alzraiee

August, 2013, Montreal, Quebec, Canada

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

DES	Discrete Event Simulation
SD	System Dynamics
HiSim	Hybrid Simulation
SPM	Strategic Project Management
OPM	Operational project Management
CYCLONE	Cyclic Operation Network
DISCO	Dynamic Interface Simulation for Construction Operation
RBM	Resource-Based Modeling
RISim	Resource-Interacted Simulation modeling
HSM	Hierarchical Simulation Modeling
CPM	Critical Path Method
WBS	Work Breakdown Structure
SPS	Special Purpose Systems
KEYSTONE	Knowledge Discovery Based Simulation System)
HTTA	Hauler's Travel Time Application
CLD	Causal Loop Diagrams
SFD	Stocks and Flows Diagram
DEVS	Discrete Event System Specifications)
DESS	Differential Equation System Specification)
HLA	High Level Architecture
OMT	Object Model Template
SPSM	Software Process Simulation Modeling
DSM	Dependency Structure Matrix

DPM	Dynamic Planning Methodology
CTM	Conservative Time Management
OTM	Optimistic Time Management
LP	Logical Processes
GVT	Global Virtual Time
TW	Time Warp
RTI	Run Time Infrastructure
TB	Time Bucket
I	Inputs
O	Outputs
M	Modules
S	Synchronization process
FPC	Fleet Production and Cost
SM-3	Saint Marguerite Three

CHAPTER 1

INTRODUCTION

1.1 General

Construction projects are by nature heterogeneous, with inherent dynamics in their behavior. Such characteristics increase the degree of uncertainty in predicting project behavior and outcomes. Planners and managers are continuously facing difficulties in testing project plans and their execution scenarios. Modeling and simulation provide a ground for overcoming such difficulties due to the powerful capabilities to mimic the real behavior of projects (Halpin et al. 2003). Developing a successful simulation model for construction project should take into account: 1) decision level; 2) system complexity; 3) type of variable; and 4) relationship among variables (Alzraiee et al. 2012a). Decision levels in construction projects are divided into: 1) strategic and 2) operational (Lyneis et al. 2001). The strategic definition in this thesis is different from the definition pertaining to organizational management. The strategic level means achieving the project's set objectives within the project policy frame. This involves the adjustment of certain parameters like cost, resources, and time to meet previously set goals (Rodrigues and Bowers 1996). On the other hand, the operational level is viewed here as the actions taken to meet the project goals set at the strategic level; it focuses on the daily operational details at the micro level of the project. Construction projects are comprised of discrete and continuous variables. The system behavior is mainly generated based on the various mutual interactions of these variables. The relationships among construction system variables are in a form of causal-effect feedback loops, where a variable (cause) affects other variables positively, or negatively (effect). The dynamic behavior inherent in construction projects is mainly attributed to the causal-effect feedback loops.

Project management is one of the most important and most poorly understood areas of management and cost overruns of 100 to 200% are common (Sterman 1992). Under the stress of the industry's changing environment and the complexity of construction operations, a better understanding of the construction system and the interactions of external-internal environments are a necessity. An automated tool that has the capability of integrating the heterogeneous aspects of a project on a single computational platform is needed to deal with increased complexity of construction management issues. This allows for testing different management hypotheses, gives insight into the interactions of many factors, and ultimately selecting the best courses of action. Such a tool can empower managers by predicting the future scenarios that might be encountered during project execution. Therefore, managers can make informed decisions and change policies in advance of problem occurrence.

1.2 Computer Simulation

The decision-making process is an essential part of any construction operation. Simulation is used as a tool to assist construction managers in making informed decisions (Zayed and Halpin 2001). When projects are large or complex, they become increasingly more complicated to manage using existing tools (AbouRizk 2010). Simulation is widely regarded as an effective tool for construction process analysis due to its ability to handle the complexity and the uncertainty inherent in construction (AbouRizk and Hajjar 1998; Halpin et al 2003). The most appreciated computer modeling-simulation feature is the ability to monitor project variables' interactions in a controlled and safe environment. This allows the creation of models in the virtual environment, which helps in providing better insight into project interactions and can signal flaws. However, despite these benefits, simulation applications in construction modeling are still limited due to difficulties in learning and applying simulation languages to the construction industry (Touran 1990; Sawhney and AbouRizk 1996; Oloufa et al. 1998). Furthermore, there is a widespread skepticism among construction industry practitioners to trust simulation analyses. Construction planners and analysts, who are typically well familiar with the actual construction operations, are reluctant to base their decisions solely on the statistical text and graphical chart output provided by most simulation systems (Ioannou and Martinez 1999). This constitutes what has become known as the ‘‘black-box effect’’ and is a major impediment in validating and verifying simulation models. The resulting lack of credibility hinders the widespread use of simulation in the construction industry (Kamat and Martinez 2001).

Different methods are available to develop computer simulation models. This research focuses on only two methods, Discrete Event Simulation (DES) and System Dynamics (SD). DES and SD are the two main simulation methods that support automated systems used to analyze complex models. The DES method is appropriate for modeling issues of operational focus, reductionism in prospective, quantitative in nature, discrete in change, and narrow in details. The SD method is appropriate for problems that have strategic/context focus, holistic in prospective, qualitative in nature, continuous in behavior, and broad in details (Brailsford and Hilton 2001). Specifically, the DES method is used to model discrete variables while the SD method is used to model continuous variables. It is important to mention here that construction-related systems have a combined structure of strategic/context (continuous behavior) and operation (discrete behavior) level parameters (Lee et al. 2007). While the SD method works well for capturing the effects of context level parameters (staff skill and fatigue, change in project scope, motivation, etc.) in feedback loop systems, the DES method is powerful in modeling and analyzing operation level parameters in sequential systems (Alvanchi et al. 2009). The two methods can thus be seen as complementary.

Despite the DES method enjoying numerous successful applications in construction (Halpin 1973; Paulson 1987; Ioannou 1989; Martinez et al. 1994; Hajjar and AbouRisk 2002; Marzouk and Moselhi 2003), its simulation scope is limited to detailed analysis techniques and to the operation level of the project (Huang et al. 2003; Smith 2003; lee et al. 2002a; Baines and Harrison 1999). The DES method suffers from a major drawback, such as its limited capability to capture the strategic/context of the operation being modeled and it is incapable of modeling the cause-effect feedback loops that exist among variables (Lee et al. 2007). Additionally, it fails to determine the system stability in the surrounding environment, where performance may be driven by a hidden causal relationship that could be non-linear (Lyneis et al. 2001; Helal et al. 2007). The qualitative and continuous nature of the strategic level variables creates challenges in using the DES method to model this level (Zulch et al. 2002; Baines and Harrison 1999). Failing to model project strategic/context and operational variables concurrently will inevitably results in models that are not capable of representing real situations.

The SD method is an approach of problem solving initially developed by Jay Forrester at MIT in the early 1960s. In the concept of system dynamics, a system is defined as a collection of variables that continually interact over time to result in high-level system representation. In SD, dynamics refers to the change of model's behavior over time, and the SD model captures the variables affecting the behavior of the system through causal-effect loops. These loops depict the relationships among the variables in the system, as well as pertinent links between the system and its operating environment (Sweetser 1999). Looking at the system from a global view is valuable to decision makers as it aids in understanding complex inter-related construction operations. A competitive advantage of the SD method is its capability to trace causal relationships among system components and to trace any problematic behavior to its real roots in any part of the system.

Despite that the SD method is an excellent tool to represent the strategic/context level of the project, which provides new, promising ground for construction management modeling (Ford and Sterman 1998; Park and Peña Mora 2003; Lee et al. 2007; Alvanchi 2011), the SD method has only been used on a small scale, although it has been widely used in other fields. Finally, the SD method involves hundreds of mathematical equations developed to represent the relationships among variables. Solving such equations in the past was cumbersome; currently, with advancements in the computer industry, it has become feasible use complex models with complex feedbacks using the SD method.

1.3 Problem Statement

The SD and DES methods have many applications in modeling of construction operations and systems. The modeling and analysis are done at two levels, strategic and operational. The complexities inherent in projects are the results of interactions between strategic/context and operational aspects within the construction environment and boundary. The DES method has been widely used in modeling construction operations. However, the underlying model of the DES method often fails to represent real operations, as it assumes no relationship between project components. Meanwhile in reality, project components have a complex dynamic feedback process that requires modeling of inherent uncertainty in the execution of these projects. This dynamic nature is not explicitly addressed by the DES method (Cooper 1980; Sterman 1992; Cooper 1993 a, b, c; Cooper 1994). Project failure can be attributed to a poor representation of the inner and outer aspects that affect project dynamics. Uncontrollable external forces are often cited but the real cause may be internal, such as the feedback process among various components of the project.

Morris and Hough (1987) conducted an analytical study of 3500 projects to determine the reasons behind their failure. The results revealed that a lack of strategic analysis is the major cause behind the failure of many projects. In order to obtain a comprehensive and representative simulation model for large-scale construction projects, both strategic and operation levels, along with their feedback process, need to be simulated simultaneously. If the SD and DES methods are integrated to address the construction-modeling problem, the emerging tool can provide insight into the variables interactions and generates near real project behavior in the virtual world. In addition, after investigating the fundamentals of the DES and SD simulation methods, it can be concluded that the limitations associated with the DES method can be overcome using SD, and vice versa.

The need for a hybrid simulation of the DES and SD methods arise from the demands to address practical issues responsible for the success of the construction operations. For instance, (1) how do changes in staffing, overtime and scope affect different project operations? (2) what is the impact of overtime on quality? and (3) how do the errors generated in operations affect project completion date and quality?, as well as many other questions arising from the strategic/context level of the construction project. For example, because of the failure to meet a project schedule deadline, the manager takes corrective measures by assigning more workers or considering overtime to increase the chances of meeting the project schedule (El-Rayes and Moselhi 1998). The negative effects of an overtime policy may cause worker fatigue and burnout and hence decrease productivity. This reduction in productivity again delays the project completion time. Such interactions between the causes and effects are not possible to model using the DES method. The DES technique simply describes the project as top-to-bottom hierarchy through the decomposition of project elements to the smallest acceptable level, called tasks. Thereafter, costs, durations, and resources are estimated, mainly from experience, and loaded on tasks. Then the project's job logic is described as a network of tasks connected based on the work sequence and logic. The apparent purpose of this process is to describe the actual project behavior generated in reality. One of the main concerns associated with this static philosophy of addressing dynamic issues of planning and control lies in the ability of the restructured tasks of the network from bottom-to-top to behave based on the assumptions made at the project decomposition stage. On the other hand, SD is a modeling and simulation method with a wide range of applications in different fields, used mainly to model strategic aspects. One of the strengths of SD is modeling the whole system within a predetermined boundary. This allows an understanding of the system behavior. However, SD fails to account for the operational aspects at the tactical level.

AbouRizk and Hague (2009) recognized the need to have a robustness of comprehensive construction operation simulation by using a hybrid system of the DES and SD methods; they stated: "*Researchers have begun to look at hybrid modeling, which incorporates more traditional discrete-event approaches with system dynamics, as a valuable tool for comprehensive project planning. By considering both strategic and operational aspects, hybrid simulation can produce more complex models better attenuated to* *construction scenarios."* The coupling of the DES method with SD is expected to provide valuable complementary information. The DES method supplies detailed information while the SD method simulates the impact of management policies and strategies on project execution. Thus, this research identified an opportunity to benefit from the capabilities of both the DES and SD methods.

1.4 Research Objectives

The main research objective is to study how best to alleviate the limitations associated with the DES method by developing an integrated simulation environment that benefits from the unique capabilities of the DES and SD methods. The developed method should be able to capture the neglected interactions between construction operation's strategic and operational levels using the SD and DES methods, respectively. A hybrid simulation method that utilizes both SD and DES is to be developed. To achieve the stated main objective, the following five sub-objectives are carried out:

1. Study the heterogeneous aspects of construction projects with emphasis on variable types, decision-making levels, and feedback processes.
2. Investigate the adequacy of utilizing the existing hybrid simulation methods in construction projects.
3. Propose a hybrid simulation method that is capable of developing hybrid simulation models and applications.
4. Develop a synchronization method that integrates DES and SD simulation clocks.
5. Implement the developed method in a prototype hybrid simulation application.

1.5 Summary of Research Methodology

Figure 1.1 demonstrates the methodology followed to achieve the objective of this research. The methodology is summarized in five phases: analysis, development, implementation, validation, and conclusion. The analysis (phase I) focuses on performing an exhaustive literature review on the state of the art of modeling and simulating construction operations. It reviews past and current practices used to model and simulate management issues of social science in different fields. Issues of focus are project management decision levels, discrete event simulation, continuous simulation, system dynamics, hybrid simulation, and time synchronization algorithms. In addition, the available commercial software systems used to develop hybrid simulation models are studied and analyzed. From the analysis stage, gaps and limitations in the current simulation practice are identified.

To respond to the limitations and gaps identified in the analysis phase I, six major components are then identified. These are the core of the developed hybrid simulation method and are addressed in the development (phase II) as follows. First, the influential units that generate behavior and their boundaries are identified by decomposing the project into units. This represents the model scope. Second, these units are tested against criteria developed from the philosophy of the DES and SD methods to select the most appropriate simulation method for simulating these units. Third, a simulation model, either it DES or SD, is developed to represents the units. In this step, the hybrid simulation model structures and interface points between simulation models are selected. Fourth, a formalism that is capable of describing the elements of simulation model is developed (e.g., variable, interface variable, synchronization time). Fifth, the synchronization method responsible for integrating the simulation clocks of the DES and SD models is developed. The method allows both simulation methods to update states based on their norms. Sixth, the Executer that integrates the simulation components on a single computation platform is coded.

Phase III involves implementing the developed hybrid simulation method using a real case study from the construction industry. The implementation phase involves simulating a real-world case study (earthmoving project). The DES and SD models pertaining to the case are developed using EZStrobe and Venism software systems, respectively. By using the six developed hybrid method components, a Hybrid Simulation (HiSim) is created using the VB.NET programming language. The developed application integrates EZStrobe and Venism on a single simulation computational platform.

Phase IV involves testing and validating the hybrid simulation method, system dynamic model, and HiSim application. Finally, Phase V involves presenting the conclusions of the research. The conclusion remarks on the hybrid simulation method are stated. A detailed discussion of the challenges found and lessons learned are presented. In addition, the conclusion provides a summary of the findings, and highlights potential topics for future research. Phase V ends by discussing the limitations associated with the developed method and possible future enhancements.

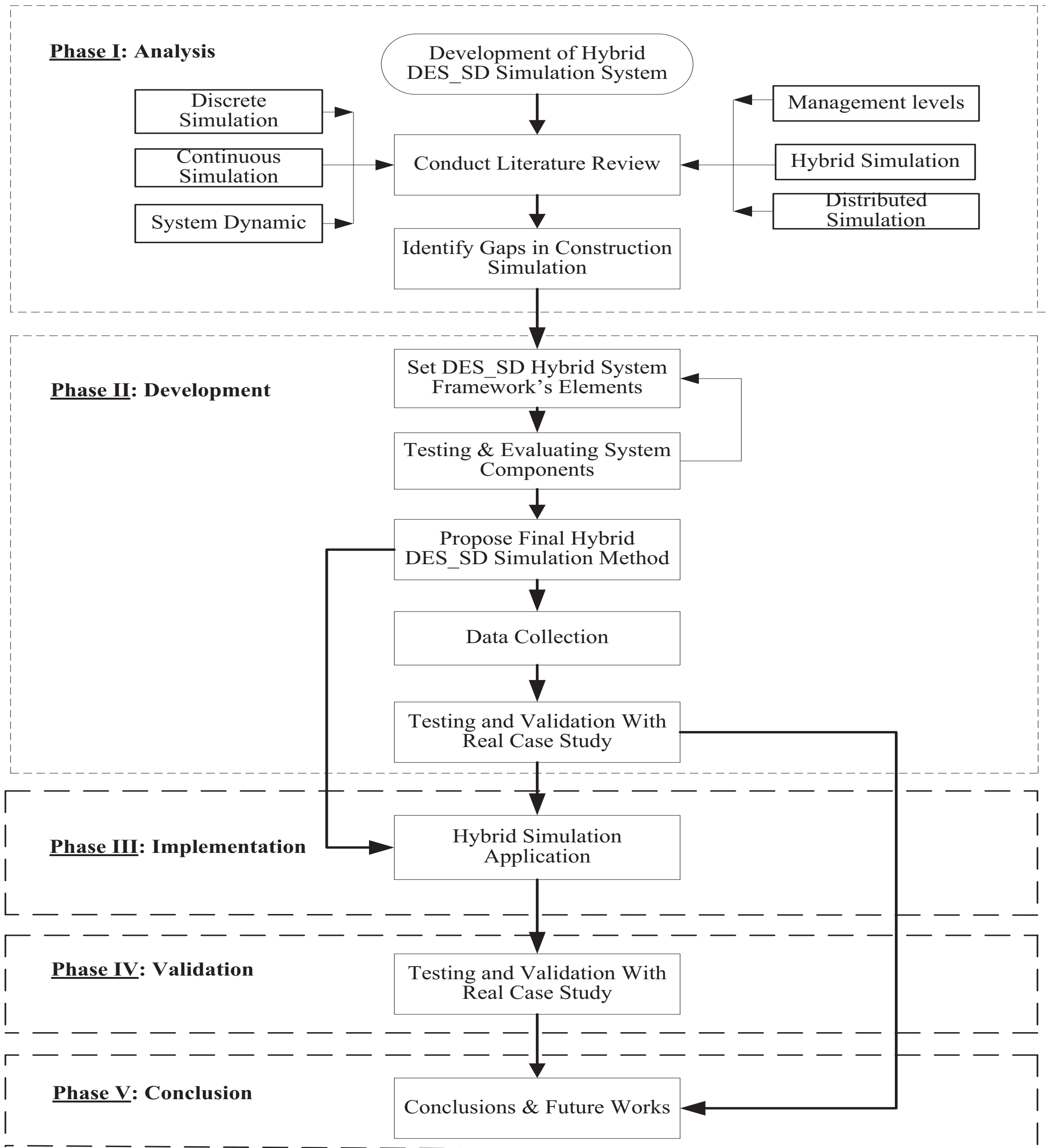


Figure 1.1 Overview of the Research Methodology

1.6 Thesis Organization

The thesis consists of eight chapters, and three appendices. Chapter 1 provides a discussion of the uses and limitations of the DES and SD methods in the construction domain and the needs for a hybrid simulation. It highlights the requirements for overcoming the limitations associated with each method. The need to have a hybrid simulation system that provides a foundation for hybrid simulation modeling practice is illustrated. This chapter also provides a description of the main and sub-objectives of the research. Finally, it presents a summary of the methodology adopted to achieve the thesis's objective.

Chapter 2 presents a literature review of the simulation theories and tools. It investigates the simulation methods used to model and simulate construction operations, such as DES, SD, and a hybrid of both simulation methods. In order to understand the evolution of hybrid simulation in other fields, a review of the existing hybrid models and methods is conducted. It reviews the current formalism and synchronization mechanisms used to develop hybrid models. Finally, the chapter presents a summary of the limitations and gaps in the existing methods.

Chapter 3 provides an outline of the developed research methodology, based on the discussion presented in Chapter 2. This chapter presents the roadmap for hybrid simulation used to achieve the research objectives. It involves five phases: 1) analysis; 2) development; 3) implementation; 4) validation; and 5) conclusion. The development phase is further divided into six core components necessary to a functional hybrid simulation method.

Chapter 4 addresses the development phase of the hybrid simulation method presented in Chapter 3. This chapter systematically develops the six components necessary for integrating the DES and SD methods.

Chapter 5 focuses on the data collection stage, which is necessary to implement the hybrid simulation method. It presents a case study that involves earthmoving operations in a dam construction. This case study involves excavation and backfilling operations. The data collection is also extended to published articles and notes.

Chapter 6 presents, in its first part, the full implementation of the developed hybrid simulation method using a semi-automated approach. It illustrates the stages of the DES and SD models development for the earthmoving operations case. The second part of the chapter focuses on synchronizing the simulation models using a new developed method for this purpose. Finally, the chapter presents a validation and testing of the hybrid simulation model, and discusses the model's outcomes.

Chapter 7 presents the fully automated computer application used to develop hybrid simulation models. The tool integrates DES and SD models on a single computation platform. This chapter demonstrates the required steps to use the developed application, in addition to validation and testing procedures.

Finally, Chapter 8 summarizes the thesis, highlights its contributions and limitations, and proposes future research work.

1.7 Summary

This chapter introduced the research problem of this thesis with focus on the limitations of the widely used simulation methods such as DES and SD in capturing all aspects of construction operations. The issues of dynamics and management decision level (strategic and operation) need to be considered when developing realistic simulation models. The drawbacks associated with the DES and SD simulation methods were highlighted in the problem statement. The main objective and sub-objectives of the research emerged from the problem statement and summarized the research scope. The development of a hybrid simulation method that is capable of providing a comprehensive simulation tool is indispensable to counter the current increase in complexity and dynamics of construction projects. Outlines of the research methodology and of the thesis were presented.

CHAPTER 2

LITERATURE REVIEW

2.1 General

Different professionals use models of one form or another, but the term "model" does not have the same meaning to engineers, managers, and business professionals. It is important before beginning the literature review in this chapter to state an unambiguous definition of three terms that are used extensively throughout this research; these terms are *module*, *model,* and *system*. In this research, *module* is defined as a part or a unit of the whole model, while *model* is defined as a representation of a real-world situation and provides a framework within which a given situation can be investigated and analyzed (Zayed and Halpin 2001). A model can be viewed as an abstract and simplified representation of a system at one point in time where it tries to capture the system's reality. A model can be made by combining several modules. *System* is defined as "a group or collection of interrelated elements that cooperate to accomplish some stated objective" (Shannon 1998). A system can be viewed as a set of related components or entities (internal components of the system) which interact with each other based on the regulations and policies (external inputs) of the system.

Simulation has been used in many fields such as industrial, business, manufacturing, environment, and construction (Banks et al. 2000). Shannon (1998) defined simulations as "the process of designing a model of a real system and conducting experiments on this model for the purpose of understanding the behavior of the system and/or evaluating various strategies for the operation of the system." Figure 2.1 depicts the modeling and simulation process of a real problem. Experiment with real situations is expensive in terms of resources. Instead, a model is constructed to mimic the real system's behavior. The model must be verified and tested before considering it as reliable. Different policies and scenarios are tested to find the optimum solutions. Eventually, the model generates a list of solutions that are studied and analyzed to select the optimum one.

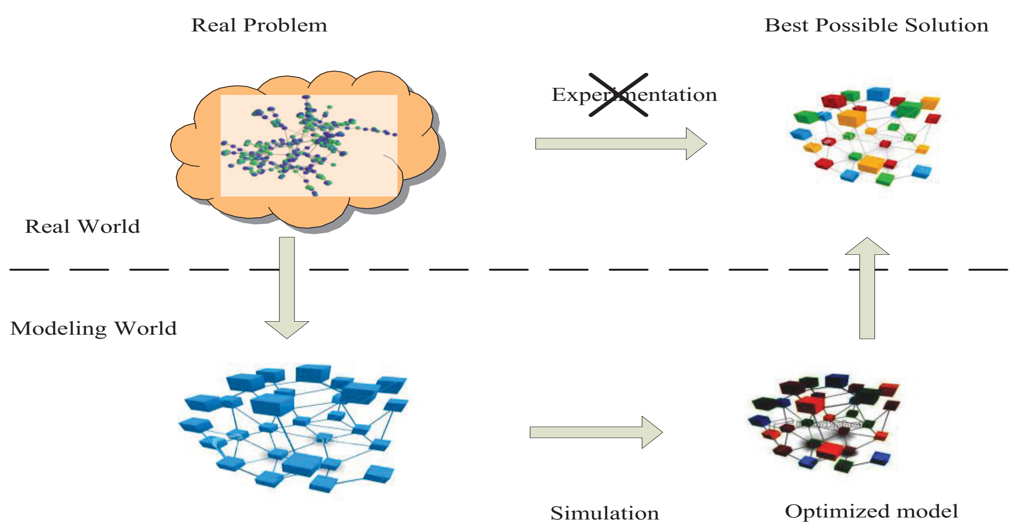


Figure 2.1 Modeling and Simulation of Real Problem (Borshchev and Filippov 2004)

A system commonly consists of two types of variables: discrete and continuous. Consequently, simulation models can be classified from the perspective of variables as discrete-event simulation models, continuous simulation models, or combined simulation models. The behavior of these models is shown in Figure 2.2. A discrete-event simulation model portrays the physical operation of a system as a chronological sequence of events. Each event occurs at an instant in time and marks a change of state in the system (Halpin and Riggs 1992). A typical behavior of a discrete variable is shown in Figure 2.2a. In continuous simulation, a system behavior is continuously monitored over time. This is performed according to a set of equations typically involving differential equations. The behavior of continuous simulation models is demonstrated in Figure 2.2b. In combined simulation systems, the change in the system variables occurs discretely, continuously or continuously with discrete jumps over the simulation time, as shown in Figure 2.2c (Pritsker et al. 1997). It should be noted that change in the dependent variables in construction operations occurs discretely, continuously or both. Using any of the existing techniques depends on the operation being modeled and the preference of the modeler. In this research, a combined DES and SD method is utilized to achieve the research objective.

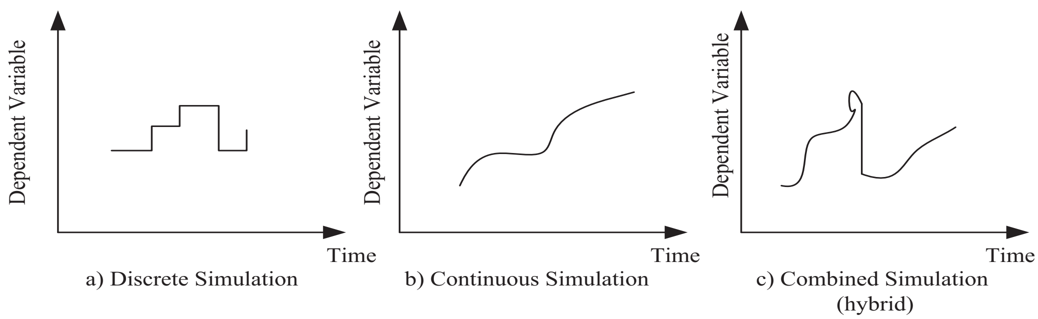


Figure 2.2 Simulation Modeling Techniques (Pritsker et al. 1997)

2.2 Strategic Project Management and Operational Project Management

Management science classifies management levels as Strategic Project Management (SPM) and Operational Project Management (OPM), as shown in Table 2.1, and the decisions taken in the project lifecycle are classified as strategic and operational. Lyneis et al. (2001) defined project strategic level decisions in the context of strategic management as "strategic project management covers decisions that are taken up front in designing the project, and then the guidance provided to operational decisions that considers the longer-term impact of these decisions on downstream performance of the project." The SPM definition in this framework is differentiated from the definition pertaining to organizational aspects. SPM means achieving the project's set objectives within the organization's strategic level. This involves an adjustment of certain parameters like cost, resources, and time to meet set goals (Rodrigues and Bowers 1996).

On the other hand, OPM can be viewed as the actions taken to meet project goals set by SPM. It focuses on the daily operational details of a project at the micro level. OPM is discrete in nature and one of its major disadvantages is its inability to function without communicating with the project's strategic targets. The OPM level is modeled using DES method; for example, an earthmoving operation, which involves loading, hauling, and dumping, is modeled using DES since these processes are at the tactical level. This approach of simulation results in misleading outcomes, since construction project behavior is highly influenced by policies adopted to drive the project execution and by the surrounding environment. OPM is unpredictable and particularly depends on SPM. Thus, a simultaneous simulation of OPM and SPM is required to develop successful simulation models (Schultz et al. 1987; Lee et al. 2006).

Table 2.1Comparison between SPM and OPM (adapted from Schultz et al. 1987)

Viewpoint	Strategic Project Management (SPM)	Operational Project Management (OPM)
Level	Macro	Micro
Assessment	Subjective	Objective
Nature of the problem	Unsaturated, one at a time	More saturated and repetitive
Information Needed	Small amount of specific information	Large amount of information
Planning Horizon	Long-term, but varies with problem	Short-term and more constant
Frame	Covers entire scope of project	Concern with only sub-project units
Level of detail	Broad and general	Narrow and problem specific
Evaluation	Difficult, because of generality	Easier, because of specificity
Perspective	Holistic and continuous	Reductionism and discrete
Focus	Strategic/Context	Operational

2.3 Discrete Event Simulation (DES)

DES method is a powerful tool in analyzing and simplifying a complex system, and has occupied the mainstream of construction simulation research, focusing particularly on construction operations (Walsh et al. 2002; lee et al. 2005). The technique deals with a list of events (instantaneous occurrence changing the state of a system); this list is filled once every future event is scheduled and is depleted by firing elapsed events. One important rationale in DES modeling is that a construction project is envisaged as a collection of its constituting processes (AbouRizk et al. 1992). DES models a system as a network of queues and processes, where state changes occur at discrete points of time (Brailsford and Hilton 2001). In DES model, entity flows through the system and seizes resources to perform a work task, and when the task is accomplished, the entity releases the resources. If resources are busy or unavailable, entity waits in queue until the resources become available. These actions are called events. Another important concept in the DES is the simulation clock (variable representing simulated time), which schedules all events on the list intended to occur during the simulation. When an event occurs, the simulation engine is triggered to advance the simulation clock to the next scheduled event on the list. Pritsker et al. (1997) defined three methods to model discrete systems: 1) describing the changes in the state of the system at each event time; 2) describing the task in which the entities engage; or 3) describing the process of entities flow.

The literature has demonstrated many successful applications of DES in the construction modeling area (Halpin 1973; Paulson 1987; Ioannou 1989; Martinez et al. 1994; Hajjar and AbouRizk 2002; Marzouk and Moselhi 2003; Elwakil and Zayed 2011). In construction, DES is primarily applied to model the operational level, such as earthmoving, concrete placement, tunneling, road construction, and underground pipe-jacking (AbouRizk et al. 1992; Sawhney et al. 1998; Walsh et al. 2002; Marzouk and Moselhi 2003). However, DES has several major drawbacks, such as its limited capability to capture the strategic/context of the construction operation being modeled and its inability to address the cause-effect feedback loops that exists between project variables (Lee et al. 2007). Additionally, DES fails to determine the system's stability in the surrounding environment, where performance may be driven by a hidden causal relationship that could be non-linear (Lyneis et al. 2001; Helal 2008). The qualitative and continuous nature of the strategic level variables creates challenges when using DES to model this level (Zulch et al. 2002; Baines and Harrison 1999). In addition, the DES method requires a large, detailed amount of data which may not be available at some stages of the model building. Furthermore, DES may face difficulties in containing the project's strategy due to the inherent time step advancement mechanism (Martin and Raffo 2001).

Lastly, DES fails to provide the modeler or manager a clear picture of the system state between two consecutive events (e.g., E_1 to E_2). To demonstrate, the simulation behavior of a DES model is demonstrated in Figure 2.3. The figure shows three events (E_1, E_2, and E_3) that occur at three times (T_1, T_2, and T_3). At the start of the simulation run, time is recorded as (T_0) and the system's state is recorded as (S_0). Then, the simulation clock is advanced to T_1 due to the occurrence of event E_1, and the system's state is recorded as S_1. Again, the simulation clock is advanced to T_2 due to the occurrence of event E_2 and the system's state at this point is recorded as S2. The figure also illustrates a continuous line, marked A, between event E_1 at T_1 and event E_2 at T_2. This line represents the system's state at the occurrence of E_1 and the system's state just before the occurrence of E_2. The system's states between times (T_0 and T_1) and (T_1 and T_2) are not updated from the earlier recorded state at Tn_{-1}. The system's state between the occurrences of subsequent events is not updated until the occurrence of the next scheduled event. This behavior of DES models will limit the full understating of the interactions between the operation's parameters as the states between events remains unknown. In addition, this can cause delays in measures taken to correct any deviation in the operation. Curve B demonstrates the expected actual system state updates after considering the states between events.

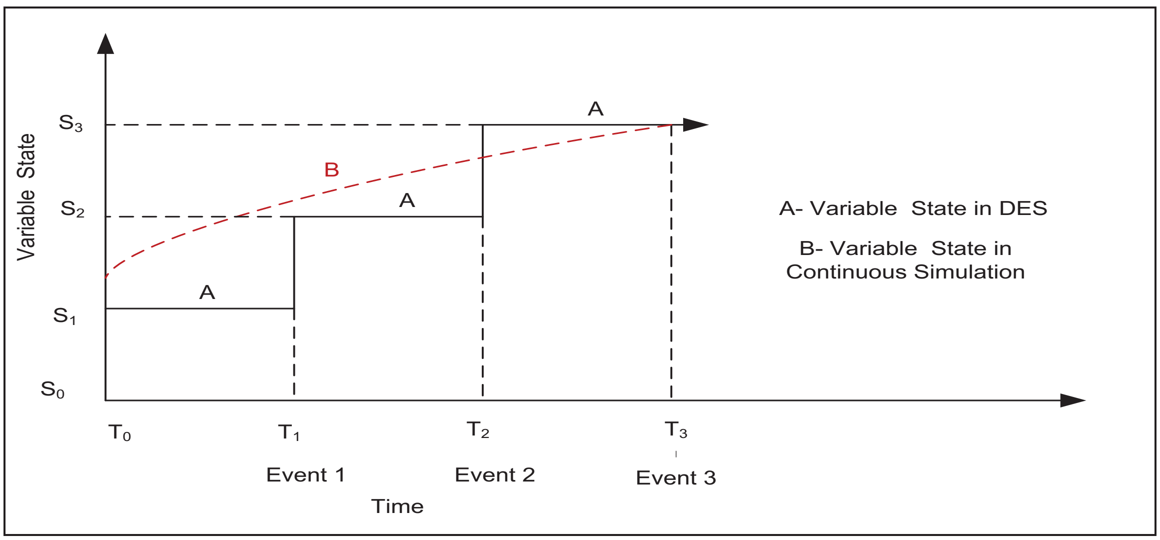


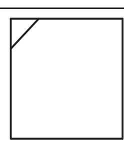
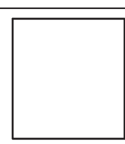
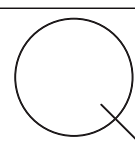
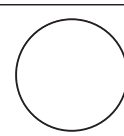
Figure 2.3 System State Update in DES and Continuous Simulation

2.3.1 Applications of DES in Construction

Remarkable efforts have been made by researchers in the construction field to model and simulate construction operations, such as Halpin (1977), Ioannou (1989), Shewchuck and Chang (1991), Oloufa (1993), Huang and Halpin (1994), Tommelein et al (1994), Sawheny and AbouRizk (1995), Shi and AbouRizk(1997), Martinez and Ioannou (1994, 1999), Martinez (1996), Oloufa et al (1998), Hajjar and AbouRizk (1999), Chua and Li (2002), Marzouk and Moselhi (2003) and Elwakil and Zayed (2011).

Halpin pioneered the research in construction simulation by introducing a simulation language for construction operations called CYCLic Operation NEtwork (CYCLONE) (1976). CYCLONE is a general-purpose modeling system that provides a systematic way of planning, organizing, analyzing, and controlling construction operations. It consists of a network simulation language that best suits operations of cyclic nature, such as highway construction, concrete poring, earthmoving, and piping. CYCLONE consists of six essential elements used to model construction operations, as shown in Table 2.2. The elements are: 1) NORMAL represents an unconstrained work task at its start and indicates active processing resources; 2) COMBI represents a work task constrained by the availability of resources and always preceded by a minimum two QUEUE; 3) QUEUE classifies the resources by either active or waiting for use by entity (idle state); 4) FUNCTION facilitates unit consolidation for units associated with a selected QUEUE (CONS) or generates units that are associated with a selected QUEUE (GEN); 5) COUNTER is the simulation clock that controls the iteration of the simulation run; and 6) ARROW represents the entity's directional flow. With the rapid advancement in computer technology, many general and special purpose simulation applications have evolved from CYCLONE. An example of the simulation languages that have been developed are INSIGHT (Paulson 1978), RESQUE (Chang 1987), UM-CYCLONE (Ioannaou 1989), Micro-CYCLONE (Halpin and Riggs 1992) and CIPROS (Tommelein et al. 1994).

Table 2.2 CYCLONE Modeling Elements (Halpin 1976)

Element Name	COMBI	NORMAL	QUEUE	FUNCTION	COUNTER	ARROW
Modeling Element Symbol						

The object-oriented concept was first introduced in 1991 in the manufacturing system simulation and modeling (Shewchuk and Chang 1991). This new method of programming substantially reduced the necessary coding effort and improved simulation modeling capability and quality. Researchers in the field of construction modeling and simulation have benefited from the object-oriented programming method. It was widely adopted, and as a result, many systems and applications had been developed using object-oriented programming. MODSIM (Oloufa 1993) was one of the first systems developed using object-oriented programming. It attempted to simplify the understanding and building process of simulation models by adopting a graphical application interface rather than writing programming code. The approach works by creating objects of different classes that represent construction operation resources and entities, while message transfer is used to communicate the objects. An enhancement along this same track to divert simulation programming in construction from extensive code writing to a graphical interface application was attained in DISCO (Dynamic Interface Simulation for Construction Operation) (Huang and Halpin 1994). DISCO employs a schematic modeling format demonstrating the dynamics of the construction operation. It deploys the abstract model diagram of the construction operation as a static display and then dynamically demonstrates the associated information on a computer screen, such as the simulation clock, idleness of resources, number of resources, etc. Finally, it reports node statistics information graphically and in tabular form at the end of simulation run.

Other researches such as Tommelein et al. (1994); Shi and AbouRizk (1997); Oloufa et al. (1998) and Chua and Li (2002) attempted to build simulation modeling processes based on matching the project’s resource properties (material and equipment) with design element properties and operation durations. The CIRPOS (Tommelein et al. 1994) system uses a modular to create a DES network with stochastic activity duration and to relate the simulation output to construction designs and plans. Resource-Based Modeling (RBM) (Shi and AbouRizk 1997) is another system developed for construction simulation, in which the operating processes of active resources are defined as atomic models. The atomic models are stored in the model library and can be modified to form project-specific atomic models according to the user's project information. The end user can simply construct a simulation model for any operation by using the atomic model as a base, then specifying the required resources from the resources library. Then, the model is generated by formatting different processes from CYCLONE to SLAM II. Special Purpose resource-based simulation libraries were developed to serve specific purposes (Oloufa et al. 1998). These applications are preprogrammed libraries of construction resources. The user needs only to select the required resources and determines the project logic by linking these resources together. The library is developed by a simulation programmer that addresses a specific project type such as earthmoving, tunneling, and others.

Resource-Interacted simulation modeling (RISim) was developed to simplify and speed up the model development cycle (Chua and Li 2002). It adopts a resource-oriented methodology. The operation is modeled in two abstraction levels: 1) the resource level, where operation logic is mainly represented with internal complex resource flows; and 2) the process level, where the processes involved are generated from the resources involved. For instance, the work of a crane (resource) is the repetition of two activities, lift up (process 1) and lift down (process 2), in addition to its idle status.

The main objective of researchers in construction simulation was to minimize the use of a deterministic approach in construction project planning, in addition to other advantages that simulations can offer. The deterministic, static nature of traditional planning methods such as CPM is a major obstacle to provide realistic project duration. Hierarchical Simulation Modeling (HSM) is used to overcome this limitation (Sawheny and AboRizk 1995). The approach combines two basic things, the work breakdown structure (WBS) and process modeling. HSM is centered on the development of the project plan using a symbol-based graphical format.

Computer programming languages have continued to advance. Consequently, construction simulation has benefited from this progress and has continued to evolve. Sophisticated simulation systems have been developed using C-programming language, such as STROBOSCOPE (Martinez and Ioannou 1994, 1999 and Martinez 1996). STROBOSCOPE is a programmable and extensible simulation system designed for modeling complex construction operations and for developing special-purpose simulation applications. The developed system has the ability to make complex dynamic decisions based on the simulation system state, characteristics, attributes, and resources. It is based on three-phase activity scanning. STROBOSCOPE simulation models can be in a code format or a graphical network-based developed using EZstrobe. They can access the state of simulation such as simulation time and the number of entities in queues. STROBOSCOPE is considered the most reliable and advanced general-purpose simulation system for modeling and simulating construction operations such as earthmoving, highways, piping etc.

Construction projects have become more complex and challenging than ever due to their increased size and expectations. Demands to increase certainty in project success have also increased, and as a result, enhanced specific applications that serve particular operations have become a necessity. For instance, earthmoving operations are more complex and dynamic. They require a built-in special simulation application called Special Purpose Systems (SPS) that addresses all aspects of the specific types of operations (Hajjar and AbouRizk 1999; Marzouk and Moselhi 2003). A computer simulation system called Simphony (Hajjar and AbouRizk 1999) was built. It provides a standard, consistent, and intelligent environment for developing simulation models as well as the utilization of construction SPS tools. Simphony is a result of the accumulation of three SPSs developed prior to its development, which are: 1) AP2-Earth (Hajjar and AbouRizk 1996), which allows for the analysis of large earthmoving projects; 2) CRUISER (Hajjar and AbouRizk 1998); built for modeling aggregate production plants; and 3) CSD (Hajjar et al. 1998) which allows the optimization of dewatering operations in construction sites. SPS templates can be created in the Simphony environment. This can be done by either using the Simphony Designer (graphics) or Simphony Editor, which allows users to create and execute simulation models based on the elements available in the modeling element library. SimEarth (Marzouk and Moselhi 2003) is another SPS developed to optimize and simulate earthmoving operations through using a generic algorithm. SimEarh provides a tool to select a near-optimum fleet configuration that minimizes the total project cost and duration. The methodology uses the DES and object-oriented modeling. The three-phase simulation approach that was used by Martinez (1996) rather than process interaction was employed to control the dynamics of the simulation process and to track the activities. The optimization process uses genetic algorithm to search for the near-optimum fleet configuration. Both qualitative and quantitative variables that influence the earthmoving operations are included in the modeling process. KEYSTONE (Knowledge Discovery Based Simulation System) is the most recent simulation language developed (Elwakil 2011). The system emerged from the needs to account for fuzziness, missing data and outliers in input data, in addition to account for the subjectivity ignored in most of the previously developed systems. The developed system elements are: 1) the Knowledge Discovery Stage to model qualitative variables using the Fuzzy Clustering technique, 2) the simulation stage to model unit movement in the model, and 3) the optimization stage to select the optimum solution using the Pareto ranking technique.

2.4 Continuous Simulation

Continuous simulation is defined as "modeling system behavior over time where variables states change continuously with respect to time" (Law and Kelton 2000). Continuous simulations are based on a set of differential equations. These equations define the peculiarity of the state variables, the environment factors so to speak, of a system. These parameters of a system change in a continuous way and thus change the state of the entire system. The set of differential equations can be formulated in a conceptual model representing the system on an abstract level. Models developed by using continuous simulation are generally deterministic. Continuous simulation is considered simple and requires less data to finalize a small model. As the continuous model becomes more complex, the differential equations tend to become cumbersome to solve, especially with higher orders of differential equations.

2.5 System Dynamics (SD) Simulation

SD was introduced by Forrester in 1961 as a modeling and analysis approach for solving complex social systems in the industrial sector (Forrester 1965). It is based on the concept that a system's behavior over time is determined by its structure. SD has been successfully applied to systems requiring a holistic consideration as well as feedback loops among system parameters. SD modeling is applied to social problems, economic, engineering, environmental systems, and management (Wolstenholme 1990; Abdel-Hamid and Madnick 1991). The SD model strength is in the feedback loops. These loops are the main source of dynamic behavior observed in a system (Sterman 2000). SD is an elaboration on continuous simulation with a focus on system complexity and the nonlinearity of feedback processes. It is an approach to solve problems at top management levels (Forrester 1975; Sterman 2000; Lyneis 2001). Two common forms of notations exist in SD, Causal Loop Diagrams (CLDs) that capture the conceptual relationships in the system and Stocks-Flows Diagrams that describe the movement of entities from start to end in a model.

Rodrigues and Bowers (1996) and Sterman (2000) summarized the motivation to apply the SD modeling method in project management as follows.

1. The need to consider the whole project rather than a sum of individual elements.
2. The need to examine non-linear scenarios described by balancing and reinforcing feedback loops.
3. System of highly dynamic
4. Involving both “soft” and “hard” data
5. The need for experimenting with the project behavior by applying different hypothetical scenarios, and
6. The failure of the traditional analytical tools (DES) to solve parts of the project management problems.

2.5.1 SD Modeling Process

Building an SD model begins by defining the conceptual model that represents how the modeler perceives the system behavior. Next comes defining the boundary within which the system behavior is generated. The relationships among the system components are represented in a form of feedback processes. Then, a set of mathematical equations to describe the interactions of the model components are developed. Finally, components are input into computer simulation software for computation. In the following sections, the modeling and producers involved in building SD models are discussed in detail.

2.5.2 Steps to Building SD Model

The SD method presents systematic procedures for modeling the system. Sterman (2000) identified five steps for modeling a system using the SD method: 1) system understanding; 2) conceptualization; 3) formulation; 4) validation; and 5) policy design and analysis. These steps are explained in detail as follows.

1. **System understanding:** This step is crucial to building a successful model as the SD approach is based on how the whole system is comprehended and represented. Sometimes, the model goes beyond human mental capabilities; therefore, breaking the problem without violating the holistic concept of SD is necessary. The modeler should understand the problem in depth, identify the key variables, identify the time horizon, and understand the historical behavior of the variables.
2. **Conceptualization:** What are the current theories related to the problematic behavior? This step involves the development of maps of the causal loops' structure and feedback based on the initial hypotheses, key variables, and reference modes.

3. **Formulation of the simulation model:** Specification of the structure, estimation of parameters and behavior relationships.
4. **Testing the model:** A comparison to reference models and checking the model robustness under extreme conditions.
5. **Policy design and evaluation:** Checking the environment conditions that may arise, new decision rules and strategies.

(i) Model Boundary

The feedback process in the SD model has a closed boundary within which the behavior of the system is generated. Defining the boundary involves selecting the components of interactions necessary to generate the behavior of interest as specified by the model's purpose. The model boundary summarizes the scope of the model by listing which key variables are *endogenous*, *exogenous*, and *excluded*. *Endogenous* variables are the main concern of all model variables. They are variables in a causal-effect structure whose value is determined by the states of other variables in the system. These variables usually portray the dynamics inherent in systems. Examples of endogenous variables in construction are fatigue, overtime required, error, and quality.

Exogenous variables come from outside of the model and are unexplained by the model's feedback structure. They are involved in a causal-effect structure whose value is independent from the states of other variables in the system; a variable whose value is determined by variables outside the causal system under study. The system's internal interactions have no influence on such variables. Examples of exogenous variable are planned project duration and planned productivity. Finally, variables categorized as excluded variables are cautiously not included in the structure's causal-effect feedbacks. They are considered as beyond the scope of the model. The exclusion of these variables should not have a great influence on the model representation; otherwise, they have to be added to the model. Excluding some variables is a necessity, as including unnecessary variables in the modeling complicates the model, and makes the model more difficult to comprehend and develop.

(ii) Causal Loops Diagrams (CLD)

Complexity and uncertainty in construction are usually driven by feedback loops (Lee 2006). CLDs represent the conceptual feedback structure of the system as understood by the modeler (Richardson and Pugh 1986; Sterman 2000). According to Sterman (2002), CLD systems can be classified as open or closed systems. Open CLD systems have outputs that respond to, but have no influence upon, their inputs. On the other hand, closed CLD systems have outputs that respond to the inputs and are influenced by these inputs. CLD can be either positive or negative. Positive loops are a series of causal relationships that signify a self-reinforcing process and create results that are amplified. Generally, positive loops cause a destabilization of the operation, but may occasionally work to stabilize the operation. Negative loops are series of casual relationships that tend to direct the operation toward a specified goal value. For instance, two variables 'A' and 'B' in CLD can be considered, which have a cause and effect relationship, as shown in Table 2.3. When variable 'A' increases, variable 'B' is affected and either increases or decreases in its magnitude. When variable 'A' increases and variable 'B' increases then it is said 'A' has a positive effect on 'B' and a positive sign is put at the arrow's end. When increases in variable 'A' cause variable 'B' to decrease, then variable 'A' is said to have a negative effect on variable 'B' and a negative sign is put at the end of the arrow connecting the two variables. The change in the variable can be mathematically computed by integrating variables' rate of change.

CLDs are useful in representing interdependencies and the feedback process (Sterman 2000), but they are unable to capture the stocks and flows diagram (SFDs) of the system (Richardson 1986; Richardson 1997; Sterman 2000; Binder et al. 2004). The SFDs are generally generated from the CLDs (Sterman 2000).

Table 2.3 Denotations for Causal Loop Diagramming (Sterman 2000)

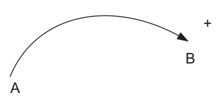
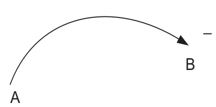
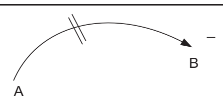
Casual link	Description	Mathematics Formulation
	All else remaining equal, if variable 'A' increases (decreases) then 'B' increases (decrease) in variable above (below)	$\partial B/\partial A > 0$ in this case of accumulations, B $=\int_{t0}^{t}(A+\cdots)ds + B_{t0}$ (2.1)
	All else remaining equal, if variable 'A' increases (decreases) then 'B' decreases (increases) in variable below (above)	$\partial B/\partial A \quad < 0$ in this case of accumulations, B $=\int_{t0}^{t}(-A+\cdots)ds + B_{t0}$ (2.2)
	Significant time delay is involved in implementing the casual relationship between the variable 'A' and 'B'	

Figure 2.4 demonstrates a CLD of a 'work to do' in a typical construction operation. For instance, the figure consists of three loops 'A' (reinforcing +), 'B' (balancing -) and 'C' (balancing -). Loop 'A' consists of 'work to do', 'overtime hours required', 'fatigue', and 'error' variables. As the 'work to do' increases, the demand on 'overtime' to meet the project deadline increases (+). This in turn causes 'fatigue' to the workers (+), and the fatigue increases the 'errors' in the completed work (+). Finally, due to the increase of 'errors', the initial defined scope of 'work to do' increases by an amount of work that needs to be reworked due to errors. The positive polarity of the loop 'A' is calculated by multiplying all the signs of the variables (+, +, +, +). Loop 'B', negative in polarity, consists of 'work to do',

‘overtime’, ‘work done’ and again ‘work to do’. When the ‘work to do’ increases, the ‘overtime’ is positively impacted, then ‘work done’ or accomplished is increased. When the ‘work done’ is increased, the initial work to do (the scope) decreases, and so on for loop ‘C’.

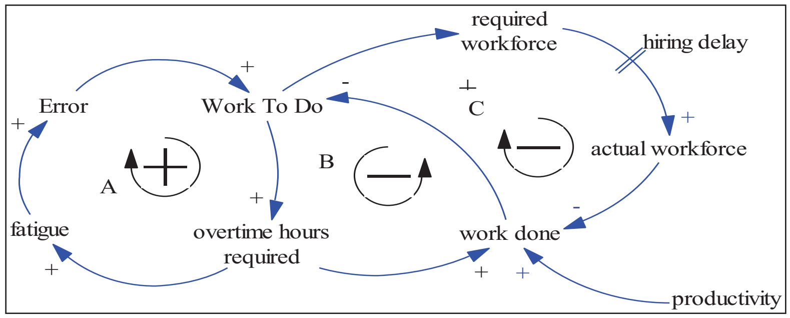


Figure 2.4 Causal Loop Diagram in a Typical Construction Project Workflow

The exogenous productivity variable in CLD (Figure 2.4) represents the planned work productivity, which is different from the actual productivity. The last important concept in SD terms is called *delay*. In Figure 2.4, the delay is represented by parallel lines (hiring delay). *Delay* is defined as a process whose output lags behind its input (Sterman 2000). Delays are a critical source of dynamics in nearly all systems, where some delays cause danger by creating instability. For example, when management makes a decision to increase the workforce to recover from schedule slippage, the decision has to pass through a hiring process and the new workforce needs to be trained. Therefore, it takes time for decisions made to influence productivity. Time consumed from the moment of making the decision to the moment where the result of the decision is noticed is called *delays*. Loops ‘A’ and ‘B’ have a different polarity, which means counter influence. Loop ‘A’ works toward increasing the ‘work to do’ while loop ‘B’ works toward decreasing the ‘work to do’. Thus, it is the responsibility of the management to foster the conditions surrounding loop ‘B’ and decrease the consequences of loop ‘A’.

(iii) Stocks and Flows Diagram (SFD)

Dynamic behavior in SD is raised due to the principle of *stock* or level. As the name implies, stock represents a variable state resulting from decisions. Stock is accumulation, characterizing the system state, and it generates information upon which decisions and actions are based and accumulated. Stock changes only through flows, and creates delays in the model by accumulating the difference between inflow to and outflow from the stock. Stock: 1) has a memory; 2) changes the time shape of flow; 3) decouples flow; and 4) creates delays. Finally, stock is modeled by the mathematical integration of the sum of the flows coming in to the stock and the flows dispatched from the stock.

On the other hand, *flow* represents actions or variables that influence the stock level or accumulation. Decoupling the rate (flow) from the system, stock becomes the source of disequilibrium in system dynamics (Sterman 2000). Stock and flow are explicitly included in the CLDs to enhance the clarity of the model schema.

SFD is composed of three elements, rectangles, valves, and clouds, as shown in Figure 2.5. Stock is represented by a rectangle while flow is represented by a pipe with a valve pointing to the stock. The cloud represents the source of inflow or source of outflow from the model. The cloud signals the model boundary in which the input to the model before the cloud and the output from the model after the cloud is considered outside of the model's boundary. Generally, stocks and flows diagrams are mapped from CLD.

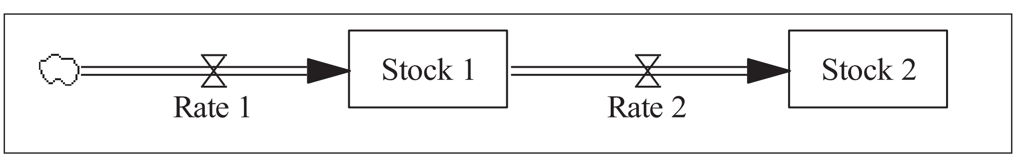


Figure 2.5 Stocks and Flows Diagramming (SFD)

(iv) Mathematical Representations of Stocks, Flows and Auxiliaries

Forrester (1961) originated SFD convention based on a hydraulic metaphor. The quantity of water that flows into the reservoir at any time is the accumulation of the water flowing in through the tap (flows) subtracted from the water flowing out of the reservoir (stocks). Stocks accumulate or integrate their flows; the net flow into the stock is the rate of change in the stock. Forrester (1965) stated that CLD, when mapped into mathematical equations, should be capable of describing the system being modeled and should handle continuous interaction such that any discontinuities resulting from solution time interval do not affect the results. When the mathematical equations of the SD model are finalized, the solution of the SD model starts by initializing the stocks (initial state of the system). During model run, the management monitors the system performance over time and intervenes when needed. The updated stocks and flows allow management to change the initially adopted policies and take corrective measures based on the model performance. In SD, the simulation time-length is broken into small equal time intervals (ΔT) called STEP TIME as shown in Figure 2.6. For instance, at the start of the simulation run, time is T1 and the system state is S1. When the system advances to time T2, the system state updated is S2. At time T2, the system state is the summation of the system state at T1 and the flow during (T2 –T1). When the simulation time reaches T3, then the system state S3 is the result of the summation of the system state at S2 and the flow rate during (T3 - T2).

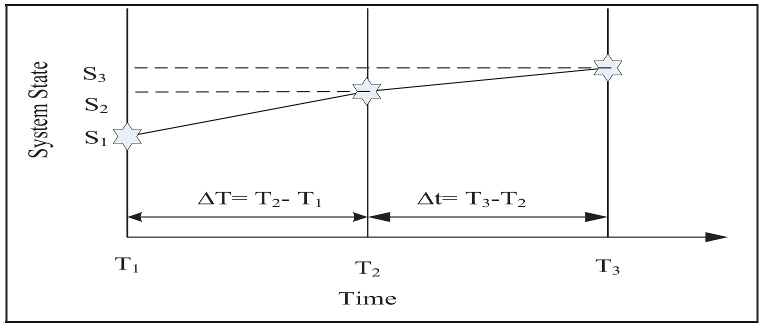


Figure 2.6 System State Computations in SD Model Simulation

In Figure 2.7, a simple stock is used to represent a system state with two flows affecting the stock level. The inflow increases the stock by a certain value and the outflow decreases the stock by another value. The stock value at any given time (t), where the initial time value is (t_0) and initial stock value is ($s_{0)}$, is given by the integral Equation (2.3).

$$Stock_t = Stock_{t0} + \int_{t0}^{t}[(Inflow\ (s) - Outflow\ (s)]ds$$

(2.3)

Equivalently, the net rate of change of any stock at time (t) is the derivative of inflow less the outflow (Equation (2.4)).

$$d(stock)/dt = Inflow\ (t) - \ Outflow\ (t)$$
(2.4)

Figure 2.7 Stock, Inflow and Outflow Diagrams

The last issue in the SD mathematical modeling is the auxiliary variables. It is stated by Sterman (2000) that the SD model requires only stocks and their rates of change to be described by mathematical equations, but for better communication and clarity, it is helpful to define auxiliary variables. Auxiliaries and flows are functions in stocks and some other auxiliaries. Equations (2.5) and (2.6) give a generic mathematical expression for the flows and auxiliaries.

$$\text{Auxiliaires} = \; f\,(Stocks, Auxiliaries) \qquad (2.5)$$

$$Flows = \; f\,(Stocks, Auxiliaries) \qquad (2.6)$$

2.6 SD Application in Management

Many SD models have been developed to address issues of dynamics inherent in construction operations. A wide range of SD models exist in the literature, ranging from full scale models that represent a group of projects or organizations in a single model (Levitt et al. 1999), to low scale SD models that address a single project. This research focuses on modeling a single project with the acknowledgement that a single project might include many modules. The SD models vary in the level of detail depending on the problem addressed and the preference of the modeler. Some SD models have detailed descriptions and others are highly abstract.

Robert (1974) was the first to develop the SD work cycle model called '*Work To Do'* for a construction project. The model accounted for the resources, productivity, actual progress, and perceived progress. Modelers later on used Robert's model as a base model to suggest enhancements and improvements. Project features such as the development stages of a project and management aspects were included by Cooper (1980) and Richardson & Pugh (1981). A quality assurance cycle and a rework cycle were modeled by Abel-Hamid (1984), nonlinear constraints imposed on work availability and progress by Homer et al. (1993), the concurrence constraints limiting the execution of work in parallel by Ford and Sterman (1998), releasing completed work to downstream by Ford (1995), managing fund contingency by Ceylan and Ford (2002), creating schedule buffer and dynamic planning by Park and Pena-Mora (2003), and managing iterative errors and change cycles by Lee (2005). In the following sections, the features included in the aforementioned models are described and discussed.

2.6.1 Rework Cycle

The rework cycle is included in most of the developed SD models. This cycle recognizes that the completion of a project task may be defective, resulting in a need for rework. Rework can itself be flawed, requiring additional rework in a recursive cycle that can extend project duration and workload beyond what is originally conceived. In the absence of the rework cycle, project completion is a function of the number and scope of tasks, the available resources, and their productivity. By considering defects, quality, and testing through a rework cycle, many path-dependent reinforcing loops are generated (e.g., burnout, error) that critically affect the fate of projects (Lyneis and Ford 2007).

Cooper (1993a) analyzed more than 60 large projects in different management disciplines to build a computer-based model capable of precisely capturing the performance of large projects from the design stage to the completion stage. Large portions of the models analyzed were incapable of simulating real behavior. The major part missing in the modeling process was the rework cycle. The traditional methods of modeling or scheduling treat the project as being composed of a set of individual, static, and discrete tasks. They tend not to account for the flaws in work and the needs to rework. Cooper (1993 a) stated "*indeed the analysis have shown that the rework can account for the majority of content on complex development projects".*

Figure 2.8 demonstrates the performance outputs of a typical simulated construction project without a rework cycle. The outputs demonstrate that including the rework cycle in the model produces an approximation of the real results. The project rework cycle is included as part of the project dynamics. Figure 2.9 shows the SD model that includes the rework cycle (Abdel-Hamid 1984). The model includes four stocks of work (Work to be Done, Undiscovered Rework, Rework to Do, and Work Done). When the project starts, all work resides in the stock "Work to be Done." Progress of the work depends on the project resources and productivity, and a part of the work being done contains errors, which flows to the stock "Undiscovered Rework." Approved work that is executed according to the required quality enters the stock "Work Done". Errors are not immediately identified, but detected at downstream work checking. Once errors are discovered, the work enters the stock "Rework to Do" and more resources are needed to correct flawed work. Sometimes reworking the flawed work generates more rework.

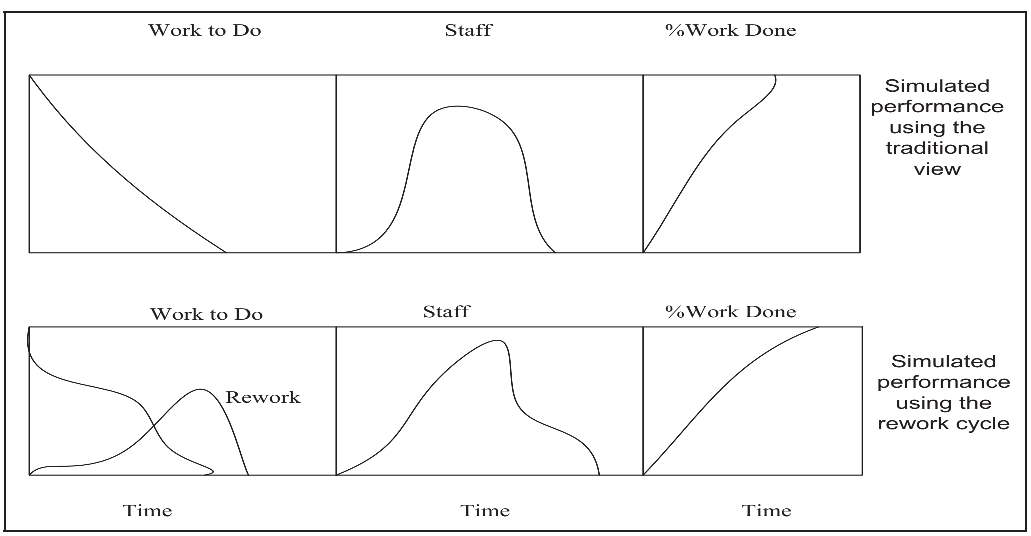


Figure 2.8 Simulation of Project Performance using Rework Cycle (Cooper 1993 b)

Other researchers such as Cooper (1993 b), Ford and Sterman (1998; 2003) and Park and Pena-Mora (2003) have developed models that involve rework cycles not drastically different from Abdel-Hamid (1984), but with some features enhanced and others added.

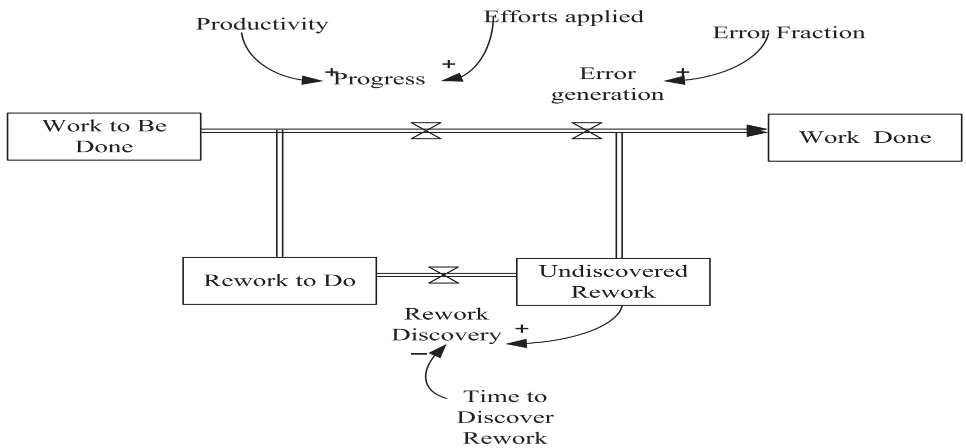


Figure 2.9 The Project Rework Cycle (Abdel-Hamid 1984)

2.6.2 Controlling Feedbacks

A project is measured in terms of scope, schedule, cost, and quality. While modeling the controlling feedbacks of a project, modelers focus on the information processing of project managers (Rodrigues 1996; Lyneis and Ford 2007). Managers' decisions are primarily based on bridging the gap in between the project's actual performance and its planned performance. Examples of possible strategies used are overtime, hiring more workers, or intensifying the work (Sterman 2000). These strategies involve delays, and as previously discussed, delays have an adverse impact on the project.

2.6.3 Ripple Effects

Decisions or correction measures taken by project managers to recover from the slippage in project performance have side effects. These effects generate resistance to decisions adopted (anti policy) called *ripple effects.* Ripple effects manifest themselves through nonlinear relationships, and consequently reduce productivity and generate more errors in the work (Lyneis and Ford 2007). Ripple effects can be best described by the following example. First, the productivity of newly hired workers is usually less than what has been anticipated by managers. This is due learning curve effects, skills, and familiarity with the nature of the work, and as a result, productivity is negatively impacted. Second, new workers need training and this results in lost hours (Abel-Hamid 1984; and Rodrigues 1996). Third, having more workforce than what the workspace capacity permits will eventually causes congestion and communication difficulties. Fourth, adopting an overtime strategy will results in fatigue, consequently deteriorates quality and productivity. Fifth and last, intensifying the work also results in errors and fatigue. These five examples show how the ripple effects are generated in construction operations based on different strategies adopted.

2.6.4 Knock-on Effects

The ripple effects generate secondary and tertiary feedback loops; some are due to the workflow of the project and others due to human reactions to project conditions (Lyneis and Ford 2007). Most of the ripple and knock-on feedbacks modeled are attributed to the internal structure of the project; however, other feedbacks can be generated from external factors to the project such as scope change by client (Rodrigues and Williams 1998; Mckenna 2005). While the ripple and knock-on feedbacks are included in the SD models to improve model accuracy, it is rare that the secondary consequence of adjusting the project target has been investigated (Lyneis and Ford 2007).

At this point of the literature review, the distinctive characteristics of SPM, OPM, DES and SD are discussed in detail. The pros and cons of each are also mentioned, followed by a detailed discussion conducted on SD modeling techniques and strategies. The above discussions justify the research goal of enhancing the current management practice through hybrid modeling. The above sections provide the foundation for the following discussion.

2.7 Comparison between DES and SD

Before investigating the existing hybrid DES_SD simulation applications, it is imperative to conduct a comparison between DES and SD simulation philosophies, as shown in Table 2.4. Research on the comparison of DES and SD is scarce; existing comparison studies tend to be biased toward either the DES or SD. Most of the views in the comparisons are expressed from the author's personal view as well as area of expertise (Brailsford and Hilton 2001; Tako and Robinson 2009). From the comparison demonstrated in Table 2.4, it can be stated that DES focuses on the daily operations at the activity level, while SD focuses on the holistic project level and its evolution over time.

Table 2.4 Comparison of DES and SD Modeling

Aspect of Comparison	DES	SD	Author(s)
Problem scope	Tactical operational	Strategic	(Sweetser 1999; Lane 2000; Rabelo et al. 2005)
Feedback effects	Models open loop structures	Models causal relationships and feedback effects.	(Coyle 1985; Sweetser 1999; Brailsford and Hilton 2001)
System representation	Analytic view	Holistic view	(Baines et al. 1998; Lane 2000; Rabelo et al. 2005)
Complexity	Narrow and focus on complexity and details	Wider focus, general and abstract system	(Lane 2000)
Data type	Quantitative	Qualitative	(Sweetser 1999; Brailsford and Hilton 2001)
Randomness	Random variables (Statistical distribution)	More deterministic	(Meadows 1980)
Validation	Black-box approach	White-box approach	(Lane 2000)
Model Results	Provide a statistically valid estimates of system performance	Provide a full picture, qualitative and quantitative of system performance	(Meadows 1980; Mak 1993)
State change	At discrete points in time	Continuous	(Morecroft and Robinson 2005; Rabelo et al. 2005; Han 2008)
Level of model complexity	Increases exponentially	Increases linearly	(Morecroft and Robinson 2005)

2.8 Hybrid Models of DES and SD

The hybrid DES_SD is defined as "an integrated model that incorporates both DES and SD models within hybrid simulation environment." Although construction projects exhibit coupled strategic and operational levels as shown in Figure 2.10, limited research has attempted to capture the cross-functional interactions between the two levels. These two levels are responsible for the dynamics and complexity inherent in construction projects. According to the changing behavior of system variables, three possible interactions between the continuous and discrete variables can occur, as shown in Figure 2.11 (Pritsker et al. 1997). First, a discrete change in a variable may cause a discrete change in another continuous variable. Second, a continuous change in a variable by reaching a threshold may cause a discrete change in interacting variables. Third, a discrete change in a variable may change the function describing the continuous variables.

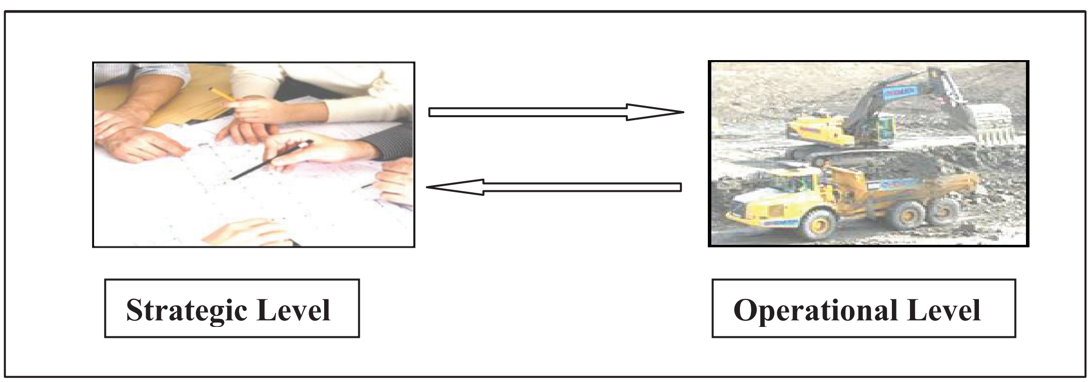


Figure 2.10 Construction Strategic and Operational Levels Interaction

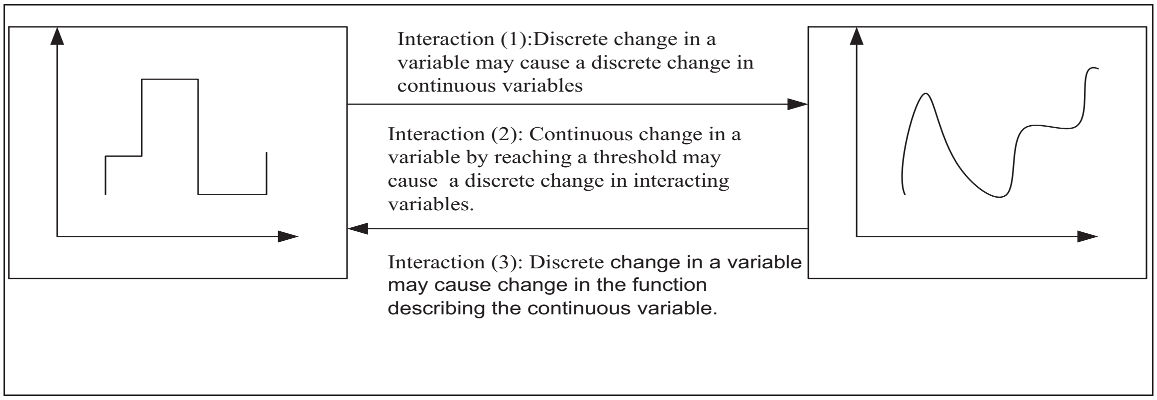


Figure 2.11 Interactions between Discrete and Continuous Variables (Pritsker et al. 1997)

As construction becomes more complex and integrated, decision-making has been facing increasing challenges. Proper modeling and simulation can help make appropriate decisions. However, the available simulation tools do not allow modeling of certain influential aspects of a project. For instance, the objective of strategic project management is to depict the project behavior over time based on plans and strategies applied, then measure their impact on the project (e.g., if new workers are hired, how does this affect the overall worker productivity and how does the project progress). On other hand, the objective of operational project management is the detailed analysis at the tactical level (e.g., if the number of trucks increased for earthmoving work, how would it be allocated to optimize project productivity). These problems are better addressed with tools such as a hybrid of the DES and SD methods.

Most of the existing simulation tools that are used in construction modeling have been developed based on the DES philosophy. Few of these simulation tools have been developed using SD philosophy. Each tool has its own strengths and weaknesses. SD models attempt to test management decisions against the unstable operation's inner and outer environment. The major drawback of SD models is the inability to represent the operational details, while DES is a powerful tool to do so. Since construction operations are neither completely discrete nor continuous, building models with discrete-event and continuous simulation becomes a necessity. For instance, the management sets the project scope and objectives, then decisions are made by management before and throughout the project execution to achieve these goals. Tracking the project's progress is conducted periodically to check the effectiveness of the strategic decisions, and based on the evaluation results, corrective steps might be taken. On the other hand, the execution of these strategies cannot be carried out without the operational details of the project processes. The operational details provide the means to implement strategies.

2.9 Existing Methods for Developing Hybrid DES_SD Simulation Models

The interest in developing hybrid DES_SD models first appeared managing software projects in computer science and control system fields (Maler et al. 1992; Kowakewski et al. 1999 Rus et al. 1999; Pepyne 2000; Zeigler et al. 2000; Martin and Raffo 2001; Lee et al. 2004). Three methods are widely used in the literature to develop hybrid DES_SD simulation systems: 1) hybrid state machine (Harel 1987; Maler et al. 1992); 2) DEVS and DESS formalism (Ziegler et al. 2000) and 3) distributed Simulation using HLA (Kuhl et al. 1999). In the following subsections, these three methodologies are discussed in detail.

2.9.1 Hybrid State Machine

The state machine is a traditional object-oriented way to describe the discrete behavior of a system and to document how an object responds to events (Harel 1987). One of the approaches to model hybrid systems is to assign algebraic differential equations that describe the continuous behavior of a system to the state machine (Borshchev et al. 2000). Maler et al. (1992) added, to Harel's state machine, a concept called *phase transition* (differential equations) to account for the continuous behavior of system. This resulted in hybrid state machine used to model hybrid systems. Hybrid simulation systems built using the state machine are composed of a discrete engine and equation solver, where the discrete engine maintains virtual time and the discrete events created take care of concurrency, synchronization, etc. The equation solver numerically solves systems of algebraic differential equations supplied by the discrete engine. At the beginning of each time step, the discrete engine invokes the equation solver, giving it the current global equation system and a stop time, which is the time when the next discrete event is scheduled. While solving the system, the equation solver makes a periodical callback to the discrete engine to check if the current combination of variable values satisfies any of the change event conditions currently awaited by the model. Through this interaction between the equation solver and the discrete simulation engine, the state of the system changes.

Figure 2.12 demonstrates the state update mechanism of the hybrid state machine, and the process of altering between continuous and discrete states. For instance, for the initial system state S_0 at time T_0, the system might change to state S_1 and then advance time from T_0 to T_1 or it might advance time from T_0 to T_1 and then update the system state from S_0 to S_1. This method of system updating results in a hybrid system simulation called in the software industry control-based hybrid simulation. Hybrid models developed using this methodology have dominant discrete behavior and cannot describe the project context level (Raffo 1995). Only the discrete simulation engine is responsible for changing the system states.

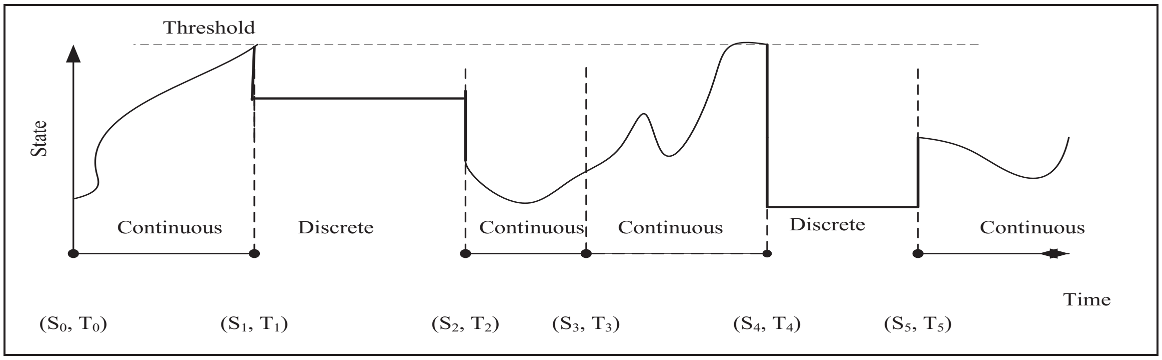


Figure 2.12 Hybrid Simulation Behavior

2.9.2 DEVS and DESS Formalism

(i) DEVS Formalism

DEVS (Discrete Event System Specifications) is a formalism developed by Zeigler et al. (2000) to model discrete event systems that are of combined discrete and continuous nature. In order to describe discrete models, DEVS formalism consists of three sets (inputs set, outputs set, and states set) and four functions (internal transition, external transition, output, and time advance) that are needed to describe the simulation model. DEVS system specification is given by Equation (2.7).

$$DEVS = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, ta) \tag{2.7}$$

Where,

X *is the set of external inputs*

Y *is the set of outputs*

S *is the set of sequential states*

δ_{ext}: $Q \times X \rightarrow S$ *is the external state transition function; it represents the interaction between X, and Q to demonstrate how inputs cause state transitions*

δ_{int}: $S \rightarrow S$ is the internal state transition function; it describes how to transit from state (s) when ta(s) is reached (focuses on transitions from one state to another when time reached).

λ: $S \rightarrow Y$ is the output function

ta: $S \rightarrow \Re_0^+ \cup \infty$ is the time advance function

With $Q = \{(s, e) \mid s \in S, 0 \leq e \leq ta(s)\}$ is the set of total states. Total system states Q is represented by state s and time elapsed e.

System transits from state (s) to another when e = ta(s). When reaching the new state, the elapsed time counter e is set to zero.

(ii) DESS Formalism

DESS (Differential Equation System Specification) is formalism developed by Zeigler et al. (2000) to model continuous models. In DESS, the state transition function of DEVS is replaced by a rate of change function to account for the rates of state variable change. In order to describe continuous models, DESS formalism consists of three sets (inputs set, outputs set, and states set) and two functions (rate of change and output function) that are needed to describe the simulation model. DESS system specification is given by Equation 2.8

$$DESS = (X, Y, Q, f, \lambda) \quad (2.8)$$

Where:

X is the set of inputs

Y is the set of outputs

Q is the set of states

$f : Q \times X \rightarrow Q$ is the rate of change function;

$\lambda: Q \rightarrow Y$*(Moore-type)* or $\lambda: Q \times X \rightarrow Y$ *(Mealy-type) is the output function.*

DESS formalism is used to describe continuous models but this methodology has a limited capacity to capture the feedback loops associated with continuous models.

(iii) DEVS and DESS Formalism

DEVS and DESS are integrated as demonstrated in Figure 2.13 to describe a hybrid simulation model.

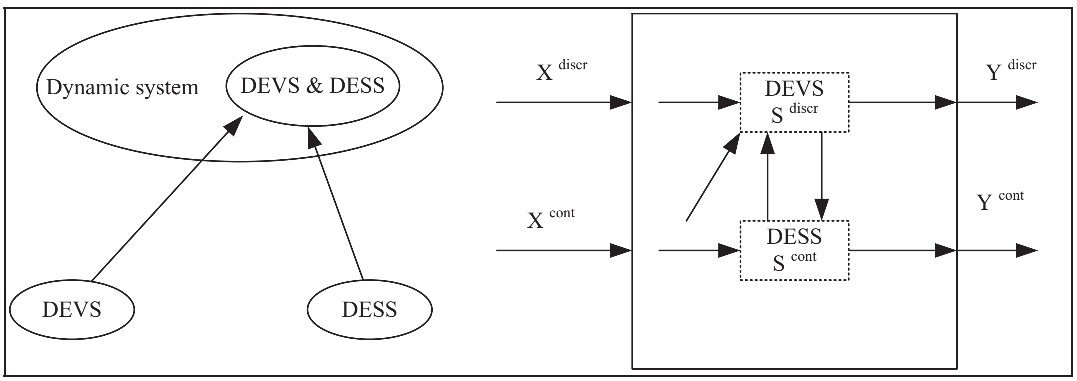


Figure 2.13 DEVS and DESS Combined Model (Zeigler et al. 2000)

The combined DEVS and DESS formalism consists of five sets and six functions as given in Equation (2.9).

$$DEVS\ and\ DESS = (X^{discr}, X^{cont}, Y^{discr}, Y^{cont}, S^{discr}, S^{cont}, \delta_{ext}, \delta_{int}, \lambda^{discr}, \lambda^{discr}, f) \quad (2.9)$$

Where,

X^{discr} & Y^{discr} *are sets of discrete event inputs and outputs, respectively*

X^{cont} & , Y^{cont} *are sets of continuous inputs and outputs, respectively*

S^{discr} & S^{cont} *are sets of discrete and continuous states, respectively*

$\delta_{ext:} Q\ _X X^{discr}\ x\ X^{cont} \rightarrow S$ *is the external transition function*

$$Where\ S = S^{discr}\ X\ S^{cont}\quad is\ the\ sequential\ state\ set,\ and$$

$$Q = \{\ (s,e)|\ s\ \in S,\ e\ \in R_0^+\ \}\ \ is\ the\ total\ state\ set$$

δ_{int} : $Q\ _X X^{cont} \rightarrow S$ *is the discrete event internal transition function*

λ^{discr} : $Q\ _X X^{cont} \rightarrow Y^{discr}$ *is the discrete event output function*

λ^{cont} : $Q\ _X X^{cont} \rightarrow Y^{cont}$ *is the continuous output function*

f : $S\ _X\ X^{cont} \rightarrow S^{cont}$ *is the derivative function*

C : $Q\ x\ X^{cont} \rightarrow Bool$ *is the event detection condition predicate*

The semantics of the DEVS and DESS formalism are given in terms of the subclass of dynamic systems that it defines. A state event is the occurrence of a change in the value of the event condition predicate from false to true. In the DEVS and DESS formulation, the concept of state event is generalized so that it can occur due to a change in any of its arguments, of which the continuous state is one component (Zeigler et al. 2000).

Hybrid models developed using DEVS and DESS have the same mechanism as that of the state machine. DESS causes state events to occur, and then detect those events at the discrete points in the model. For instance, between two events in the DESS, the input, output, and state of a system continue in their continuous behavior until the threshold specified for a continuous part of the model is reached or a condition is met, as shown in Figure 2.14. Thereafter, a discrete change in the system or the variable will take place.

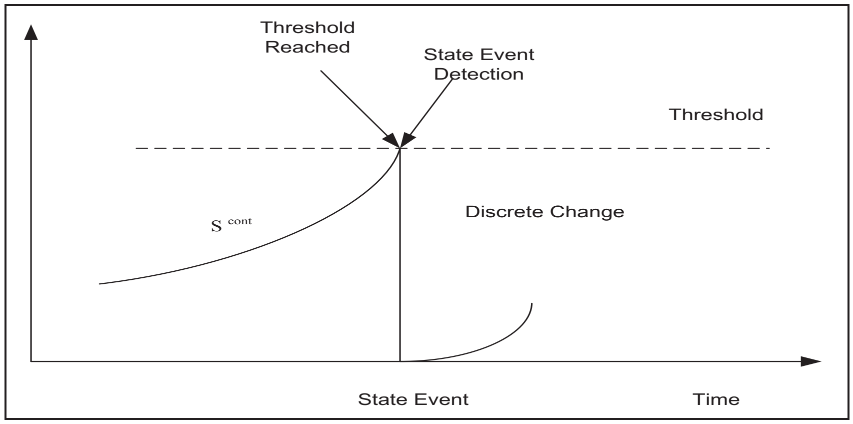


Figure 2.14 Hybrid Simulation behavior using DEVS & DESS Formalism (Zeigler et al. 2000)

The characteristics of hybrid systems developed using the state machine and DEVS-DESS formalism are similar to those in control system theory (direct or regulate the behavior of other systems by an on/off process). For instance, water flows in a channel (continuous behavior) until the tank is filled (discrete behavior). Here, the process goes on/off based on the condition being met or the threshold reached. This mechanism of describing the interactions between the discrete and continuous variables is difficult to apply in construction management. Specifying a threshold to trigger states change from continuous to discrete would rather cause oscillating behavior of the variables. This hybrid simulation technique may work well for the software projects but not for construction as management works to smooth the oscillation. Furthermore, the surrounding environment of software project involves a limited number of variables while in construction projects, the surrounding factors are tremendous and play a major role in the success or failure of the project outcomes. On the other hand, SD is a holistic modeling methodology with causal feedback loops acting as guidelines for the system behavior. Splitting the SD model into small objects as required per DESV and DESS formalism and allowing them to interact indirectly leads to a breach of SD theory (holistic view). Furthermore, an important limitation of DESV and DESS formalism is that interactions between discrete and continuous variables are not triggered through data update demands by any of management levels, but due to a threshold reach, which is precisely what this research is trying to avoid. Every management level should be modeled fairly without permitting the behavior of any simulation method to prevail over the other.

2.9.3 Distributed and Parallel Simulation (DPS)

Parallel/distributed simulation is concerned with issues introduced by distributing the execution of a discrete event simulation program over multiple computers. Distributed simulation is concerned with the execution of simulations on loosely coupled systems where interactions take much more time, e.g., milliseconds or more, and occur less often. It includes execution on geographically distributed computers interconnected via a wide area network such as the Internet. Parallel discrete event simulation is concerned with execution on multiprocessor computing platforms containing multiple central processing units (CPUs) that interact frequently, e.g., thousands of times per second. In both cases, the execution of a single simulation model is distributed over multiple computers. Fujimoto (2001) listed the benefits of using distributed simulations as: 1) reduction in the execution simulation time of large models; 2) integrating different simulators; 3) reusability of existing models; 4) flexibility in model extension and refining; and 5) fault tolerance in which execution of the model continues in spite of partial failure of hardware components.

The SIMNET (SIMulator NETworking) demonstrated the viability of using distributed simulations to create virtual worlds for training soldiers in military engagements (Miller and Thorpe 1995). By the mid-1990s, Distributed Interactive Simulation (DIS) Standards (IEEE Std 1278.1-1995 1995) and Aggregate Level Simulation Protocol (ALSP) were developed (Fujimoto 2000 and 2001). ALSP and DIS have since been replaced by High Level Architecture (HLA) that has a broad range of scope for developing distributed simulations.

(i) High Level Architecture (HLA)

HLA has become the de-facto standard in distributed simulation (Kuhl et al. 1999; McLean and Riddick 2000). HLA is the standard technical architecture for all Department of Defense-USA (DoD) simulations based on a "system of systems" approach (DoD 2000). It has created the standard (IEEE 1516) that describes rules for integrating distributed simulation. HLA is software architecture for creating computer simulations out of component simulations (Kuhl et al. 1999). In HLA, individual simulators are called federates and a group of federates are called a federation. Runtime Infrastructure (RTI) is the software implementation of the HLA framework and it implements the HLA rules as shown in Figure 2.15. The HLA consists of three important components: 1) interface specification, 2) object model, and 3) template (OMT) and HLA rules (Dahmann et al. 1998; Buss and Jachson 1998).

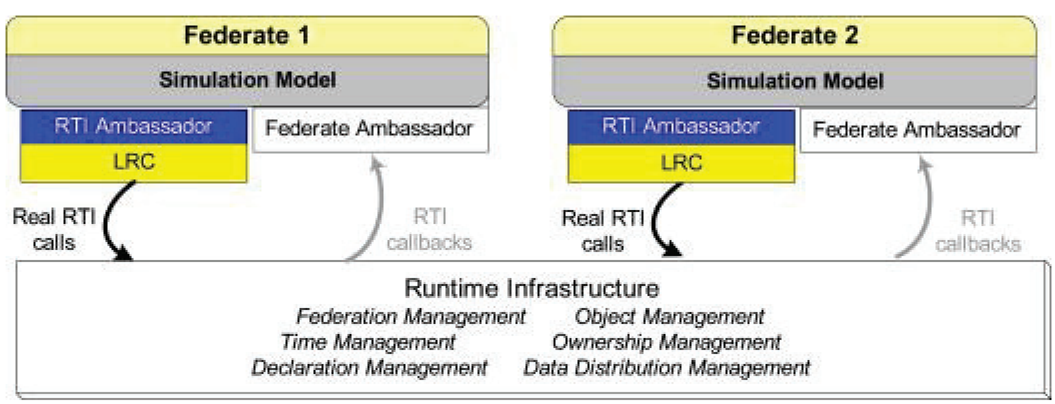


Figure 2.15 HLA Structure Implementation (Chen et. al 2008)

In construction, HLA was used by Alvanchi et. al (2011) to develop a hybrid simulation model of DES and SD models. Distributed or parallel simulation methods are traditionally used to develop large scale models and require experts for their use. The deployments of these methods in construction hybrid simulation are subjected to a good understating of the hardware arrangements and in many occasions require experts in code programming. This may hinder their use in construction where managers are more excited to have simple and applicable simulation tools that similar to the developed DES applications. This research is trying to develop simple and practical method of hybrid simulation that can be adapted and used by managers without prior knowledge of code programming.

2.10 Attempts of Integrating DES and SD models

The hybrid DES_SD simulation has emerged as a promising and useful area of research and application (Pritsker 1997). Despite the potential benefits of hybrid DES_SD simulation, few attempts have been made to develop a hybrid system that integrates the strategic and operational levels of a project. Table 2.5 summarizes most of the developed hybrid DES_SD systems and models in different management science. Martin and Raffo (2001) proposed a hybrid DES_SD model for software projects. The model addressed problems faced by managers when dealing with project's context parameters, such as staffing, training, and overtime, and their impact on productivity and quality. DES was used to capture operational details while SD was used to capture the project's context. The model lacks a full-scale representation of feedback loops and assumes that workload for an activity is constant, which hinders computing the activity duration dynamically. The limitations of Martin's and Raffo's (2001) model were addressed by Choi et al. (2006). They proposed a hybrid software process simulation modeling (SPSM) using the DEVS formalism that was proposed by Zeigler et al. (2000). Choi et al. defined the DEVS_Hybrid_SPSM formalism by extending DEVS to a hybrid SPSM domain. This extension was performed through using numerical integration methods to account for project dynamics. SD was used to account for the details concerning activity behavior and managerial policies, while DES controls the activity start/completion sequence. This method is undermined by the prevailing discrete behavior of DEVS. In general, attempts made in the software industry focused on enhancing the software developing process by modeling variables such as hiring, staffing, overtime, policies, and fatigue along with the project's operational level. These efforts in the area of hybrid simulation resulted in commercial simulation software applications such as Anylogic , ExtendSim and SimuLog.

Other researchers from manufacturing enterprises have also made substantial efforts toward developing hybrid DES_SD systems. This because of the strong interactions between the three management decisions in business systems: strategic, tactical, and operational (Miller 2002). The DES method is used extensively in modeling manufacturing systems. However, due to the emerging needs for an integrated strategic and operational manufacturing management system, DES became insufficient. A strong relationship exists between the strategic and operational levels in manufacturing. Several continuous variables such as customer demand, market condition, organizational structure, system behavior over a long period of time, etc. exist and their impact needs to be estimated and evaluated. To overcome DES limitations in the manufacturing field, differential equations were used in hybrid simulation architectures developed to simulate the supply chain. This allowed for modeling of the heterogeneous variables such as discrete and continuous.

Table 2.5 Summary of the developed hybrid DES_SD Models in Different Fields

No	Author(s)	Tools used	Model Description	Comments
1.	(Martin and Raffo 2001)	Hybrid DES_SD model for software development projects	The model attempts to address project context using SD and project process using DES. The model helps managers to model variables such as staffing, training, and the effect of overtime on quality. It investigates the effects of discrete resource changes on continuously varying productivity.	The required workload is constant during the life-cycle, so it does not represent the feedback structure properly, where work load increases because of reworks.
2.	(Lee et al. 2002a)	Simulation with discrete-continuous combined modeling	Architecture of combined modeling for supply chain simulation. Equations for the continuous portion of the SC model are used to overcome the DES limitation.	Not enough details on how the two paradigms are communicated
3.	Venkateswaran & on 2005)	Hybrid DES_SD Architecture using HLA and optimizer	Hybrid simulation-based hierarchical production planning architecture consisting of SD for enterprise level planning and DES for the shop-level scheduling	Integer programming makes decisions static and arbitrary daily data interaction.
4.	(Choi et al. 2006)	Hybrid model using DEVS formalism.	Hybrid software process simulation. DEVS_Hybrid_SPSM formalism by extending DEVS to the hybrid SPSM domain	DES is the platform for developing the hybrid DES_SD model, hence the DES paradigm is dominant. The model is limited to small software projects.
5.	(Lee et al. 2006)	SD and dependency structure matrix planning	Dynamic planning and control methodology, to support both the strategic and the operational aspects of project management, by integrating SD with a network-based tool.	The methodology does not involve DES, though they propose extending the work to include it. SD is the core simulation engine.
6.	(Lee et al. 2007)	Hybrid DES and SD	Theoretical study demonstrates the needs to develop hybrid DES_SD system for construction management.	The study provides justified results to encourage further research in hybrid DES_SD fields.
7.	(Rabelo et al. 2005)	Hybrid DES and SD. HLA used to facilitate the DES module interactions with SD model	Enterprise simulation: A hybrid system approach. Attempt at comprehensive modeling of manufacturing enterprise using distributed simulation.	Preliminary analysis of potential DES and SD integration. The enterprise model is developed as a distributed simulation model, in which the SD and the DES models run separately and data is exchanged between them manually.
8.	(Rabelo et al. 2007)	Hybrid DES and SD	Value Chain Analysis Using Hybrid Simulation and AHP to investigate decision, demand, customer satisfaction, and profits.	Detailed estimates of production and lead times are required from the DES model to input into the SD model.
9.	(Helal et al. 2007)	Hybrid DES_SD using set theory	Methodology to integrate and synchronize DES and SD in manufacturing enterprise	The methodology uses In_port and out_Port for communication of data between the models. Only a top-down hybrid structure is used.
10.	(Umeda & Zhang 2008)	Hybrid DES and SD	Modeling supply chain by using DES for shop floor and SD for customer satisfaction of the manufacturing enterprise.	Propose integration of DES and SD without explanation of the methodology.
11.	(Pena-Mora et al 2008)	Hybrid DES and SD with using *MatchFactor*.	Concept of integration is provided by using earthmoving as a case study to verify the concept. *MatchFactor* is used to incorporate the context level represented in the management actions.	Only feedback loops are used from SD. Other elements of SD such as stocks and flows are not used. The model proves the need to develop a generic hybrid DES_SD system.
12.	(Lee et al. 2009)	Hybrid DES_SD model based on Pritsker (1995)	Integrating the construction operational using DES and context using SD in Large-Scale construction	High computing time. The model developed using AnyLogic 6 in which the DES method prevails.
13.	(Alvanchi et al. 2011)	Hybrid DES_SD. The communications is based on HLA.	Architecture of hybrid DES_SD to track the dynamic behavior of construction	It is the first initial detailed efforts in developing hybrid DES_SD architecture in construction. The authors recognized that further efforts are needed for enhancements.

Distributed simulation has also been utilized by other researchers to model hierarchical production planning (Venkateswaran and Son 2005). The proposed method of hybrid simulation consisted of the SD model for enterprise level planning and the DES model for shop-level scheduling. The architecture consisted of an optimizer to select the optimal set of control parameters based on the estimated modeled system behavior. Feedback loops are used at each level to monitor the performance and update the control parameters. The models are interfaced using HLA in a distributed simulation environment. As the models interactions need to be refined and enhanced, the authors stated they are working on improving this issue, but no further work was traced in the literature.

A hybrid DES_SD simulation methodology for simulating the manufacturing enterprise with a focus on policy design and control was proposed by Rabelo et al. (2005); Rabelo et al. (2007) and Helal et al. (2007). The model consists of a generic SD model for the enterprise top management level and a number of DES models for selected units at the operational level. The DES models interact with the generic SD model in an integrative feedback approach. The model consists of: 1) internal supply chain; 2) strategic decisions related to resource allocation and financing function, 3) suppliers, 4) customers satisfaction; and 5) DES models used for production units and internal business units. The integration between the DES and SD models is achieved using formalism adapted from Ziegler et al. (2000). The Arena software system developed by Rockwell Automation is used to model the DES model while Vensim developed by Ventana Systems, Inc. is used for the SD model.

In construction, a limited number of researches have been found in the area of hybrid DES_SD modeling and simulation. Research has focused on the potential benefits of integrating strategic/context and operational levels of construction operations by using a hybrid simulation (Lee et al. 2006). The SD method was integrated with CPM network-based tools to account for the static nature of the CPM tools. The focus was mainly on developing a special purpose dynamic planning and control methodology (DPM), to support both strategic and operational aspects of project management, in addition to accounting for the causal-effect loops (Lee et al. 2006). SD is used to model the iterative cycle of error generation in construction operations, while the dependency structure matrix (DSM) is used as the interface to input activity characteristics. The method consists of: 1) the strategic core (SD) that works as the main simulation engine to represent project behavior; 2) a tactical layer to improve the operational aspects of DPM, by including a CPM network-based tool; and 3) an operational layer, which modifies the strategic core; and 4) an interface layer for communication between the four layers and the users. The proposed methodology does not utilize the DES in its operational scheduling to account for uncertainty. The method mainly focused on developing a special purpose application that accounts for the dynamics generated from the errors iterative cycle, generated while executing construction operations. A study on the potential benefits of the integrated context and operational level of construction projects was conducted by Lee et al. (2007). The study concluded that despite the need for a simultaneous consideration of the construction context and operation, there has been little effort to integrate them.

The benefits of integrating SD and DES on a single computation platform were presented with an earthmoving example by Pena-Mora et al. (2008). In this study, the operational level of the earthmoving project was modeled using the STROBOSCOPE software system. The MatchFactor concept (Smith 1995) was used to synchronize the truck circulation rate with the loader circulation rate for optimum cost effectiveness and to trigger management actions. Pritsker's (1995) principals have been deployed to capture the interaction between the DES and SD variables. The proposed model is used to demonstrate the need for hybrid simulation techniques and tools that account for heterogeneous aspects. Pritsker's (1995) three principles that were used in the model can best describe the physical control system, not the management context. Hybrid models built using control theory (e.g., Pritsker's 1995) categorize SD for the lower level of the system and DES for the higher level. The hybrid model behavior is controlled by the DES model only. However, the proposed hybrid model was able to highlight the improvements in physical representation and results. The study did not include a generic framework and synchronization tool for integrating DES and SD methods. The authors stressed, first, the need for a hybrid SD_DES framework, and second, for integration procedures. Along the same track, Lee et al. (2009) developed a hybrid DES_SD model for a pipeline installation process. The model was built using AnyLogic 6 software system (XJ Technology Inc.2008).

The most recent attempt in construction operations simulation to integrate DES and SD models used distributed simulation technique (Alvanchi et al. 2011). A framework and architecture of a hybrid SD_DES for construction modeling and simulation was proposed. The method attempts to capture the mutual effects of construction operations and the project context level by using DES and SD. It uses HLA and Anylogic software systems as platform for developing the simulation models. The method lacks a novel approach in synchronizing DES and SD simulation clocks on a single computational platform. In addition, it lacks the required components in its architecture for developing a hybrid computer application. However, an enhancement in the hybrid simulation practices in construction operation simulation using the existing tools was presented. The author stated that the purpose of the research was to present a hybrid simulation model rather than a hybrid simulation application.

2.11 Commercial Hybrid DES_SD Software Systems

The efforts of the software industry in the area of hybrid simulation have resulted in commercial simulation software systems such as ExtendSim, Anylogic and SimuLink. Anylogic supports building SD models and allows for importing SD models built in Vensim software systems. It is popular software and its characteristics are discussed in the following paragraphs.

AnyLogic is a software system application developed by XJ Technologies Company (http://www.xjtek.com). It supports building DES, SD, and AB models in an object-oriented environment. Anylogic was developed based on the concept of combining discrete and continuous simulation models as proposed by Harel (1987) and Maler et al. (1992). The modeling language of Anylogic is an extension of UML-RealTime (RT), and the main building blocks of the model are based on the *active object*. An active object is an instance of an active object class. Figure 2.16 demonstrates the architecture of the AnyLogic simulation engine. Anylogic's simulation engine computes hybrid models by triggering a discrete simulation engine that solves the discrete part of the hybrid model. Thereafter, it generates a set of differential equations to account for the continuous part of the model. Differential equations are solved by the solver. If the solution of the equations crosses a threshold condition, the variables involved will be updated, and a discrete event that changes the state of the variable will be triggered. Otherwise, the simulation engine continues updating continuous variables until the next scheduled discrete event.

Two mechanisms can be noticed during the event executions. First, the discrete simulation engine alters the event's state only when a condition is met or a threshold is reached by an event, then the simulation engine generates a new set of equations to replace the executed set. In this mechanism, discrete behavior affects the continuous behavior.

Second, when the preset condition is not met or the threshold is not crossed, then computation continues to update continuous variables until the next event. In this way, the continuous behavior of variables controls variables of discrete behavior.

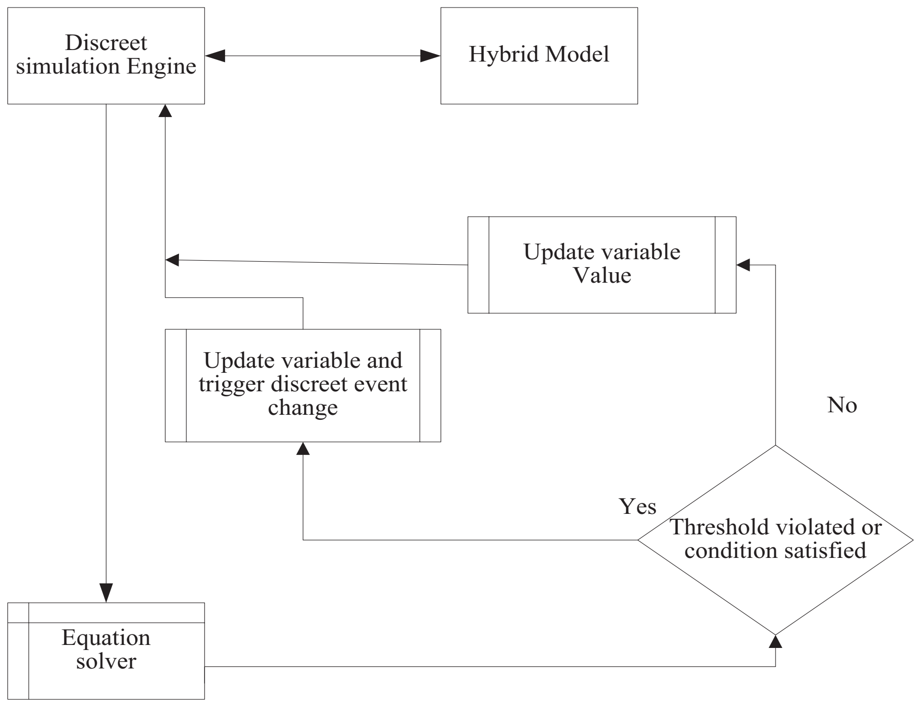


Figure 2.16 Hybrid Simulation Engine Architecture in AnyLogic

Anylogic architecture, as discussed earlier, is partly based on the state chart that was proposed for control systems, but with the extension of UML_RT as modeling language. In construction, using software systems developed based on the state chart limits the model's accuracy as construction operations involve many variable interactions. The number of states in construction variables can be numerous, and using the state chart which requires listing all the possible states would be difficult. Using a threshold to trigger state change from continuous to discrete would rather cause an oscillating behavior of the variables. This might work well for the management aspects of software projects, but not for construction, where management works to smooth out the oscillations.

2.12 Simulation Time Management and Synchronization in Hybrid DES_SD Models

Time management or synchronization means the execution of events in distributed simulation in a correct order and ensures that repeated executions of a simulation with the same inputs produces identical results (Fujimoto 2003). Time management algorithms broadly fall into two categories, termed conservative and optimistic synchronizations. These time approaches are mainly developed to serve the execution of multi-simulation programs on multiprocessor computing platforms (Parallel Simulation) or to execute simulations on geographically distributed computers interconnected through a network (Distributed Simulation). In both cases, the execution of a single simulation model, composed of several simulation programs, is distributed over multiple computers (Fujimoto 2001).

Simulation time in DES advances at the occurrence of events and subsequently system states are updated. Events usually occur at different time points. The state of SD models is updating continuously and the modeler just interrogates the system for its state at every time step. Two important issues are observed based on this mechanism. First, SD updates states at equal time points while DES updates states at unequal time intervals. This an important note used in developing the new synchronization method for integrating DES and SD simulation times. Second, SD models are a continuous system and not capable of generating events like DES models. The concerns of synchronizing DES and SD modules arise from the fact that hybrid models require a creative method that synchronizes the simulation time clocks of both methods.

The area of distributed simulation is rich in synchronization mechanisms developed to synchronize different DES simulators and models. The time synchronization algorithm assumes that simulation consists of a collection of logical processes (LPs) that communicate by exchanging time-stamped messages or events. Hence, the goal of the synchronization mechanism is to ensure that each LP is processed in proper order. The developed synchronization time algorithms broadly fall under two main categories, conservative time management (CTM) and optimistic time management (OTM).

2.12.1 Conservative Time Management (CTM)

CTM means that the synchronization algorithm takes precautions to avoid local causality constrains. These mechanisms usually assume that the simulation model consists of a collection of logical processes (LPs) that communicate by exchanging time-stamped messages or events. The goal of the synchronization mechanism is to ensure that each LP processes events in timestamp order. This requirement is referred to as the local causality constraint (Fujimoto 2001; Fujimoto 1999). For instance, if an LP is at a simulation time of 20 minutes, a conservative protocol guarantees that no event has an LP simulation time of less than 20 minutes. The first algorithm generating CTM was developed by Bryant (1977) and Chandy and Misra (1978). Each LP sends a message with a non-decreasing time stamp to support interactions between models. The communication network ensures that messages are received in the same order as they were sent from the LP. Messages organized in order (first-in, first-out) are the same as scheduled event execution. The simulation process starts with the event of lowest time step. Local events scheduled within the LP can be handled by having a queue within each LP. When the message queue of any model become empty, the process becomes deadlocked and cannot proceed any more. Null messages are used to avoid this deadlock. They have timestamps that cannot create any event or update model states. Null messages introduce a key property called the lookahead concept. If the LP is at simulation time T, it guarantees that any message sent in the future will have a timestamp at least T+L. Then, LP is said to have a Lookahead of L period. The null messages algorithm results in an excessive number of null messages, which is not efficient (Fujimoto 2001). However, this method generates a high computational overhead to ensure sequence time advancement and requires enormous memory (Mattern 1993). This type of time management is used when the causality constraint is unlikely to be violated.

(i) Time Bucket Synchronization Method

The Time Bucket concept is one of the conservative time synchronization algorithms. It is classified as a conservative synchronization mechanism. It was developed to synchronize distributed DES simulators. The Time Bucket synchronization means dividing the overall DES model's simulation length into small intervals of time called a time bucket. Then, the simulation models are allowed to interact or interface at the end of the time interval (Steinman 1990). The time bucket size should be large enough to overcome any overhead computations and small enough to capture any radical change in the system state. One of the main drawbacks of the Time Bucket is an inability to capture event states that have an event time less than the time bucket size.

2.12.2 Optimistic Time Management (OTM)

Unlike CTM, OTM methods allow a violation of the local causality constraint, but they also allow a detection of the violations (processing event of higher timestamp before receiving event with lower timestamp), and recovery from it. The OTM methods have two important features, first, they have a tendency to exploit a greater degree of parallelism in the execution process, and secondly, the synchronization mechanism is more transparent to the application program than in CTM methods. OTM methods require more computations as they need to recover from causality constraint violations. The Time Warp (TW) algorithm (Jefferson 1985) is the best-known optimistic method of time synchronization. It allows free simulation time advancement. When a causality violation occurs, TW rolls back and reprocesses these events in timestamp order. This requires restoring the state which existed prior to the violation. Anti-messages sent to the same queue cancel the previously sent messages (Fujimoto 2001). Two problems arise in this situation. First, certain computations' input/output of operations cannot be rolled back. Secondly, computations consume more memory between sending, un-sending, and roll back of messages. Both problems are solved by Global Virtual Time (GVT), which is a lower bound on the timestamp of any future rollback, and any data stored for the LP before GVT will be destroyed. TW is sometimes overly optimistic (Steinman 1993) and involves many process cancelations. The conservative and optimistic time synchronization approaches were developed to synchronize large discrete event simulation models that have similar state updating and time advancing mechanisms. Distributed simulation is best used for simulating large models, which is not the case in construction. Running simulation models on multiple processers and networks is a complex task and time consuming for modeling construction operations.

2.12.3 Time Management in HLA

Time management in HLA supports CTM and OTM within their federates. Time management in HLA requires a synchronization algorithm to advance the simulation clock using an event stamp or timestamp. It is more flexible, designed to accommodate a wide variety of applications. Message ordering from federates and time advance mechanisms are the two major principals in HLA time management (Fujimoto 2003). Run Time Infrastructure (RTI) provides a conservative time management service to coordinate the message exchange between federates. It ensures that a federate is not advanced to simulation time T until it guarantees that no time stamp messages less than T remained unprocessed. Optimistic time management is also supported by HLA. Optimistic execution allows federates to process events through messages with a smaller time stamp that may later arrive (violation of causality). It should be noted that federates provide roll back service. These time management synchronizations were developed for discrete distributed simulation models and limited attempts are found in the literature for synchronization of discrete and continuous simulation in fields other than construction.

2.13 Summary

This chapter has presented an exhaustive literature review of the state of research in the simulation field. Project management splits the decision-making process into strategic and operational, so developing a simulation model must consider these management decision levels and their characteristics. DES and SD methods have gained a wide range of applications in social science, with a superiority of the DES method in construction operation simulation. However, both methods have limitations associated with their way of approaching the simulation problem. Thus, hybrid simulation of the DES and SD techniques has emerged to overcome such impediments. The hybrid simulation has been implemented in many fields such as manufacturing, health, and recently construction. Adapting these tools for construction operation simulation needs investigation as construction is unique and involves many variable interactions. The DES and SD methods are different in the mechanism of states update; therefore, there is a need to resolve issues that hinder the full integration of both methods (e.g., simulation clock, interfacing, and model structure) from the perspective of construction project nature. Investigating a variety of hybrid simulation systems has provided an insight into the problem being studied and pointed to the shortcoming of the current practice of simulation in construction. The outcome of the analysis stage conducted in this chapter is summarized in the following points.

- The integration benefits of DES and SD on a single platform for construction simulation and modeling have been well argued and established. However, a generic hybrid simulation system for building hybrid simulation models and applications has not yet been developed. The current state of the research, as reviewed, focused on the potential benefits that construction operation modeling could gain through the integration of DES and SD methods.

- The area of hybrid simulation has only been explored very recently in construction modeling. Therefore, the research in this area is scarce and in its initial stages.
- The developed hybrid simulation systems originated in fields other than construction. This presents many hurdles for the applicability of these systems to construction. In construction, numerous variables interact to generate the project behavior. Therefore, it is required to consider the uniqueness and complexity of the construction industry.
- The developed synchronization methods and algorithms were initiated to integrate discrete simulators, and allow the discrete behavior to be dominant in hybrid models. This contradicts the objective of this research where both simulation methods should be deployed to address the simulation problem fairly and based on specialization.

The aforementioned limitations are raised while reviewing the hybrid simulation literature in construction and other fields. These limitations are further addressed in Chapters 3 and 4.

CHAPTER 3

METHODOLOGY

3.1 Methodology Overview

This chapter provides a detailed explanation of the research methodology followed to realize the research objective of this thesis. The research strategy was designed based on the process of solving engineering problems (study/analysis, development, implementation, and validation). The methodology as illustrated in Figure 3.1 encompassed five main phases, namely, *analysis phase (I), development phase (II), implementation phase (III), validation phase (IV) and Conclusion phase (V)*. Initially, in the analysis phase (I), a wide range of literature pertaining to the state of research in construction operations simulation and other fields were reviewed. This helped frame the research problem statement and the objective. The outcomes of the analysis phase have pointed out the pitfalls of the current practice of simulation applications in the construction field. These limitations have triggered a course of action, which are addressed in the development phase (II). The six main components of the developed hybrid simulation method are described in phase (II). The implementation phase (III) involves implementing the developed hybrid simulation method using a real-world case study adopted from the construction field to demonstrate its use. The validation phase (IV) involves validating the SD model, the hybrid simulation model, and the developed hybrid simulation application. Finally, conclusions, recommendations, limitations, and future enhancements are summarized in the conclusion phase (V).

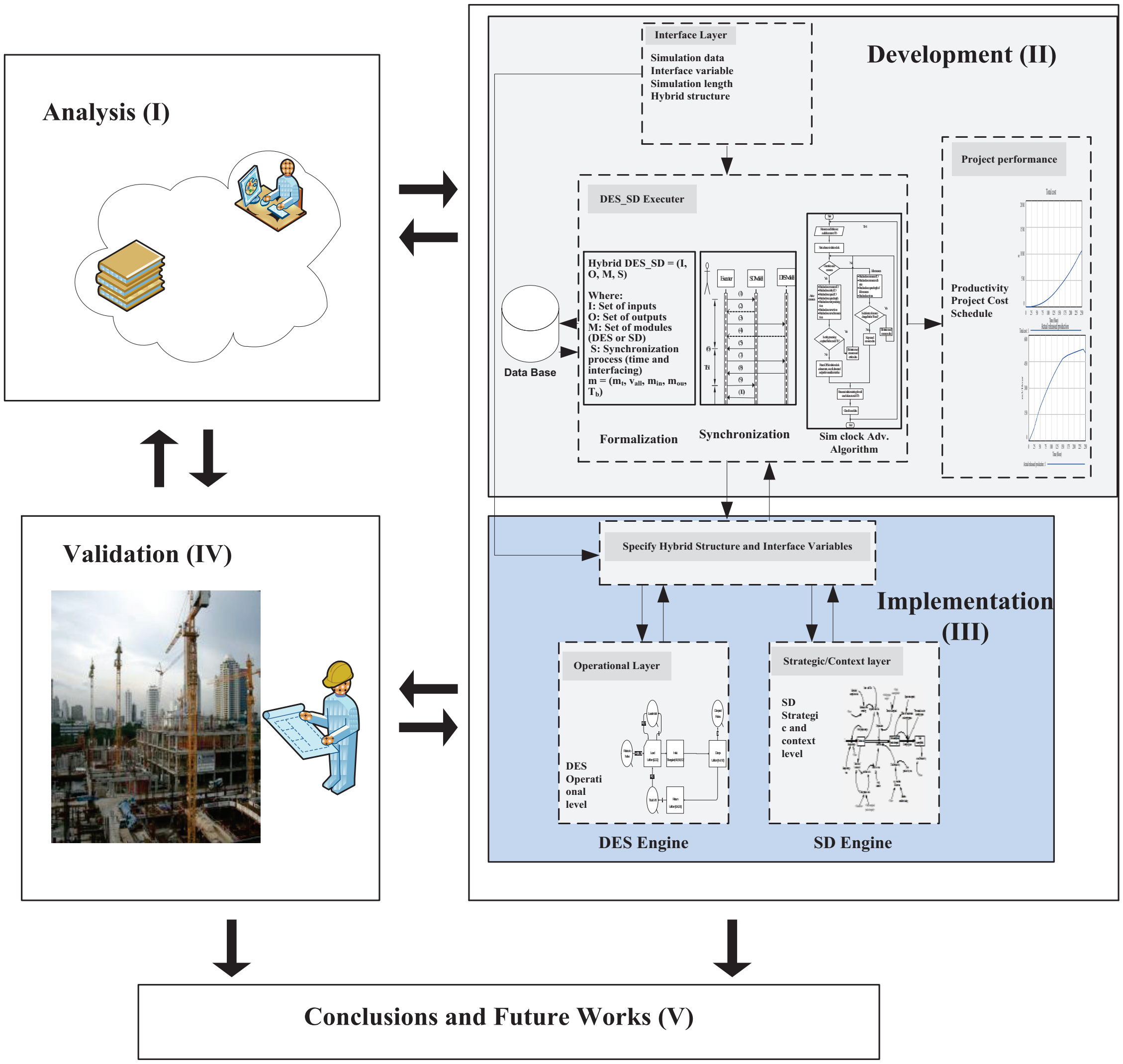


Figure 3.1 Roadmap to the Development of the Hybrid Simulation System

3.2 Analysis Phase (I): Literature Review

The literature review was conducted in Chapter 2. It covered a wide range of research in computer simulation theories and their applications in management. The review mainly focused on applications of simulation in construction management and simulation applications developed in other fields (e.g., enterprise, manufacturing, and software projects). The elaborated methodology components are demonstrated in Figure 3.2. *Firstly*, the literature review focused on the characteristics of the project management levels (SPM and OPM) and the decision levels. The next part discussed the characteristics of modeling elements in a construction project. This allowed identifying the influential elements that generate the operations real behavior. *Secondly*, simulation methods such as discrete, continuous, and system dynamics were presented. Each method was discussed and analyzed from the perspective of both philosophy and application. Procedures used to develop the models using the different methods were outlined. Most well-known applications of DES and SD were presented along with their shortcomings when applied to the construction management. The limitations associated with DES and SD methods have led to the next step of finding a solution to the research problem. *Thirdly*, based on the drawbacks of DES and SD, combining DES to SD (hybrid simulation) is believed to be a solution to the research problem. Then, hybrid simulation literature was investigated. The review extended to a variety of management areas in manufacturing and software projects. *Fourthly*, tools used to develop hybrid simulation models, such as formalism, distributed simulation, and synchronization, were investigated. Finally, limitations associated with current methods were summarized at the end of the review stage. The outcomes of the analysis phase (I) was presented in Chapter 1 and Chapter 2.

3.3 Development Phase (II): Hybrid Simulation System

This phase builds on the outcomes of the analysis phase (I). The development phase involves creating the hybrid simulation system. The *hybrid simulation system* in this research is precisely defined as "*The steps needed to develop a hybrid simulation model and a hybrid simulation application*." Therefore, the system's first component is concerned with developing a hybrid method required to create hybrid simulation models. The method must be capable of providing guidance, as well as rules to follow when creating a hybrid DES and SD model. The second component in the definition of the hybrid simulation system is the capability of the system to be used in developing hybrid simulation applications. Six major components pertaining to the hybrid simulation system were identified and illustrated in Figure 3.3. The *first* component of the hybrid system is identifying the boundary of the hybrid simulation model and the influential units. These two parameters represent the elements responsible for generating the real behavior of a project. The *second* component is setting criteria for classifying the influential units of a construction project. The criteria identify units of the project that will be modeled by either DES or SD methods. Failure to adequately select the appropriate simulation method thwarts the simulation model outputs. Criteria were developed from the main characteristics of the DES and SD methods. The *third* component is developing a hybrid simulation model that consists of DES and SD models. The models should be capable of capturing the purpose that they were developed to address. The norms of SD and DES were used to develop simulation models. The *fifth* component is developing formalism that focuses on how hybrid simulation model variables are sent, interfaced, and received. The *sixth* component is concerned with the synchronization mechanism that systematically integrates the DES and SD simulation times. It deals with the variables' state updates mechanism adopted by DES and SD methods. The *sixth and last* research component is developing the Executer that will be responsible for implementing the whole integration process through aggregating the mentioned components of the hybrid system on a single computational platform. It manages the simulation clocks and processes the data mapping among hybrid simulation models based on hybrid model design. These six elements are elaborated in the following subsections.

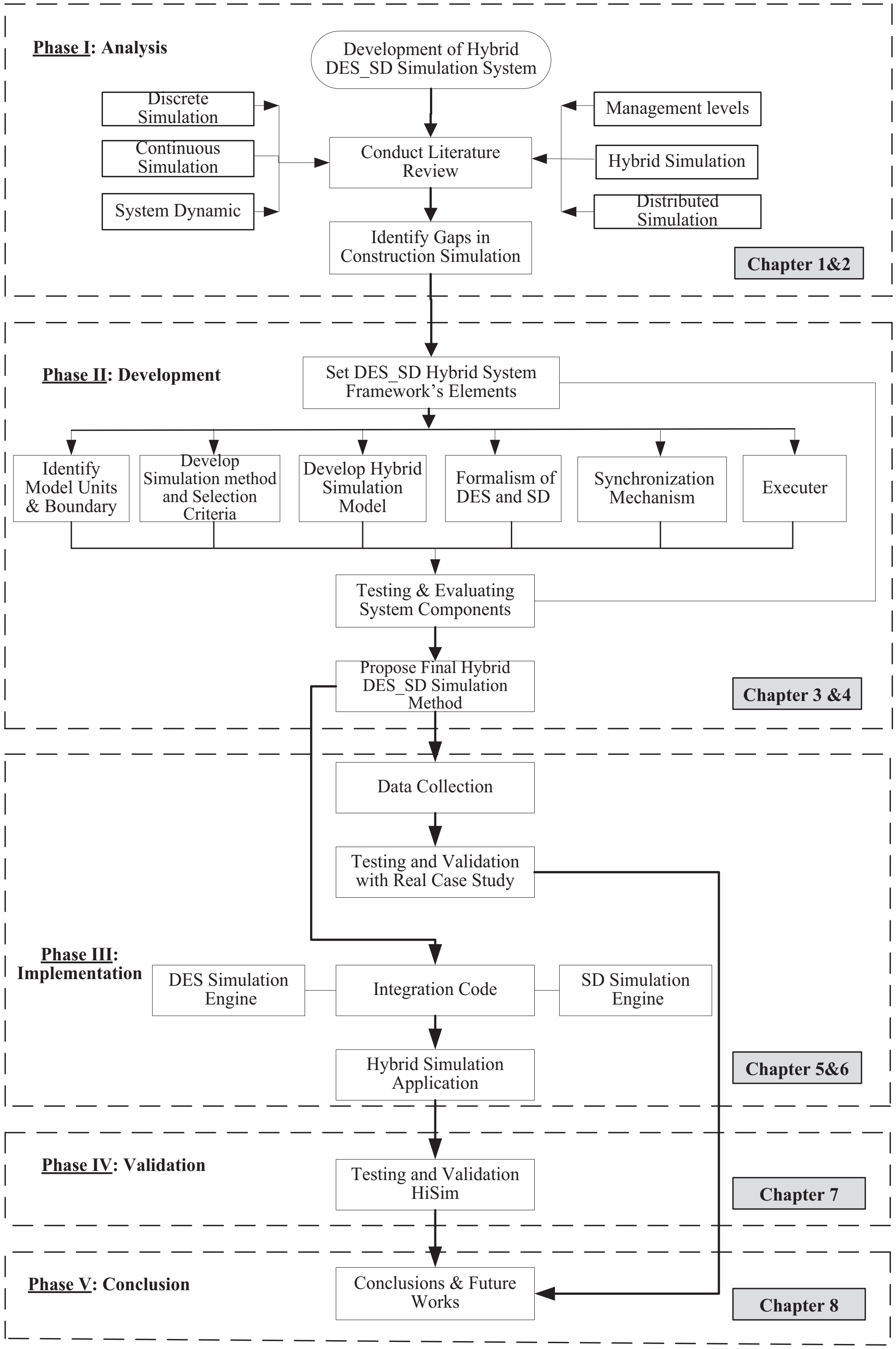


Figure 3.2 Flowchart of Research Methodology

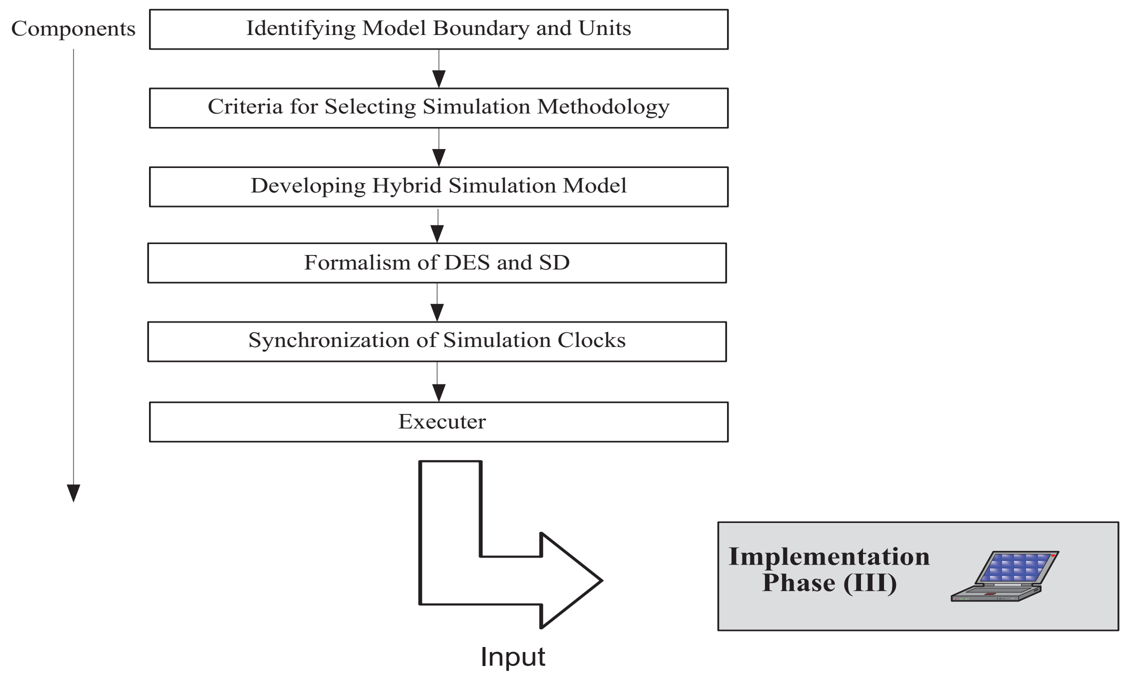


Figure 3.3 Hybrid Simulation System

3.3.1 Identifying Model Boundary and Units

Efforts to develop a hybrid simulation model are only justified if some parts of the problem are better modeled using DES and other parts are using SD. The manager should understand the problem in depth, identify the key variables and the time horizon, and understand the historical behavior of the variables. All this should be done within a specified objective and boundary for which the hybrid simulation modeling is deployed. Defining the boundary means specifying the influential variables that are responsible for the model's behavior. This can be done by decomposing the project into units. For instance, influential units that can be decomposed from the project system such as weather unit, workforce skill unit, and overtime unit. Every unit contains a certain number of variables such as precipitation rate, worker skill, and fatigue, respectively. It should be mentioned that this step is different from defining the boundary of the SD model, which involves classifying variables as indigenous, exogenous, and external. The results of this step are units (modules) that can be used to develop either DES or SD simulation modules. Figure 3.4 demonstrates an example of earthmoving project decomposition into units.

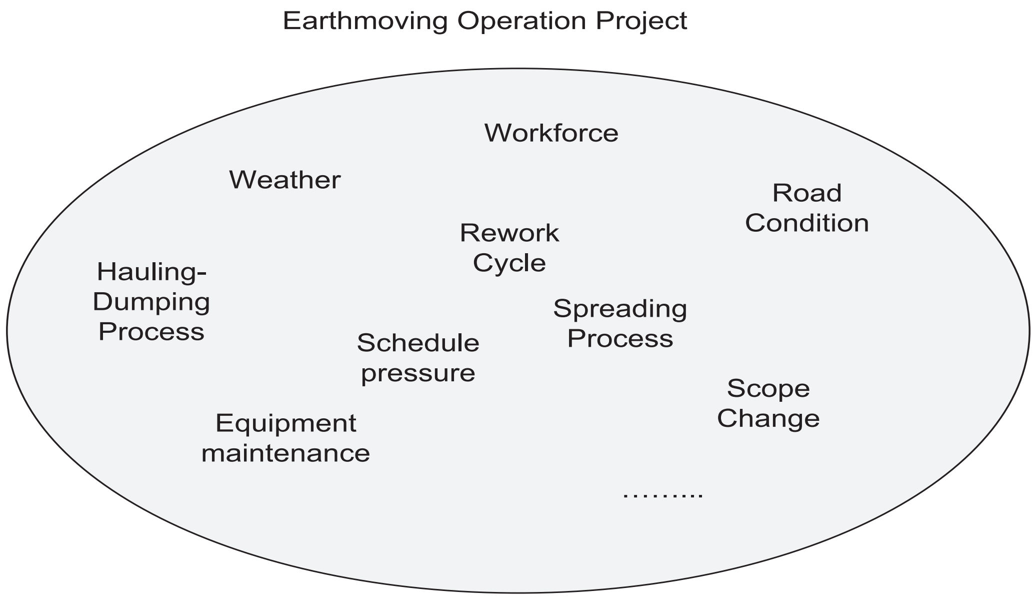


Figure 3.4 Decomposing of Earthmoving Project into Units

3.3.2 Criteria for Selecting Simulation Method

After decomposing construction projects into units, these units need to be classified based on a number of criteria extracted from the characteristics and philosophies of the two simulation methods (DES and SD). The criteria are required to assist in identifying and classifying units of the project to be modeled using the DES method and units to be modeled using the SD method. Selecting either one depends on the attributes and purposes of each unit. For instance, if a unit decomposed from the project has detailed data of daily activities, then this unit is best-modeled using the DES method. On the other hand, a unit with high abstraction that is characterized with few details and addresses the strategic level is best-modeled using the SD method. This is essential to recognize the environment within which the hybrid simulation objectives can best be attained. The process followed to decompose and select the appropriate simulation method is depicted in Figure 3.5.

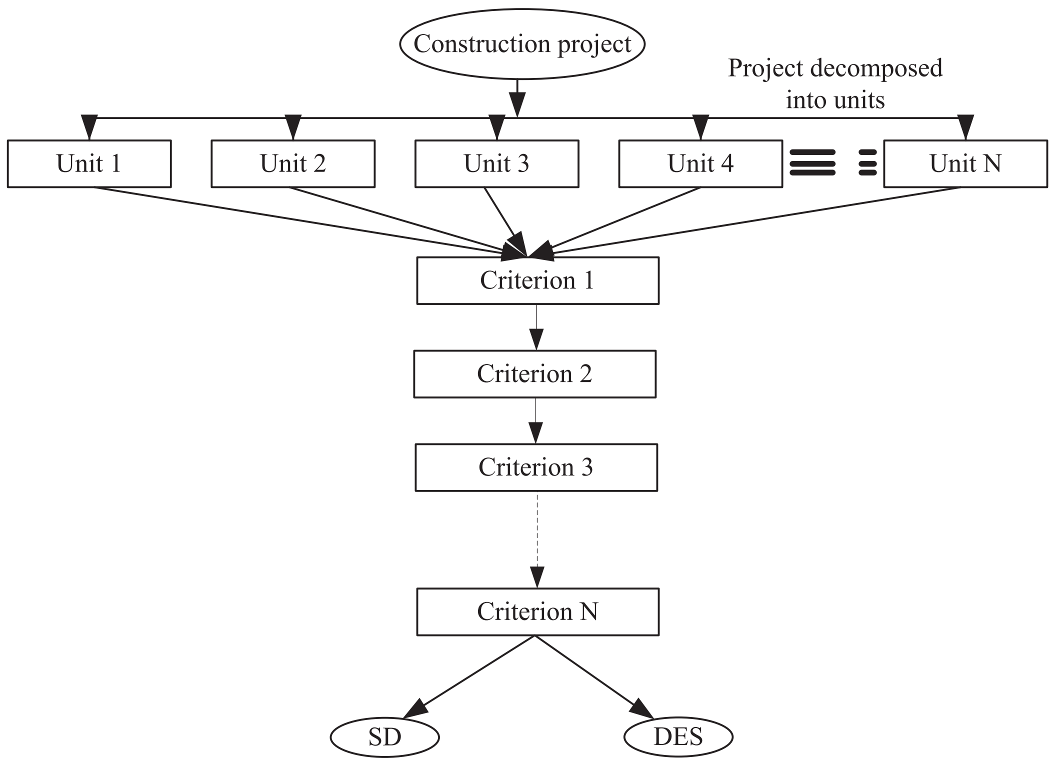


Figure 3.5 Project Decomposition to Select Suitable Simulation Method

3.3.3 Developing the Hybrid Simulation Model

The hybrid simulation model is composed of the DES and SD models. Developing these two models and preparing them for integration to create the hybrid simulation model requires certain arrangements that involve three elements: 1) developing DES and SD models; 2) selecting the hybrid simulation structure; and 3) defining the interface variables. In the following subsection, these three elements are discussed.

(i) Developing DES and SD Simulation Models

The norms used to develop DES and SD models are utilized. The DES models are developed using EZStrobe, emerged from the STROBOSCOPE simulation language. The modeling elements such as COMBI, NORM, and QUEUE are used to represent the variables pertaining to the operational level. SD models are developed using modeling concepts that are presented in Sterman (2000). Modeling elements such as causal-effect loops, Flows, Stocks, and Auxiliary variables are used to represent variables pertaining to strategic/context level of the project.

(ii) Hybrid Model Structure

The hybrid simulation structure is the process followed to organize the scattered DES and SD simulation models/modules into a hybrid simulation model network. In other words, it is a structured protocol responsible for creating the integration between the DES and SD models through interface variables. This involves classifying certain variables into data sender variables, interface variables, or data receiver variables. The researchers in the field of hybrid simulation used many structures, such as using the SD at the top level, where certain variables in the SD are updated from the DES model, or modeling the context of the DES model using SD and exporting values to variables in DES. Thus, the hybrid structure is more related to the nature of the problem being modeled, and it is this nature that dictates what type of hybrid structure should be selected.

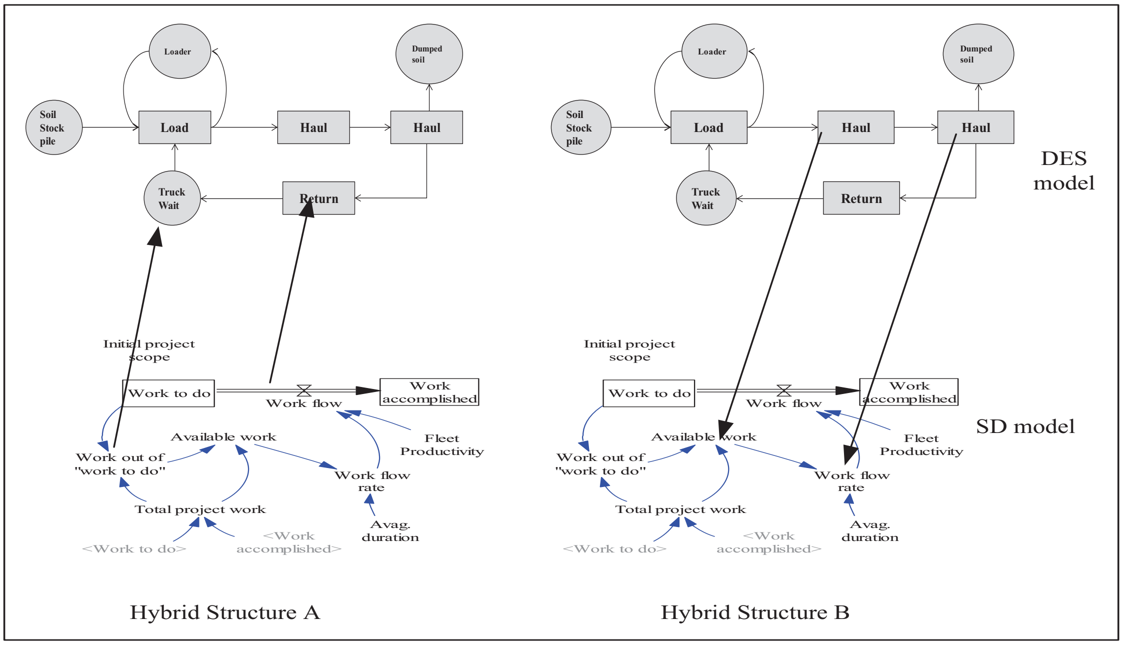


Figure 3.6 Sample of Hybrid Simulation Model Structures

(iii) Interfacing of DES and SD Models

Interfacing of simulation models means selecting variables in the DES and SD models to act as the contact points between these models. The interfacing process between the simulation models requires classifying the variables that are selected to participate in the interfacing process as sender variables, interface variables, or receiver variables. Sender and receiver variables send and receive values of variables, respectively, in both simulation models through interface variables. The selection of interface variables depends on the modeler's needs. To ensure consistency in data mapping, variables of certain characteristics should be mapped into variables of similar characteristics. For instance, if a variable has a measuring unit (unit/time), then it cannot be mapped into a variable that has measuring unit (unit). In the mapping process between sender-receiver variables, characteristics of the sender variables data must match characteristics of the receiver variables data.

3.3.4 Formalism of DES and SD Models

Formalism is a method to specify and describe the variables of the hybrid simulation model. This is needed to establish an organized mechanism of communication between the hybrid model variables. Interface variable values that need to be exported for use by other variables should be defined in such a way to make it easy for the Executer to recognize and implement the exportation. This is also true for importing variable values. For instance, an X variable in M model is needed by a Y variable in N model. In order for the Executer to accomplish this process, the sender variable X, the receiver variable Y, the interface variable Z, and the point in the simulation time at which this interaction will take place should be defined. The developed formalism consists of a set of models and synchronization functions. The model set defines the type of the model being processed, the sender variable, the receiver variable, and the interface variable, while the synchronization function deals with the simulation clocks of DES and SD models.

3.3.5 Time Synchronization Mechanism of Hybrid DES and SD Models

The hybrid simulation model is composed of DES and SD models. The two simulation methods update variable states and advance simulation clocks differently. DES models use discrete jumps based on the occurrence of events to update states. Thus, the DES method is an event oriented state updating method. In the SD method, variable states are updated based on the elapse of the time interval. The whole simulation length is divided into time intervals, and at the end of each time interval, variable states are updated. To deal with the different state updating mechanism, there is a need to: 1) present a concept and a background for new synchronization method; 2) solve the issue of advancing the DES and SD simulation clocks; and 3) develop the mechanism of time sequence messages between models and the Executer to implement the synchronization method. These three elements represent the pillars upon which the synchronization method is developed. The following subsection presents brief details about the three elements.

(i) Advancing Mechanism and Time Bucket

One of the challenging issues in developing a new synchronization method is establishing the theoretical background to start building the method. This requires upholding the argument presented in the research objective, which stated that the developed synchronization method must acknowledge the unique characteristics of each simulation method and not permit any method to dominate the other. The second important aspect is concerned with the ability of the synchronization method to allow sender variables in simulation models to prepare values when needed by the receiver variables. Through the investigation of available concepts used to develop synchronization methods in the past, the Time Bucket (TB) concept emerged as a potential candidate to develop the synchronization method. TB is time oriented and divides the simulation time into equal time intervals; at the end of each time interval, system states are updated. This is similar to the mechanism of advancing the simulation clock in SD models. Thus, as expected, the developed synchronization method did not face complications with the SD simulation clock. However, since the DES simulation clock advances in a different fashion, an algorithm had to be developed to control the DES simulation clock. This algorithm has to allow the DES simulation clock to be compatible with the developed synchronization method.

(ii) Discrete Simulation Clock-Advancing Algorithm

The SD method advances the simulation clock in a similar fashion to the developed synchronization method. Yet, this is not the case with discrete simulation. The DES method advances the simulation clock based on the occurrence of events, which is not compatible with the developed synchronization method. Therefore, a special arrangement was considered to address the compatibility issue between the simulation clocks. An algorithm that controls the DES simulation clock was developed. This algorithm provides the missing link between the DES simulation clock and the TB concept. It signals the Executer when to start advancing the simulation clock of the hybrid simulation model and when to halt. The advancing and halting is controlled by the algorithm and is based on the developed states updating mechanism.

(iii) Time Sequence Diagram

The implementation of the developed synchronization method is based on an organized sequence of messages exchanged between the hybrid simulation model (DES and SD models) and the component responsible for implementing the whole synchronization and integration process (Executer). Thus, sequencing messages that represent objects' behavior within a time period are necessary for the successful implementation of the hybrid simulation model. The elements of the developed simulation method must be implemented in a highly organized means to ensure its functionality as outlined.

3.3.6 Executer

The Executer plays the central role in integrating DES and SD models. One of the main tasks of the Executer is to implement the synchronization mechanism, and to control the communications overhead of the hybrid simulation model. The Executer interacts with the DES models developed using DES software and the SD models developed using SD software to facilitate the integration process based on the developed formalism and synchronization. The Executer is responsible for performing the following tasks.

1. Providing a user interface layer that allows inputs and outputs as required by the user.

2. Providing modules data import-export management. The selected interface variables that are designated to receive or share their values are specified in the Executer.
3. Implementing the synchronization method.

3.4 Implementation Phase (III):

This phase implement the developed hybrid simulation method using a real-world case study from construction domain. It involves three components. The first component is collecting data of a complex real construction project. The data collection process uses numerous published data concerning project information, reference models, and the influential factors (e.g., weather). The second component is developing the hybrid simulation model (DES and SD models) for the case study. DES models are developed to simulate operational aspects of the project, and are computed using EZStrobe software which supports DES. The SD model is developed to address the strategic/policy aspects of the project, and computed using the Vensim software package which supports continuous simulation. The DES and SD models are integrated using the developed components (e.g., synchronization) of the hybrid simulation system. Then, results of the hybrid simulation model are analyzed and discussed. The third and last component is developing an automated simulation tool (hybrid simulation application). This tool integrates DES and SD models developed using EZStrobe and Vensim, respectively. Thus, it should be mentioned that the implementation phase is executed in two stages. The first is a semi-automated process of developing hybrid simulation model from DES and SD models as described in Chapter 6, and the second is developing a fully automated simulation tool that integrates DES and SD models using the developed Executer, as illustrated in Chapter 7. This allows the validation and testing process to be addressed from different perspectives as is discussed in these two chapters. The architecture of the implementation process is demonstrated in Figure 3.7.

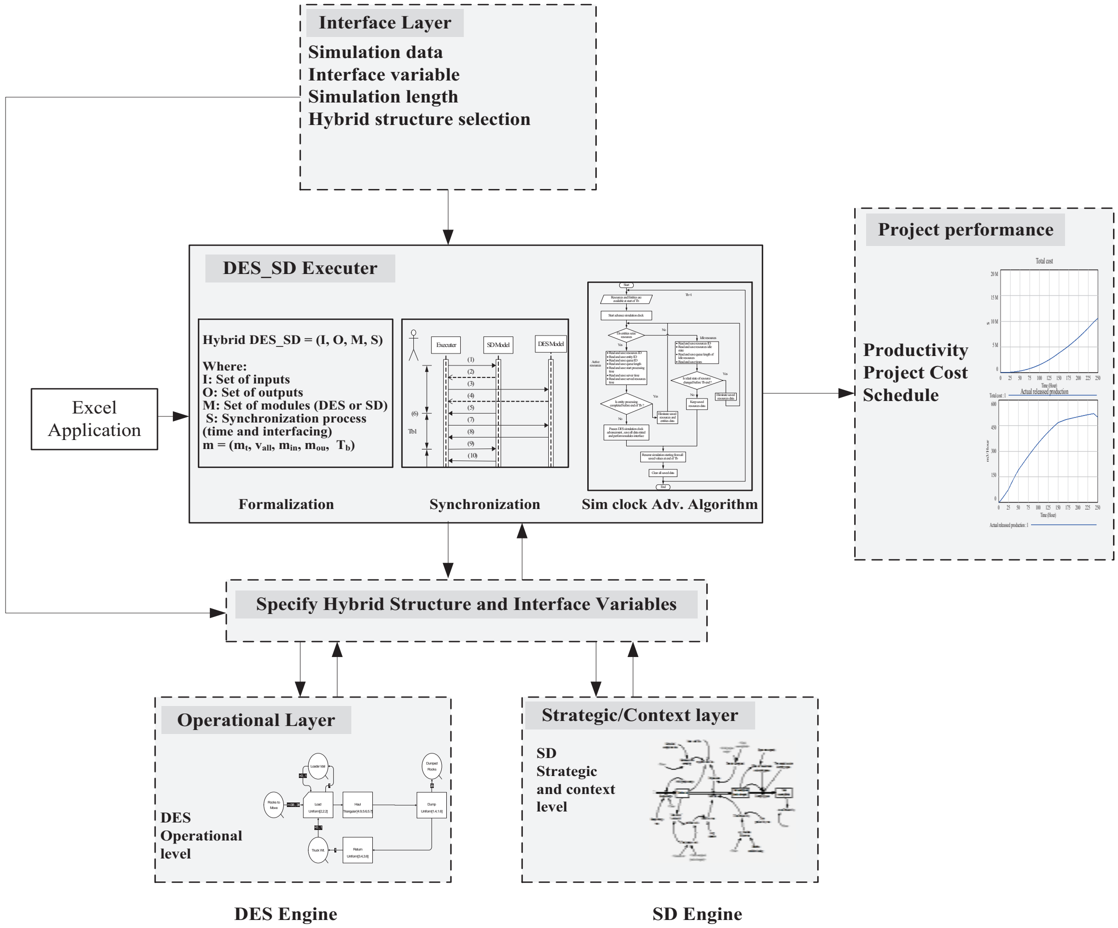


Figure 3.7 Hybrid Simulation Implementation Architecture

3.5 Validation Phase (IV): Testing and Validating the Hybrid Simulation Method

This phase is concerned with testing and validating the developed hybrid simulation method and application. Simulation is a computer-oriented research and is almost exclusively a computer-based process. Precision in simulation tools relies mainly on the implementation of the tool and on the accuracy of the data used to develop the simulation model. These two conditions of precision work in parallel and compromising on either leads to a poor simulation model. Validating the hybrid simulation method and its application is conducted at three levels: 1) the SD model level; 2) the synchronization of variables level; and 3) the comparison of results level. The first level is to test the developed SD model and ensure its robustness. Standard tests identified in Sterman (2000) for validating SD models are used to execute this level of validation. In addition, the built-in function in the Vensim software is used to check the SD model units and equations. The second level of validation is to ensure that the synchronization of variables is carried out as per the designed mechanism and protocol. The variable values imported from either model are traced and monitored to guarantee precision. The third and last validation is comparing the results of the hybrid simulation model with the actual project data at the time of implementation.

3.6 Phase (V): Conclusions, Recommendations and Future Work

In this phase, the concluding remarks on the hybrid simulation method are stated. A detailed discussion of the challenges found and lessons learned are presented. In addition, a summary of the findings and highlights of potential topics for future research are outlined. Finally, the phase is concluded with the limitations associated with the developed method and possible future enhancements.

3.7 Summary

This chapter has presented the research methodology followed to achieve the objective of this research. The methodology encompassed five stages, namely: 1) analysis; 2) development; 3) implementation; 4) validation; and 5) conclusion. Highlights of the elements of each phase considered to address the research problem were presented. In the analysis phase, an exhaustive study of the current state of research was conducted, from which the research problem and objective were identified. The second phase involves developing the six components of the hybrid simulation system that are necessary to achieve the integration between DES and SD models. The third phase involves the implementation of the developed hybrid system on a real-world project from the construction domain through both a semi-automated approach and a fully automated approach. The last phase involves presenting the findings and lessons learned from the research.

CHAPTER 4

DEVELOPMENT OF HYBRID SIMULATION SYSTEM

4.1 Chapter Overview

This chapter presents the developed elements of the hybrid simulation system. Six elements have been identified to represent the fundamental base upon which the hybrid simulation is built. These elements are: 1) identifying the model's boundary and units; 2) developing criteria to select the simulation method; 3) developing the hybrid simulation model; 4) formalism of DES and SD models; 5) synchronizing the DES and SD simulation clocks; and 6) Executer.

4.2 Hybrid Simulation System

The developed hybrid simulation system consists of the elements that are utilized to develop hybrid simulation models and applications as shown in Figure 4.1. In the following subsections, these elements are discussed in detail.

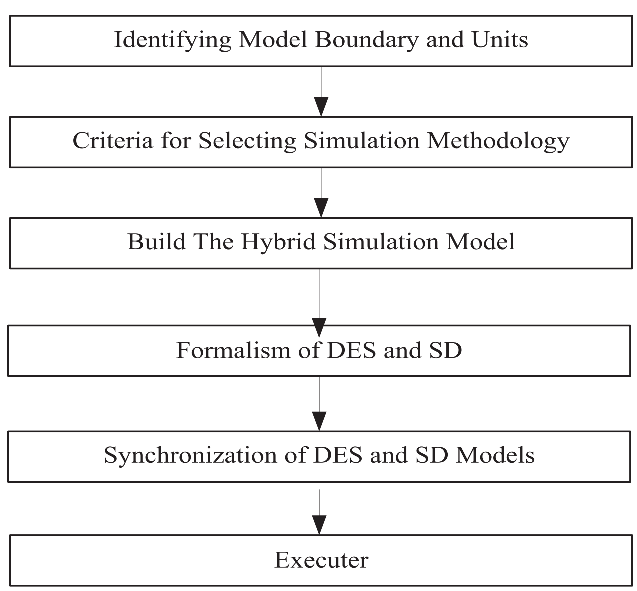


Figure 4.1 Elements of the Hybrid Simulation System

4.2.1 Identification of Model Boundary and Units

Defining the boundary of the hybrid model is the starting point for developing a hybrid simulation model. The boundary specifies the variables that should be included and excluded from the model structure. This is an indispensable aspect as it defines the hybrid model scope and eases the modeling process. The model behavior is generated only within this boundary. Traditionally, in a unique simulation model (discrete or continuous), the model is perceived as a collection of certain class of variables that represents part of a project or whole operation (e.g., loading operation). Each class forms a unit that is capable of being utilized to develop simulation model/module. Thus, the second step, after defining the hybrid simulation model, is to decompose the whole project into smaller units that represent a specific parameter, e.g., scope change, skills level, and weather. For instance, earthmoving operations can be decomposed into many units from the perspective of the operational level, e.g., loading, hauling, dumping, and excavating. On the other hand, units can be decomposed to represent the strategic/context level, e.g., productivity, weather, scope change, and rework-cycle. All units must address a certain purpose and be within the boundary that defines the model scope. Each unit, based on its complexity, should be representative so simulation model/module could be developed from it. These models can be either DES or SD. Decomposition can be also performed based on the nature of the variables, which could be discrete or continuous. Figure 4.2 demonstrates the steps involved in defining the model boundary, and the decomposition of the project into units.

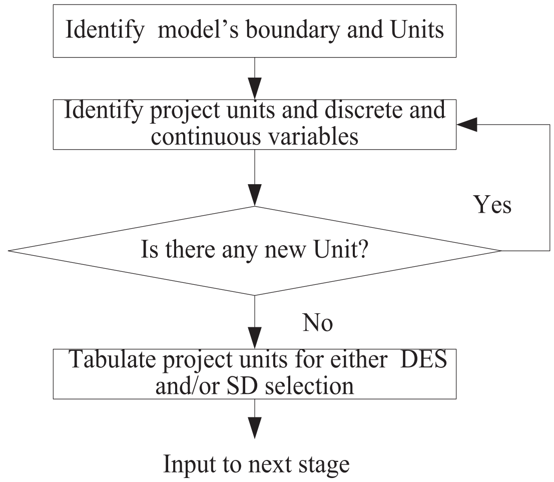


Figure 4.2 Model Objectives and Units Identification

4.2.2 Criteria for Selecting Simulation Methodology

The whole construction project is decomposed into units as discussed earlier and as shown in Figure 4.3. These units are tabulated and ready to be converted into simulation models. However, it is necessary to select an appropriate simulation method that corresponds to these units. Thereafter, units need to be tested against criteria that emerge from their characteristics and from the philosophies of the simulation methods, and then to select the appropriate simulation method. In the literature review discussed in Chapter 2, the outlined characteristics of the DES and SD simulation methods were adopted as a basis for developing the decision criteria to select the simulation method. Construction project units are assessed against the following criteria to decide which simulation method to use in the modeling process.

- **Problem Scope and Focus: Operational or Strategic**

  This refers to the scope and focus of the unit. If units have detailed data, focus on daily operation activities and are complex, then they are consistent to DES. Units with fewer details, addressing the strategic level, and which are highly abstract are more consistent with SD.

- **Prospective: Reductionism or Holistic**

  Prospective refers to the structure overview. Units that promote individuality with reductionism characteristics are consistent with DES. Units that are concerned with global, homogeneous, and holistic views are consistent with SD.

- **Data Nature: Quantitative or Qualitative**

  This describes the nature of the data. Units that are characterized with quantitative data are modeled using DES. On the other hand, units characterized with qualitative data are consistent with SD.

- **State Change: Discrete or Continuous**

  This refers to the type of state change. Units that tend to update their states discretely and are of event-oriented nature are more consistent with DES. Project units that are characterized by continuous change over time and update their states at the end of time intervals are consistent with SD.

- **Level of Details: Narrow or Broad**

  This refers to the available information about the units being modeled. Units of narrow, numerous and focused information are more consistent with DES, and generally such systems are stochastic. Units of broad information with a high level of abstraction are consistent with SD.

- **Level of Model Complexity: Increase Exponentially or Linearly**

  If the complexity of unit tends to increase exponentially, then it is more consistent with DES. When the complexity tends to increase linearly, then it is more consistent with SD.

The work in this stage is similar to the work carried out in developing the Work Breakdown structure (WBS), where the project is decomposed into work packages at the lowest level to enter the scheduling process. Similarly, the project is decomposed into units to enter the simulation stage.

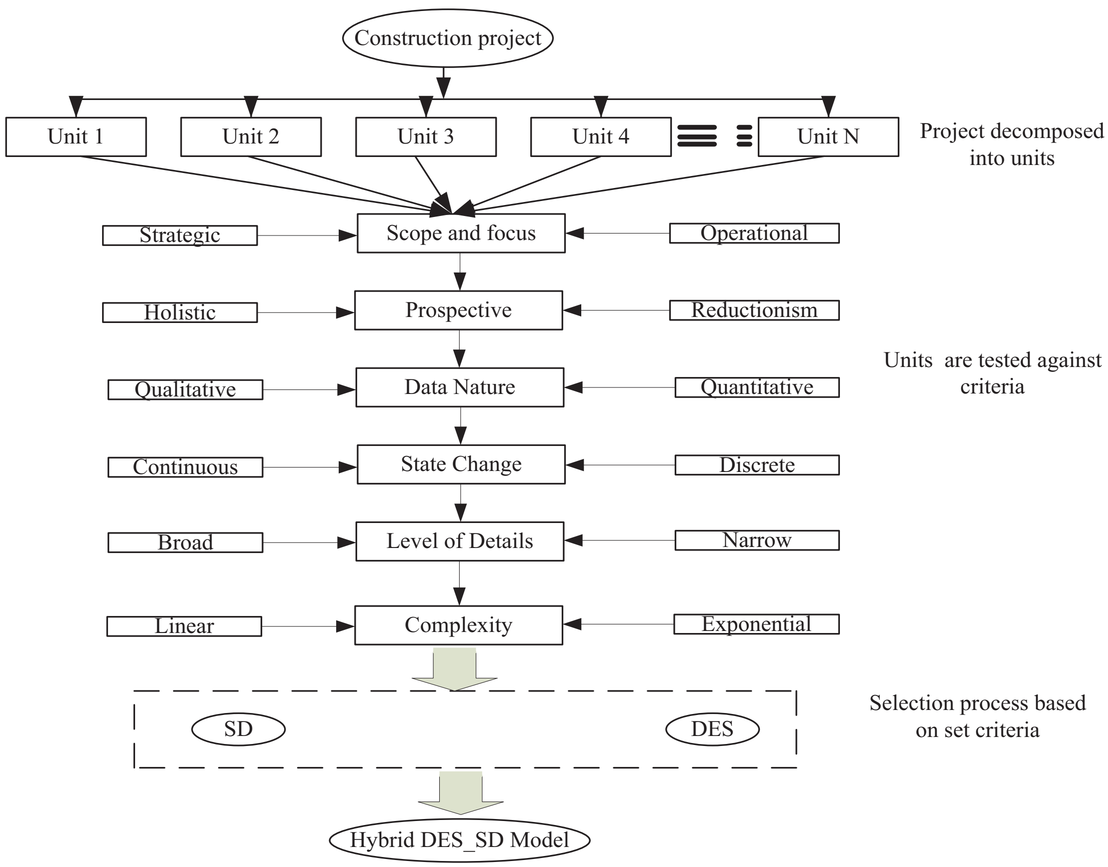


Figure 4.3 Criteria for Selecting Simulation Method

4.2.3 Developing the Hybrid Simulation Model

This component of the hybrid simulation system addresses three issues. The first is building the DES and SD simulation models. The second issue deals with the hybrid simulation structure that governs the data mapping process among simulation models. As stated earlier, the hybrid simulation model consists of a number of models developed using

DES and SD methods, and in order to integrate and interface these models, a predefined structure arrangement of the different simulation models in hybrid environment should be followed. The third and last issue is the interfacing process, as it provides the rationale behind selecting which variables in model X can interface with variables in model Y without violating the consistency of variables' data. The following sections provide detailed explanations.

(i) Developing DES and SD Simulation Models

The DES models of the hybrid model are developed using the norms used to model DES models in the STROBOSCOPE simulation environment, while SD models are developed using causal-effect loops, stocks, and flows that were outlined in Sterman (2000). Both simulation methods were described in Chapter 2. Developing the DES and SD models, selecting a hybrid structure, and selecting the interface points are demonstrated in and explained in the following sections. The steps needed to develop the simulations models until the point of formalism are shown in Figure 4.4.

(ii) Hybrid Simulation Model Structure

The structure of the hybrid simulation model is defined as "*organizing the simulation modules/models (DES and SD) that compose the hybrid simulation model based on the required data mapping profile among modules/models to fulfills the purpose of deploying the hybrid simulation technique*." The structure of the data communication channels among variables of the simulation models is mainly specified by the selected hybrid structure. This definition takes us back to the question of how the hybrid simulation would be used to enhance the real representation of construction operations. The answer to this question is found in the hybrid simulation structure that should be adopted for solving the problem addressed by simulation. For instance, if the DES model is developed for the tactical level of construction operation and the effects of the surrounding environment factors (context) need to be accounted for, then SD modeling is deployed to model the interactions of those factors through using causal-effect loops and inject their effects into the appropriate interface point in the DES model. Other possible ways of a hybrid model structure can be in a format where tactical level parameters are permitted to interact based on strategic decisions. Thus, as a conclusion, the problem requiring hybrid simulation dictates the format of the hybrid simulation structure that should be selected. It is essential to determine in advance the structure that best solves the problem without hindering the model's outcomes.

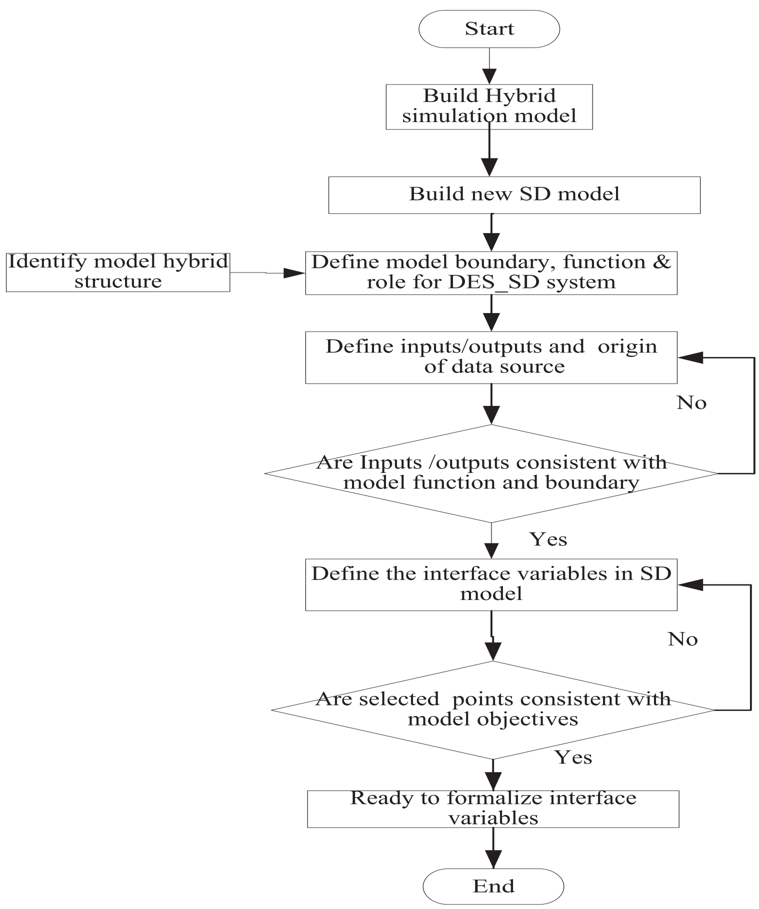


Figure 4.4 Developing DES and SD Models for the Hybrid Simulation Model

The process of simulation models interactions based on the selected hybrid structure can be better explained with an example. For instance, the management of earthmoving operations project needs to maintain a production rate of hauling fleet around 500 ton/h. This is clearly a continuous variable designation; therefore, it is modeled using the SD modeling method. Now, suppose the production rate drops below the permitted limit set earlier by the management at planning stage (500 ton/h). A reduction in project productivity triggers controlling measures to be considered by management in order to boost the production rate within the set target. An example of the measures taken can be increasing the number of trucks in order to finish the project on time. This corrective action is classified as a discrete change in a continuous variable's state. Subsequently, increasing the number of trucks will increase the production rate but may cause a change in other continuous variables (skills, learning curve, etc.). Lastly, the change in the continuous variable due to the corrective measure causes a change in the function governing production rate. Considering such interactions between discrete and continuous variables will inevitably enhance the modeling process and results in a better decision supporting system. Researches in different hybrid simulation fields have developed hybrid simulation models in different ways as shown in Table 2.5. These model structures are adapted to be applied in construction and take the following three forms.

a. Operation-Context Structure: In this structure, DES is used to model the operational aspects while SD is deployed to represent the context of the operation being modeled. The influence of the context variables on the operation is quantified using SD, and then interfaced with DES variables to reflect the real surrounding environment. The rationale behind this structure is that operations are greatly influenced by the context factors; hence, modeling

DES without accounting for these factors will hinder the model results. Information flows from the SD model to the DES model is demonstrated in Figure 4.5 (A). Data of stocks are mapped into QUEUES[1] and COMBI, and data of flows are mapped into NORM.

b. Strategic-Operation Structure: In this structure, the SD model is used at the higher level of the decision-making process (strategic level) to model policies adopted to manage the project. Nevertheless, the SD model fails to compute the variables at the operational level, which is one of the main limitations of the SD modeling method. Therefore, DES models are used to work as small gears in the global SD model to supply the operational data needed by management policies and strategic objectives. The SD model updates states of certain variables from the DES model as depicted in Figure 4.5 (B). Data of QUEUES, COMBI and NORM are mapped into stocks and flows. This structure type is used when the strategic level of the project has strong interactions with the operational parameters. Furthermore, this structure is useful when strategic/context can't perform well if not provided with feedbacks from the operational level. In this structure, DES models act as an auxiliary to the SD model and data mapping is from variables in the DES model to variables in the SD model.

[1] Mapping process among variables is discussed in section 4.2.3 (iii)

c. **Strategic Framework-Operation Structure:** In this structure, the SD model is developed to represent the strategic level of the project. The SD model will provide the boundary for the interactions of the DES model elements (e.g., resources and entities). Generally, in a typical DES model, elements of the model are allowed to interact among each other without an influence of the boundary or constraints set by decision makers at the top level. Such a mechanism allows many approximations. Allowing free interactions and approximations will eventually result in models that fail to represent the real situation. This problem can be overcome by using SD modeling to provide the framework for interactions of the DES model elements, as shown in Figure 4.5 (C). For instance, in earthmoving operations, the SD model will provide the perceived production rate through the project life cycle, project duration, and skills level, and the DES model elements are expected to interact under the umbrella of the defined frame of the SD model.

Consequently, after deciding on which hybrid structure to follow, it is imperative to select the variables that should be directly involved in the synchronization process. These variables provide the common simulation platform environment for the DES and SD model integration. The interactions among simulation models are executed through these points only. In the following section, more details are provided.

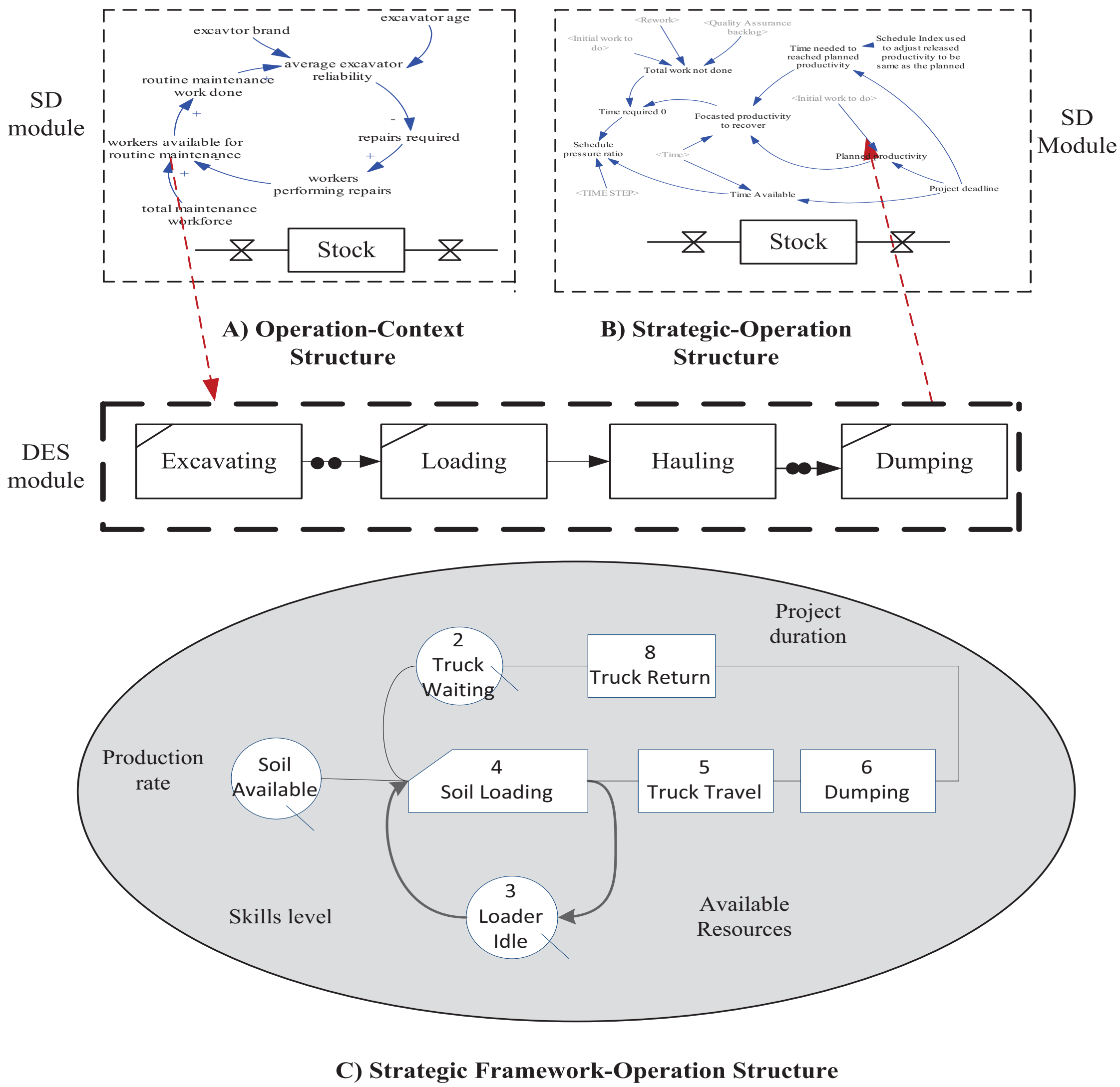


Figure 4.5 Mapping and Interfacing between DES and SD Modules

(iii) Interfacing DES and SD Models

The interface points between DES and SD models are variables, which actively participate in data mapping. These variables are defined as variables of a model whose values are modified or replaced by the influence of other variables in other models. In a hybrid simulation environment, the variables participating in the integration process of simulation models are classified into three types. The first is called a *sender variable*, which sends state updates. The second is called a *receiver variable*, which receives the state updates from the sender. The third is the *interface variable* or *conveyor*, which acts as a point of contact between the sender and the receiver. The interface variable's role is to receive state updates from the sender variable and then sends state updates to the receiver variable. The modification and replacement of values occur during running of the simulation model. Interface variables are selected based on the requirements of the modeler and the needs of the case under investigation. No clear guidelines can be set for selecting the interface variables, as every model's goal is unique.

Hybrid model boundary and function should be defined clearly before interfacing the variables. This assists in selecting which hybrid structure to use in the modeling process. Variables of DES and SD models are mapped as shown in Table 4.1. The DES model generates two types of observational data, queue length (QUEUES) and the random variant (COMBI and NORM) (Halpine 1992). On the other hand, the SD model generates three types of data computed by stocks, flows and feedback loops (Sterman 2000). Stocks are an accumulation of entities or material resulting from policies and strategies (flows). Assume the simulation clock stops at any point during the simulation run, rationally; the stocks preserve the values that were recorded just before stopping the simulation clock. This occurs because stocks represent materials that physically exist when time stops, while flows that represent the effects of policies disappear. Flows are a function of time, and are expressed as rates (units/time).

Table 4.1 Comparison between DES and SD Modeling Elements (Han et al. 2005)

Item	DES	SD	Measuring Unit
Modeling elements that carry observational data	Queue	Stocks	Unit
Modeling elements that carry time oriented data	COMBI and NORM	Flows	Unit/Time

As a result, Queues data in the DES model have the same characteristics as those observed in SD stocks. Hence, Queues are mapped into stocks and vice versa. The COMBI and NORM data in the DES model have characteristics similar to those of flows in the SD model. Hence, COMBI and NORM are mapped into the flows and vice versa. The mapping process must be consistent (data of X characteristics are only mapped into counterparts of data holding only X characteristics).

4.2.4 Formalism of DES and SD models

Simulation modeling is not accomplished by directly writing out a dynamic system structure, but indirectly, by using *system specification formalism*. System specification formalism is a shorthand means of specifying a system (Ziegler, 2000). In this research, formalism is the method used to describe the hybrid simulation modules to the Executer. Also, it facilitates the integration and the interfacing of the modules involved in the hybrid simulation. To describe hybrid simulation modules, formalism needs to specify the following properties of the module:

1. Module type: whether it is DES or SD.
2. Input variable (receiver) to the module and source of the input variable.
3. Output variable (sender) generated from the module.
4. Synchronization function that describes the simulation time management and the interfacing variables.

Analyzing the aforementioned four requirements to develop formalism that serves the hybrid simulation computational platform, the results reveal that properties 1, 2, and 3 of the module mentioned above are data sets and can be represented using set theory. Property 4 is the function that describes the state and point in simulation time when interfacing of variables takes place. This function can be obtained through developing a time management algorithm. The synchronization function should deal with two important aspects. The first aspect describes the simulation time advancing mechanism, and the second aspect describes the variables interfacing process. Based on the above analysis, Equation (4.1) represents the generic description form of the developed formalism

$$Hybrid\ DES\_SD\ module = (I, O, M, S) \qquad (4.1)$$

Where:

- I: Set of inputs
- O: Set of outputs
- M: Set of modules (DES or SD)
- S: Synchronization process (time and interfacing)

The generic Equation (4.1) provides a base for deriving Equation (4.2) to represent either DES or SD modules to the simulation Executer. The proposed simulation Executer requires three sets and two functions for describing DES or SD modules (m), as shown in Equation (4.2).

$$m = (m_t, v_{all}, m_{in}, m_{ou}, T_b) \qquad (4.2)$$

Where:

- m_t: Module type (SD or DES)
- v_{all}: Set of all interface variables in the module
- m_{in}: Set of module input variables (receiver), described by Equation (4.3).
- m_{ou}: Set of module output variables (sender), described by Equation (4.4).
- T_b: Time point where the interfacing of variables occur

$m_{in} = \{(m, v_i, m_s, op_{ms}, m_d) / m \in M, v_i \in V_M, m_s \in M, op_{ms} \in outports_{all}, m_d \subset v_{all}\}$
(4.3)

Where:

- v_i : Input variables to module m
- m_s : The module source from which input variables are imported
- op_{ms} : The output port in m_s from which the input variables are imported to m module
- m_d : Describes the variables in *m* that need to use the input variables from m_s. (m_d describes the variable input v_i *input port* (*ipm*) in *m,* the variables in *m* need to use v_i and the time point in the simulation clock where the interfacing of variables occur)
- *M* : Describes the simulation model (DES and SD) in the hybrid environment (hybrid model)
- V_M : All variables in model *M*
- $outports_{all}$: Set of all output ports in *m*
- v_{all} : All variables in *m*

$m_{ou} = \{(m, op, ov,) / m \in M, op \in outport_m, ov \in v_{all}\}$ (4.4)

Where:

- *op*: Output port in *m*
- *ov*: Output variable given through *op*

Figure 4.6 demonstrates a graphical representation of the developed formalism. Two types of modules are shown. The first is the *source* module that contributes the variable (sender) needed by the module in the hybrid model through *outputports* (op_{ms}). The second is the *receiver* module that receives the variable contributed by the *source* module. The *source* module delivers the variable through *outputports* (op_{ms}) (interface variable) while the *receiver* module receives the variable through *inputport* (inp_m). V_s is the variable defined in m_s and needs to be used by module *m* variables and input through *inputport* (inp_m) as V_i.

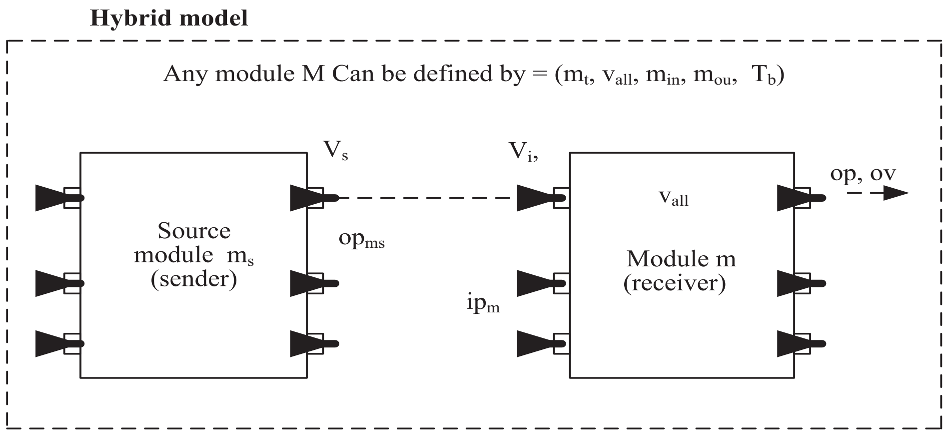


Figure 4.6 Graphical Representation of DES_SD Formalism

4.2.5 Simulation Time Synchronization Mechanism[2]

Synchronizing DES and SD models simply means integrating the models by providing the appropriate platform for advancing the simulation clocks, and facilitates the interfacing of variables. Synchronization deals with designing a communication protocol that manages the time-points in the simulation time where variables of the hybrid model are interfaced. The need for the synchronization mechanism arises from the fact that DES and SD methods update system states differently. The simulation clock in the DES method advances at the occurrence of events and subsequently system states are updated. This means that events in the model occur at unequal time points due to the random nature of events in

[2] The synchronization of DES and SD models was published and presented in winter Simulation Conference, Berlin, 2012.

DES models. In the SD method, simulation time is divided into equal time intervals. The system states are updated at the end of each time interval. This mechanism of state updating continues until the end of the simulation time. Another interesting point is that SD models are continuous systems and incapable of generating events like DES models. The concerns of synchronizing DES and SD models arise from the fact that the simulation clocks of both models need a creative time synchronization method that is capable of dealing with the two distinct simulation clock updating mechanisms. Available time synchronization methods, such as the conservative and optimistic methods discussed in Chapter 2, were developed to support distributed or parallel DES models on diverse platforms. Thus, in order to deal with the discussed challenges, the method developed in this research consists of three elements: 1) the advancing mechanism; 2) the DES-clock advancing algorithm; and 3) the message sequence mechanism. The following subsections provide detailed explanations of these three elements.

(i) Advancing Mechanism

A concept of simulation clock synchronization called *Time Bucket Synchronization,* proposed by Steinman (1990) and enhanced by Fujii et al. (1994) to synchronize DES simulation models, is investigated as a possible advancing mechanism. It is found that this concept is a potential candidate to be adapted and used for developing the synchronization of the DES and SD methods. The developed synchronization method works in the following sequence. The length L of the hybrid simulation time is divided into equal time intervals called *Time Buckets* (Tb_s), and at the end of these time intervals, the interfacing and interaction of simulation models take place. T_b should be small enough to capture any significant change in system state and large enough to discard unnecessary computations. The rationale behind using the *Time Bucket* is that SD updates state at equal and known time intervals while DES updates state at the occurrence of events. These events generally occur at unequal time points, difficult to predict. Therefore, it is simpler to trace model states at stipulated time points (which are guaranteed by Tb), rather than tracing model states based on the occurrence of events, where the time point of occurrence is not known in advance. *Time Bucket* ensures that every simulation method (DES or SD) will preserve its unique characteristics during the simulation computation process, and this is one of the main objectives of this research. In the developed synchronization method, the simulation time of length (L) is divided into small equal time intervals called Tbs. The time bucket size is equal to the SD model STEP TIME as shown in Equation (4.5) and Figure 4.7.

$$Tb \text{ (hybrid model)} = SD \text{ model } STEP\ TIME \qquad (4.5)$$

At the start of the simulation time of length L, the hybrid simulation system initializes the simulation clocks of the DES and SD engines as well as variables in the models. Now, the Executer is positioned to advance the simulation clock of the hybrid simulation model to Tb_1, where,

$$Tb = Tb1 = Tb2 = \ldots.\ T\,bn \qquad (4.6)$$

At the end of Tb_1, the interface of the variables and data exchange occurs between DES and SD models as shown in Figure 4.7. This generates a new system state called the hybrid DES_SD state. The data flow direction is based on the selected hybrid model structure that was discussed earlier in this chapter and through only the selected interface variables. Variables of the simulation models that are not selected to participate in the interfacing process and the data exchange will not participate in the mapping process. However, the effects of the updated variables are propagated to influence those exempt variables. When the process of interfacing and data exchange is completed, the hybrid simulation clock is advanced toTb$_2$, and again at the end of Tb$_2$, interfacing of variables takes place. This process continues on this mechanism until the simulation clock time reaches the end of simulation time, length L.

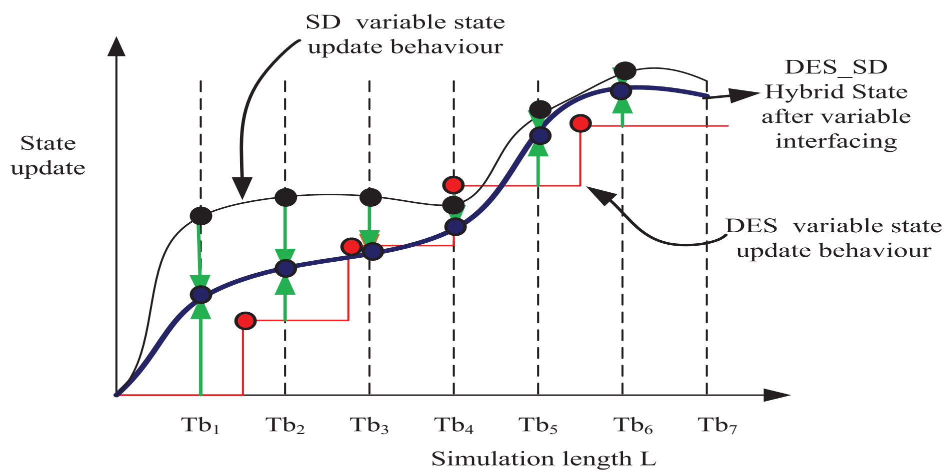


Figure 4.7 Synchronizing DES and SD Models using Time Bucket (Alzraiee et al. 2012c)

Although the *Time Bucket* synchronization method is simple and easy to implement, there are three main drawbacks inherent to this method. *Firstly*, it faces difficulties to capture DES events of zero duration. *Secondly*, the size of the time bucket needs to be carefully selected, otherwise results will be misleading. The Tb size must be large enough so that models have a low synchronization overhead and small enough to capture any significant change in the hybrid model states. *Thirdly*, any two consecutive events occurring in the DES model that have a combined total event time less than the Tb size will cause the first event not to be captured by the hybrid model. This phenomenon happens because the initialization and occurrence of the first event takes place before the stipulated time for the interfacing of DES and SD variables. Therefore, when selecting Tb size, it should be ensured that no two consecutive events can occur within the Tb interval.

The developed time synchronization method is time oriented. Therefore, it is expected not to face any complications with the SD method. However, for the DES method, an algorithm (DES advancing algorithm) that breaks the simulation length L into time buckets to facilitate the integration is needed. This makes the DES simulation clock compatible with the proposed synchronization method. Unless the DES models are capable of preparing variable at the tactical level needed by SD model, at the point where interfacing of the simulation models occur, SD model will not be able to update its variables from the project's operational level. Hence, results of the hybrid model are likely to be doubtful. Therefore, it is essential from the modeler to pay attention to the Tb size and the event expected occurrence time.

(ii) DES Simulation Clock Advancing Algorithm

The second component of the developed synchronization method is to make the DES simulation advancing mechanism compatible with the synchronization process. An algorithm that divides the DES simulation length (L) into intervals, facilitates integration, and resumes the simulation is needed. The developed algorithm is depicted in Figure 4.8. Initially, for the DES engine to start advancing the simulation time, conditions such as the required resources and entities should be available at the start of Tb_1. Now, the simulation is in position to start advancing at the beginning of Tb_1. If entities seize the required resources, then all data of active resources and entities in the simulation model are read and saved. Otherwise, idle resources data are read and saved. If the process involving the active resources has not finished processing the entity at the end of Tb_1, then the DES simulation clock is paused, and all data is saved and the interfacing of DES and SD models is performed. Otherwise, saved resources and entity data are eliminated and there is a return to re-allocate the next process and its entities, attributes and resources. In the DES model, events having an occurrence time less than the Tb_1 finish their processes before the interfacing of variables can take place. Hence, their data are not captured in the next scheduled interfacing, but their effects are propagated to a second event.

After the interfacing is accomplished, all saved data at the end of Tb_1 of active or idle resources are used by the DES engine for the next round of computations that begins by the commencement of Tb_2, and continues in the same sequence explained for Tb_1. The time point between the end of Tb_1 and the start of Tb_2 is the point where the simulation clock resumes the progression of the simulation. The algorithm continues until the model reaches the initially set simulation time L, and then terminates the simulation run.

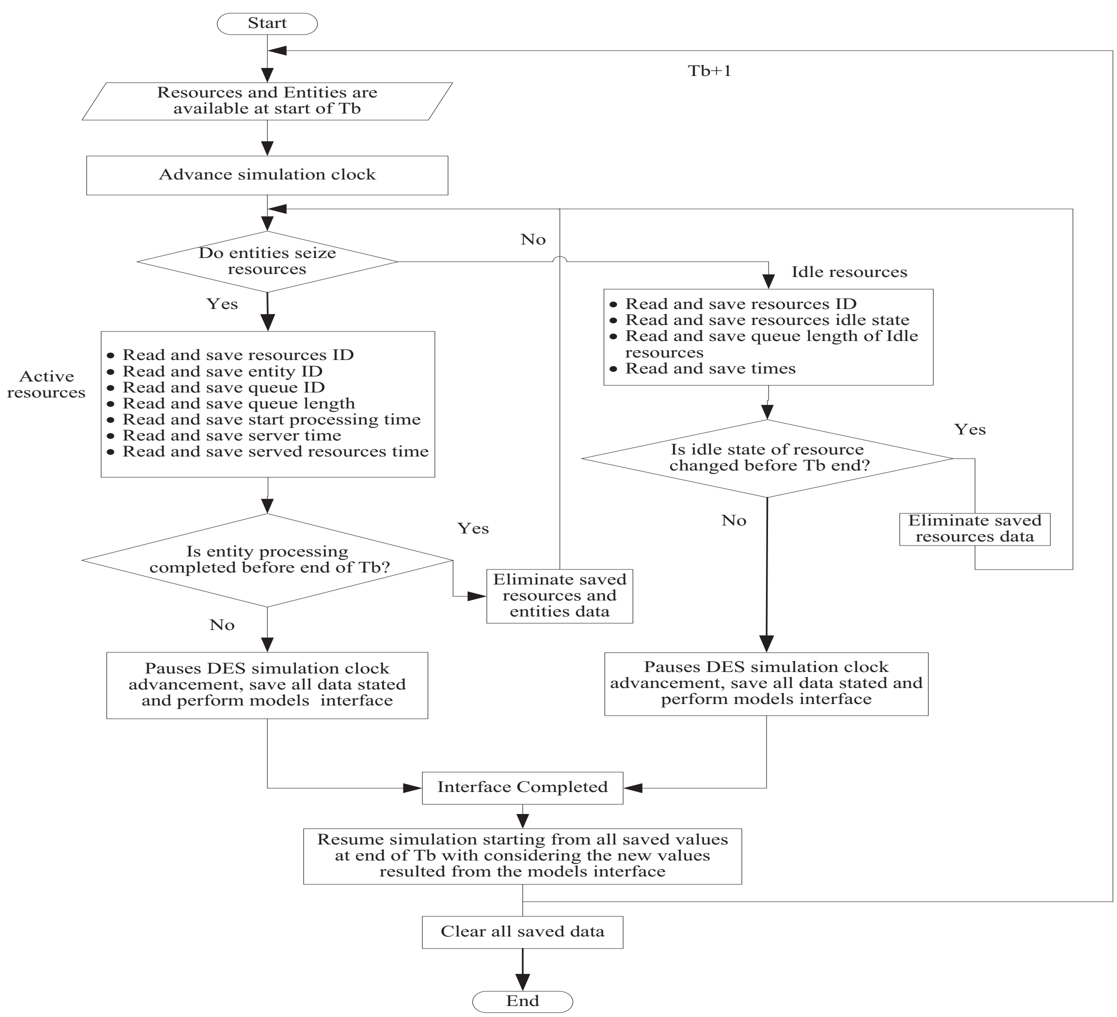


Figure 4.8 Simulation Clock Advancing Algorithm for DES Model

(iii) Time Sequence Diagram (DES, SD, and Executer)

The third and last component of the developed synchronization method is describing the sequence of messages between the models (DES and SD) and the Executer. These messages will carry the commands responsible for implementing the proposed synchronization mechanism as shown in Figure 4.9. In messages (1), (2), (3), and (4), the Executer confirms the initialization of the DES and SD models and makes those two components ready to start advancing the simulation clock. For message (5), the SD provides initial values such as Tb_1 size that triggers the Executer to start advancing the simulation clock (6). Message (7) advances the simulation clock of the DES model to the end of Tb_1, and at end of Tb_1, the states of DES model's variables are read and saved. For Message (8), the needed values for the interfacing process are read and managed by the Executer. In Message (9), the Executer starts interfacing variables by exporting the new values of the variable to SD model. This message flags the end of the computations processed in Tb_1. Starting from message (10), the contents of messages, (6), (7), (8), and (9) are repeated until the simulation time length elapses.

4.2.6 Executer

The Executer is the code developed to integrate DES and SD models through utilizing the components of the developed hybrid simulation method. It focuses on integrating and controlling the hybrid simulation model. The Executer interacts with DES and SD simulation engines to facilitate the integration process based on the developed mathematical formalism, the synchronization protocol, and the discrete simulation clock algorithm. The Executer is responsible for performing the following tasks:

1. Providing the user interface layer that allows inputs and outputs as required by the user.
2. Providing models data import-export management. The selected interface variables that are designated to receive or share their values are specified in the Executer.
3. Implementing the DES simulation clock-advancing algorithm.

The Executer code was developed using VB.NET, and the complete code is provided in Appendix C.

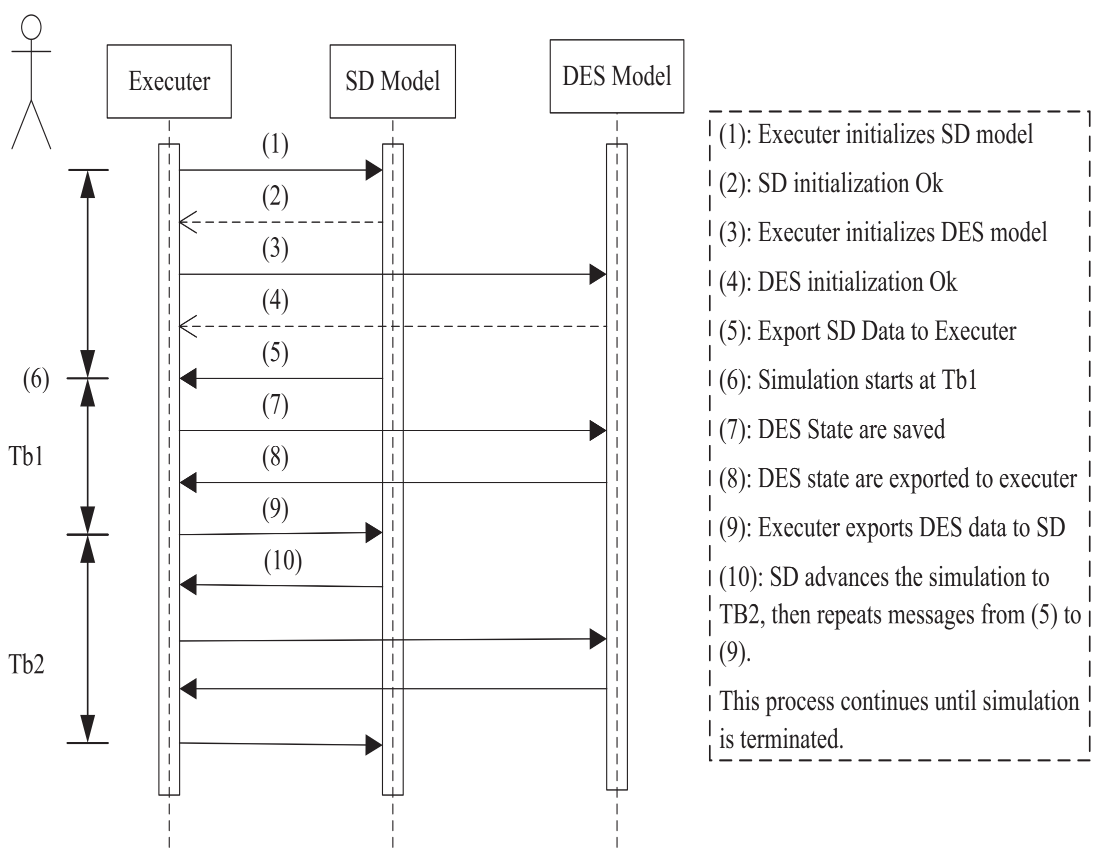


Figure 4.9 Messages Sequence between the Hybrid System Components

4.3 Chapter Summary

This chapter has presented the developed hybrid simulation components. The six components identified in Chapter 3 were developed. The hybrid simulation model is created by decomposing construction projects into units that are responsible for generating the real behavior of the project. These units are classified as operational or strategic based on criteria extracted from the unique characteristics of management decision levels and the philosophies of simulation methods (DES and SD). The DES method is used to model the operational elements while the SD method is used to model the strategic elements. The hybrid simulation model structure and the interface variables are identified during this stage. Three possible hybrid structures that have been used by researchers were summarized. The hybrid structure is controlled mainly by the data flows between the simulation models composing the hybrid simulation model.

In order to describe the hybrid simulation variables, formalism was developed. The formalism function defines the simulation model, input variables, output variables, and the synchronization of variables. The next stage involved resolving the issue of simulation clocks as the DES method drives simulation and updates states in a different way than the SD method. A new synchronization approach was developed based on the Time Bucket concept. This concept is compatible with the SD simulation clock method. Thus, it was necessary to develop an algorithm that drives the DES simulation clock to solve its compatibility issue with the developed synchronization method. Finally, the sequence of messages between the hybrid system components was developed and illustrated using a sequence diagram.

CHAPTER 5

DATA COLLECTION

5.1 Overview

This chapter is dedicated to the data collection of a real-world case from the construction industry used to test and validate the developed hybrid simulation method. The chapter presents a description of the earthmoving operations involved in a dam construction in the Province of Quebec. The earthmoving operations data and the fleet of equipment configuration used in the operations are presented and highlighted. The case study involved two main operations, the excavation of a riverbed and the backfill of three types of soils in three stages. The total scope of work involved in the case study is estimated from the structural design of the dam. The profile of routes and their rolling resistances are computed using the maps and roads characteristics. This allowed using the manufacturer's specifications to calculate the processes durations. Finally, all data are tabulated and prepared for the development of the hybrid simulation model in the next chapter.

5.2 Case Description

In order to implement the developed hybrid simulation method, a real-world case study (earthmoving operations) from the construction sector is considered. The case involves modeling and simulating the earthmoving operations involved in the construction of Saint-Marguerite-3 (SM-3) in 1994-2002. The dam is located on the Sainte-Marguerite River in Sept-Îles City, which is located 700 km northeast of Montréal, Canada, as shown in Figure 5.1. The Sainte-Marguerite-3 project consists of a 171-metre-high rock-fill dam and an underground powerhouse with two turbines of a total installed capacity of 882 MW. At approximately 330 m, Sainte-Marguerite-3 boasts the highest hydraulic head in Québec. The reservoir is 140 km long, with an area of 253 km^2, and is connected to the powerhouse via an 8.3 km headrace tunnel. Peak employment during construction reached 1,200 workers, with an average of 500 workers over the eight-year construction period. Construction included a new 86 km long access road between the Gulf of Saint Lawrence and the SM-3 generating station (Hydro-Quebec 1999), as demonstrated in Figure 5.2. The dam location was specifically chosen to benefit from the 330 m water head.

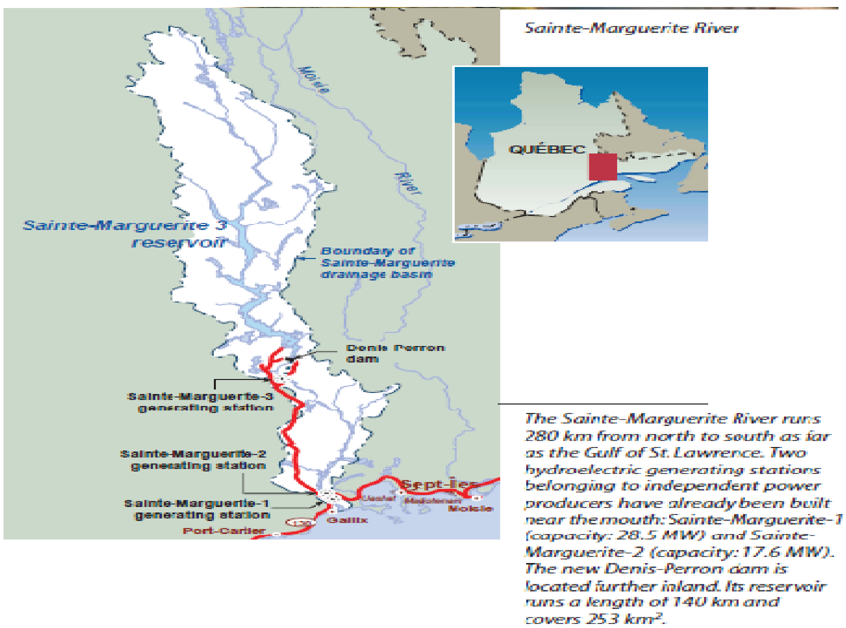


Figure 5.1 Dam Location on Saint-Marguerite River (Hydro-Quebec 2003)

Figure 5.2 The Saint-Marguerite-3 after Construction

The planners of the dam construction considered the weather condition as an obstacle facing the project execution, and planned to execute construction works between the beginning of April and the end of November of each year (244 days). The hybrid simulation model is applied with an objective to simulate the real operations, considering all necessary surrounding factors. Thus, the model is expected to deliver: 1) a better understating of the variable interactions; 2) to locate the bottleneck and identify problematic loops; 3) a realistic productivity and project completion duration; and 4) testing several planning and execution scenarios. The summary of the general SM-3 data used in this case is shown in Table 5.1. The data was collected from Hydro-Quebec (1999), Peer (1998), Hydro-Quebec (2001), and Marzouk (2002).

Table 5.1 Summary SM-3 Dam

Item	Description
Height	171 m
Length at crest	378 m
Crest width	10 m
Base width	500 m
Maximum normal water level	407 m
Minimum normal water level	393 m
Volume of the backfill	6.3 million m^3

5.3 Scope of Work

The management of the project was targeting to complete the earthmoving works involved in the dam construction over three years. The project scope involved excavation of the riverbed and backfill operations. The execution of the work was divided into three stages as demonstrated in Figure 5.3. The simulation tool for the purpose of this research is focused on simulating the excavation and backfill operations. The backfill operations involved backfilling three types of soil: 1) compacted moraine (clay); 2) granular (sand and gravel); and 3) rock. The total scope of the three types of soil was estimated from drawings as 6.3

million m^3. The distribution of the scope of work among the three soil types and their characteristics are shown in Table 5.2. The actual excavated natural soil from the riverbed was 1,038,000 m^3 (Peer 2001). The excavated soil was not used in the dam construction, and was instead hauled away and dumped in a nearby location. The 6.3 million m^3 of soil backfill were borrowed from three pits. The location of the operations zones are depicted in Figure 5.4.

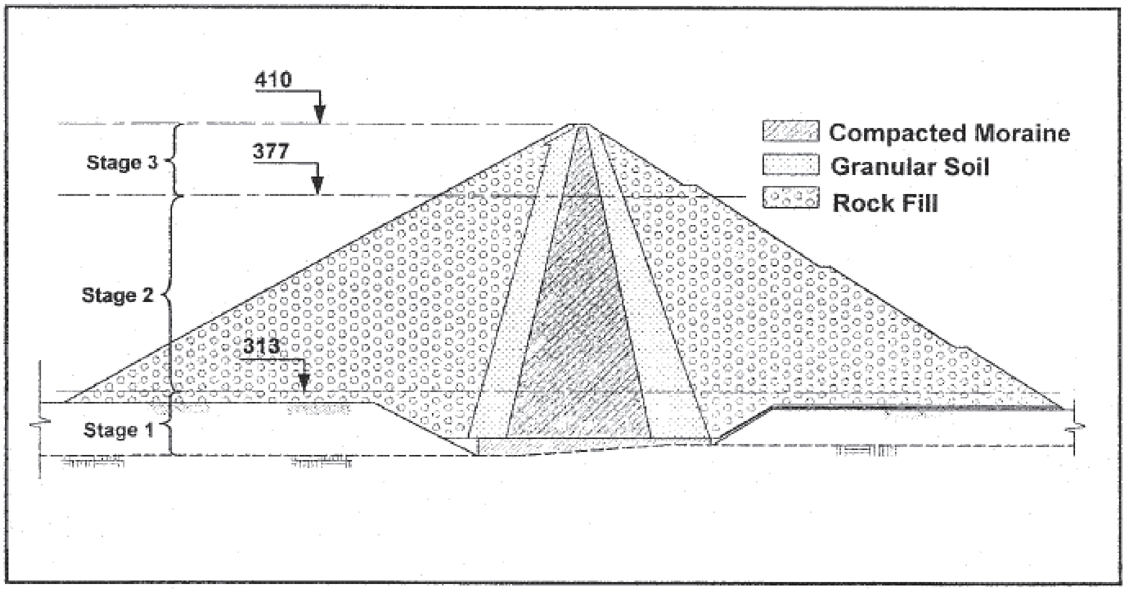
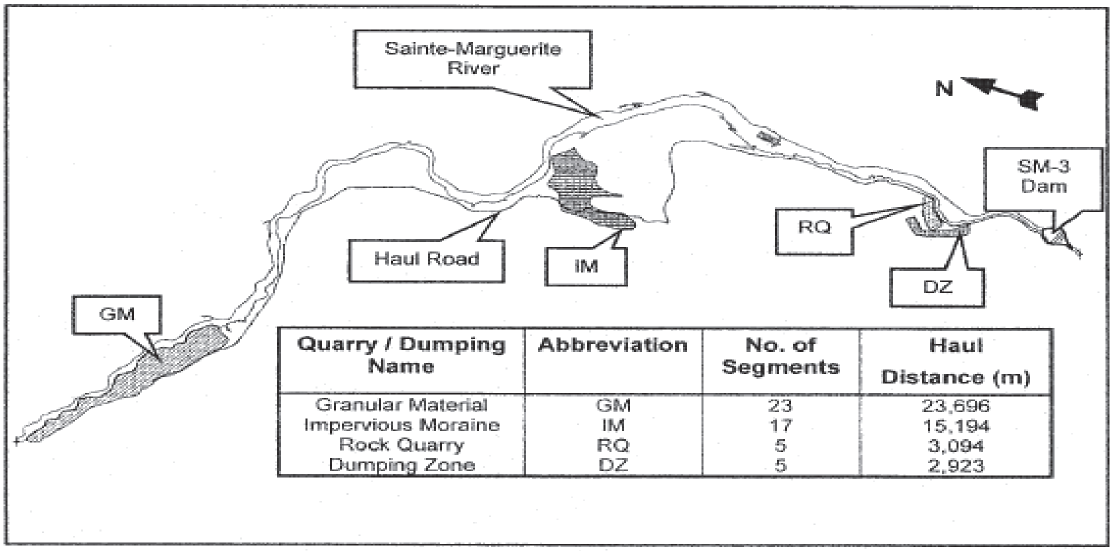


Figure 5.3 Cross-Section of SM-3 Damn

Table 5.2 Scope of Work

Soil Type	**Stage 1 m^3**	**Stage 2 m^3**	**Stage 3 m^3**	**Loose Density (t/m^3)**	**Bank Density (t/m^3)**	**Load Factor**	**Total of Soil m^3**
Rock	192,700	3,209,400	1,602,900	1.66	2.73	80	5,005,000
Granular	14,500	286,500	139,000	1.72	1.93	90	440,000
Moraine	29,200	555,900	269,900	1.66	2.02	100	855,000
Total of Stages	236,400	4,051,800	2,011,800	1.6	2.4	100	**6,300,000**
Excavation	1,038,000						**1,038,000**



Quarry / Dumping Name	Abbreviation	No. of Segments	Haul Distance (m)
Granular Material	GM	23	23,696
Impervious Moraine	IM	17	15,194
Rock Quarry	RQ	5	3,094
Dumping Zone	DZ	5	2,923

Figure 5.4 Quarry and Dumping Zones (Marzouk 2002)

The characteristics of the haul roads (i.e.; lengths, number of segments per road, and the grade of each segment) were determined from the contour drawings that show the profile of the roads used by off-highway trucks. The data pertaining to the characteristics of the different hauling road segments used by trucks are given in Table 5.3.

5.4 Fleet Selection and Configuration

The loaders and haulers, plus the supporting equipment such as motor graders, compactors, spreaders and others, comprise a total fleet. A smart fleet configuration is a significant step toward savings in both time and productivity. Matching equipment capacities and lowering idle times of equipment are essential factors considered in fleet configuration. Equipment selection for earthmoving operations is usually based on the amount of material needed to be moved, distance of hauling, equipment productivity, and project constraints. Several methods and software systems have been developed to select the optimum fleet configuration, such as the Earthmoving Genetic Algorithm (Marzouk 2002), Simphony (Hajjar and AbouRizk 1998), and Fleet Production and Cost Analysis (FPC) by Caterpillar.

Table 5.3 Different Road Segment Characteristics

Seg. No.	Haul Road from GM* to the Dam				Haul Road from the Dam to DZ (for excavated river bid)				Haul Road from IM to the Dam				Haul Road from RQ to the Dam			
	L (m)	% G	% RR	TR	L (m)	% G	% RR	% TR	L (m)	% G.	% RR	% TR	L (m)	% G	% RR	% TR
1	230	2.5	4	6.5	185	0	2	2	973	-5.5	5	-0.5	457	-2.8	4	1.2
2	315	2.4	2	4.4	89	2.4	2	4.4	709	1.7	5	6.7	787	0.1	2	2.1
3	2221	-0.5	2	1.5	710	0.2	2	2.2	824	4.9	2	6.9	710	-0.2	2	1.8
4	950	0.8	2	2.8	287	-0.1	2	1.9	1167	1.9	2	3.9	955	3.3	2	5.3
5	1062	-0.6	2	1.4	842	4.1	4	8.1	899	0.7	2	2.7	185	0	2	2
6	1094	-0.5	2	1.5					1023	-0.5	2	1.5				
7	2911	-0.1	2	1.9					1414	-5.9	2	-3.9				
8	1310	0.2	2	2.2					891	-0.5	2	1.5				
9	915	4.6	2	6.6					962	0.4	2	2.4				
10	1167	1.9	2	3.9					708	-0.2	2	1.8				
11	899	0.7	2	2.7					949	-0.6	2	1.4				
12	1023	-0.5	2	1.5					1031	1.4	2	3.4				
13	1415	-5.9	2	-3.9					1006	-0.6	2	1.4				
14	891	-0.5	2	1.5					787	0.1	2	2.1				
15	962	0.4	2	2.4					710	-0.2	2	1.8				
16	708	-0.2	2	1.8					955	3.3	2	5.3				
17	949	-0.6	2	1.4					185	0	2	2				
18	1031	1.4	2	3.4												
19	1006	-0.6	2	1.4												
20	787	0.1	2	2.1												
21	710	-0.2	2	1.8												
22	955	3.3	2	5.3												
23	185	0	2	2												
	23,696				2,113				15,193				3,094			

RR: Rolling Resistance, TR: Total Resistance, L: Length, and G: Grade
*** GM:** Granular Material**, DZ:** Dumping Zone**, IM:** Impervious Moraine**, and RG:** Rock Quarry**.**

FPC is designed to predict the productivity of each type of loader/hauler fleet, so that there can be direct comparisons between the various types of fleets. For instance, “truck/loader” versus “pusher/ scraper” is taken from the general approach developed and illustrated in the Caterpillar Performance Handbook. FPC takes into account the following factors:

- Site speed limits
- Haul road conditions: gradients, rolling resistance, and distances
- Wait times
- Machine: availability, bucket fill factor, and cycle times
- Site: material density
- Required volumes, and
- Operator efficiency.

In this thesis, the fleet configuration method used by Marzouk (2002) to analyze the Earth Dam Project (SM-3 dam) was used, as shown in Table 5.4. The fleet configuration process was limited to matching of the appropriate equipment, “truck/loader” versus “pusher/ scraper”, and the time durations for the different activities (loading time, hauling time, etc). The issues of productivity, project duration, and cost were addressed in the hybrid simulation model under different implementation scenarios.

The fleets of equipment used for performing the job consisted of three types of off-highway haulers (777D, 773D and 769C) corresponding to loaders (992G, 990SII, and 988F) that were used to haul and load, respectively. For soil spreading and compacting operations, spreader D8R and compactor CS-583C were used. The haulers’ travel times were calculated by using the manufacturer’s Charts, Total Resistances, and road segment lengths, as shown in

Table 5.4. Knowing the total positive resistance and total negative resistance for each road segment (Table 5.3), and using the Rimpull-Speed-Gradeability and Brake Performance Charts for each equipment, the speed was calculated for both the loaded and empty hauler. Knowing the speed of the hauler, then by using the Travel Time Charts under loaded and empty conditions, the loaded hauler travel time and empty hauler return time were calculated. Processes such as loading, spreading, and compaction times were calculated using the equipment's corresponding charts and tables.

The travel times calculated by Hauler's Travel Time Application (HTTA) are similar to that calculated by FPC, since HTTA does not improve the estimated travel time over that obtained from FPC; rather, it provides a generic tool that can be incorporated in simulation models (Marzouk 2000). A triangular probability distribution was considered for the process durations (loading, hauling, returning, spreading, and compaction) and a uniform distribution for the dumping activity was selected. A triangular distribution considers the maximum, most likely, and the minimum, thus it is more representative of reality. The fleet configuration and characteristics along with time duration distributions for activities are shown in Table 5.4 and Table 5.5. Finally,

Table 5.6 demonstrates the equipment fleet configuration that corresponds to the scope of work. For instance, the scope of rock hauling was 5.005 million m^3; therefore, the largest hauler capacity (777D) and loader (992G) were chosen, as logic says large haulers and loaders should be used for a large work scope to decrease the number of equipment needed. This flee configuration process is also checked using FPC and found valid.

Table 5.4 Fleet Configuration and Characteristics

Hauled Material	Hauler Model	Loader Model	Bucket Capacity of Loader (m^3)	Loose Density	Hauled Soil (m^3)	Hauled Soils (ton)	Loading Process- Time Distribution (m)	Hauling Process- Time Distribution (m)	Dumping Process- Time Distribution (m)	Returning process-Time Distribution (m)
Rock	777D	992G	12.3	1.66	49	81.67	(3.94, 4.15, 4.57)	(4.3, 4.53, 4.98)	(1.9, 2.2)	(3.17, 3.34, 3.67)
Moraine	773D	990 SII	9.2	1.66	28	45.82	(3.01,3.2, 3.32)	(19.47, 20.5, 22.55)	(1.6, 1.9)	(16.71, 17.59, 19.35)
Granular	769 C	988 F	6.9	1.72	20	34.36	(2.3, 2.42, 2.5)	(30.6, 32.34, 35.57)	(1.3, 1.5)	(25.85, 26.51, 29.16)
River Bid Soil	777D	375L	4.59	1.6	32	51.41	(4.26, 4.48, 4.93)	(5.32, 5.6, 6.16)	(1.6, 1.9)	(2.86, 3.01, 3.31)

Table 5.5 Spread and Compact Equipment Characteristics

	Bulldozer Model	Productivity (m3/Cycle)	Time Distribution
Spread	D8R	27	(2.47, 2.6, 2.86)
Compact	CS-583C	19	((1.8, 1.9, 2.09)

Table 5.6 Equipment Fleet Distribution for Scope of Work

Soil Type	Stage 1 m^3		Stage 2 m^3		Stage 3 m^3		Total m^3
Rock	192,700	777D and 992G	3,209,400	777D and 992G	1,602,900	777D and 992G	5,005,000
Granular	14,500	769C and 375L	286,500	769C and 375L	139,000	769C and 375L	440,000
Moraine	29,200	773D and 992G	555,900	773D and 992G	269,900	773D and 992G	855,000
Total of Stages	**236,400**		**4,051,800**		**2,011,800**		
Excavation	1,038,000 m^3	773D and 992G	Excavation from river bid to external dumping zone DZ				

5.5 Chapter Summary

This chapter has covered the data collection of the case study used to implement the hybrid simulation model. A case study of earthmoving operations that were involved in the construction of a dam in the Province of Quebec was utilized. The scope of work involved two main parts. The first part was excavating 1.038 million m^3 of soil from the riverbed and dumping it at an off-site location, while the second part was backfilling 6.3 million m^3 of three types of soil. The backfill processes involved were hauling, dumping, spreading, and compacting of soils that were taken from three different pits. The fleet selection primarily relied on the manufacturer's (Caterpillar) FPC software and Marzouk (2002), as well as specifications and handouts. This allowed utilizing the equipment to the maximum possible productivity specified by the manufacturer. Other data such as travel times of equipment and durations needed to perform an activity were calculated based on the manufacturing specifications and considering geographical factors such as slope and terrain nature.

CHAPTER 6

IMPLEMENTATION, RESULTS, AND ANALYSIS

6.1 Chapter Overview

This chapter presents and discusses the implementation stages of the developed hybrid simulation method. The proposed simulation method was systematically implemented by developing a hybrid simulation model for earthmoving operations. It is common in simulation to have the impression that a conceptual model of a project is correct and reflects the real system behavior in the real world. However, when this conceptual model is implemented and carefully tested, discrepancies inevitably appear. Therefore, from the conceptual model, the influential units that generate the real system behavior are identified. These project units are then modeled using DES and SD simulation methods to yield a representative hybrid simulation model. The synchronization of the different simulation models is performed by selecting variables that are involved in the integration process, such as *sender variables*, *interface variables*, and *receiver variables*. These variables create the protocol of interactions among the simulation models. The hybrid simulation model is tested using three selected scenarios to represent the ideal and real situation. Finally, a detailed discussion and analysis of the outcomes of the hybrid simulation model are conducted based on the case study scenarios.

6.2 Implementation Outlines

The hybrid simulation method is designed to support developing models that interact concurrently based on the three identified hybrid model structures discussed in Chapter 4. However, as discussed earlier, the problem being addressed by simulation dictates the hybrid modeling structure that should be adopted. The case study modeled in this thesis required unidirectional interactions of models' variables. Based on the developed method, two way interactions between models is possible, this would capture any significant change in the model behaviour. However, this interaction was not tested in this thesis using the developed method. While examining the available simulation software (e.g. EZStrobe and Vensim) to implement the method concurrently, it became clear that concurrent simulation models interaction was not visible due to limitations associated with the software systems. Further investigation of the problem revealed the need to develop new software systems that support DES and SD models in order to accomplish the concurrent interactions. Alternatively, a sequential implementation of the method was adopted (means preparing the DES models outputs and fed them into a certain variables in SD model).

6.3 Planning Scenarios from the Perspective of Simulation Modeling

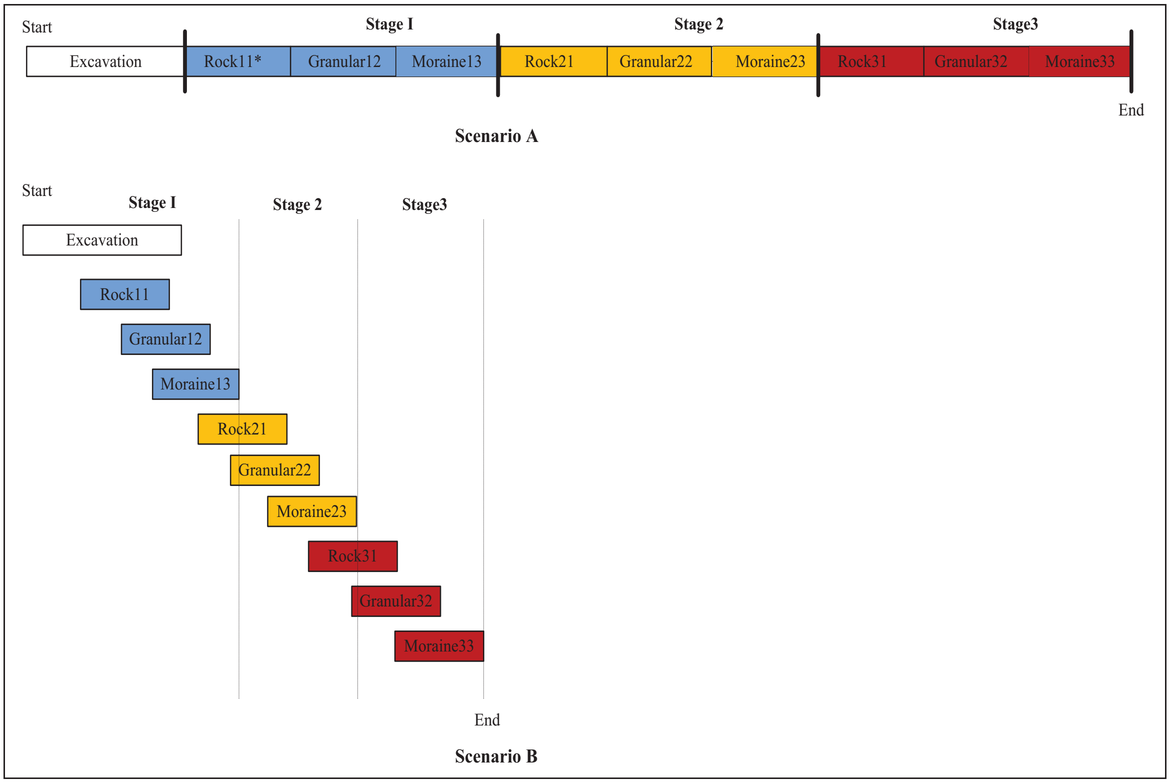
The total scope of work involved ten earthmoving operations, in which one operation was excavating the riverbed and nine operations involved soil backfilling. Due to the harsh cold weather at the dam construction location, the project execution was planned to complete on three stages over three years. Each phase started in April and ended by November.

In actual work implementation, the execution of the work can follow two scenarios, as shown in

Figure 6.1. Scenario A represents scheduling the operations in a sequential format (finish-to-start relationship), while scenario B shows another alternative of scheduling the operations with overlapping. The common practice in the real execution of construction projects is that activities not on the critical path are overlapped in execution (fast tracking), as long as there are no constraints. As shown in the cross sectional diagram of the dam structure

(Figure 5.3), work execution in reality should overlap for two reasons. The first reason is that the completion duration of earthmoving operations is limited to three years, with the work being constraint to 7 months a year due to weather conditions. For example, stage I involved three types of soil backfilling with a possibility of starting the successor operation after a certain percentage of the work was completed correctly in the first operation. This allows a significant amount of time and cost savings. The second reason is the need to backfill soil in layers to support the following layer from a structural perspective. It is clear from the cross-section of the dam structure that soil layers need lateral support in order to allow the construction process to be efficient and safe. For instance, considering stage II of the filling work, the rock layers have a sharp slope from the side of granular soil layers, and the same condition exists between granular soil layers and compacted moraine layers. Therefore, it is imperative for efficient and safe work execution that the layers of the three types of soil be executed concurrently with lead-time. Using hybrid simulation method to model the ten operations shown in the Figure 6.1 allowed modeling the earthmoving operations from system perspective. Consequently, the fragmented nature and isolation of the operations from their context and strategic level were overcome.

In order to compute the safe lead-time of overlapping operations, the heights of the backfill soil layers shown in the cross-section of the design and scope of work (Table 5.2) for each stage were used. From the geometry calculations, it was found that the ten earthmoving operations (excavation and earth filling) could be modeled using a hybrid simulation concurrently with 50% scope overlapping between two consecutive operations. This means that, as soon as 50% of the current operation scope has been completed, the successive operation can start.



***Rock11: (stage I, order of the scope)**

Figure 6.1 Work Stages Sequence

6.4 Developing the Hybrid Simulation Model

The hybrid simulation model is developed based on the proposed hybrid simulation method described in Chapter 4. While the developed hybrid simulation method is generic and can be applied to different hybrid model structures, the implementation in this thesis is limited to the *strategic-operation hybrid simulation structure*. The developed hybrid simulation model for the SM-3 project encompassed six stages: 1) identifying project units and simulation method selection; 2) developing DES and SD models; 3) identifying the hybrid structure and interface variables; 4) formalism of variables; 5) integration (hybrid simulation model); and 6) validation and testing. These six stages are elaborated on in the following subsections.

6.4.1 Identifying Project Units and Selecting Simulation Method

Developing a simulation model starts by defining the influential units in the conceptual framework of such model. These influential units should be capable of generating the real system behavior in the virtual world. The influential elements usually arise from the project management decision levels (Strategic or operational). The case study involved excavation of the riverbed and backfilling three types of soils. The excavation operation involved excavating, loading, hauling, and dumping processes, while the soil backfilling operation involved processes such as loading, hauling, dumping, spreading, and compacting. These elements represent the direct action taken by the operational level to respond to management policies and decisions. On the other hand, there are influential elements in the project that greatly affect operations outcomes, such as weather, overtime, burnout, and skills. These elements arise from the strategic/context management level. The elements that arise from this level of the project are further discussed in the following sections.

Rework: The rework cycle is considered the most important single feature in construction execution that generates dynamic behavior (Cooper 1993a). Neglecting the rework cycle of construction works is a main reason for models not representing real behavior. Its iterative nature generates more rework, thus creating problematic behaviors that often stretch out the project's duration. The backfill soil is compacted as per the dam design specifications. The Proctor Compaction Test was expected to achieve 100% density. In reality, not all the work performed passes this test limit. Layers of compacted soil that fail the test need to be reworked. The rework cycle is a dynamic cycle that is affected by the level of overtime, skills, and schedule pressure. In the implementation of the case study, it is assumed that rework needs to rework only one time and thus that all reworked scope is assumed to pass the test in the second round of work and testing, as experience suggests that compaction work highly possible passes quality standard from the first rework level.

Schedule Pressure: It is defined as "a reduction from the normal experienced time or optimal time typical for the type and size of project being planned within a given set of circumstance" (CII 1990). Schedule pressure must be included in the simulation model and handled appropriately. When people realize that they have more time to complete a process than planned, their productivity is usually reduced (Sterman 2000). On the other hand, excessive schedule pressure may deteriorate a crew's productivity. However, managing schedule pressure up to a certain limit can increase the productivity and put more and acceptable pressure on a worker to accelerate work execution (Sterman 2000). The reference model that quantifies the impact of schedule pressure on productivity cited by Lee (2005) is used.

Depth of Cut: one of the important factors that affect the excavator's productivity is depth of cut of the excavated soil. Depth of cut affects the productivity in two ways. First, depth of cut restricts fleet movement and hence increases the loading and hauling time (Amirkhanian and Baker 1992). Second, the equipment's operator faces difficulty in filling the bucket of the excavator in one pass when the depth of cut increases beyond certain depths (Kannan 1999). As the work progresses and the depth of cut increases, this will affect the excavator's productivity significantly.

Weather Condition: Earthmoving operations are executed in the outdoor environment and therefore are affected by various weather conditions. The weather factor is an influential factor that affects the productivity of the earthmoving fleet.

Heavy rainfall often leads to a complete suspension of earthmoving operations due to saturated and unworkable soil conditions. The rainwater has adverse effects on the condition of hauling roads and on the possibility of the fleet to function in an optimum manner. The impact of the precipitation rate on earthmoving projects has been investigated thoroughly by El-Rayes and Moselhi (2001). The study presented a quantification method of days and productivity losses due to the amount of rainfall, as shown in Table 6.1. The outcome of the study has been utilized in this research to predict days lost due to monthly precipitation amounts. The average monthly precipitation amount for the site location (Sept-Ile, Quebec) has been acquired from the National Climate Data and Information Archive-Canada. The data represents the average monthly precipitation amount in millimeters, taken from 29 years of data (1971-2002).

Table 6.1 Working Days Lost in Earthmoving Projects Due to Weather Conditions (El-Rayes and Moselhi 2001)

Item No.	Amount of precipitation (mm)	Daily Raining Hours	Daily Period	Total Time lost in Days
1	13–25	5	Morning	2
2	13–25	5	Afternoon	1.5
3	13–25	11	10 Morning 1 afternoon	2
4	13–25	14	Overnight	1.5
5	6–13	5	Morning	1
6	6–13	5	Afternoon	0.5
7	6–13	10	Morning + Afternoon	1
8	6–13	14	Overnight	0.5

In order to use the data shown in Table 6.1 for modeling weather impacts, some assumptions should be considered to prepare the data for modeling. For instance, the precipitation rate is represented by maximum and minimum values (e.g., item number 1, amount of precipitation 13-25 mm). To simplify the problem, the midpoint of the maximum and minimum values is taken (e.g., 16.5 mm of precipitation causes two day loss from the schedule). The second assumption is concerned with the total amount of precipitation in a month. For example, the data given by the National Climate Data and Information Archive-Canada as an average per month, e.g., in April, the total average precipitation is 88.8 mm. However, Table 6.1 quantifies the loss of days based on hourly periods. Thus, the model needs to consider all possible hourly periods of precipitation per day in its calculation. This is done by preparing a standard table of the total monthly precipitation in the given periods, and then finding the accumulated lost days corresponding to each period. For instance, in column number (4) of Table 6.2, for the month of May, there is 102.8 mm total precipitation. To estimate days lost, we search in column (2) to find the corresponding crisp number or range of two values. In this case, the illustration example falls between 95 mm and 104.5 mm, corresponding to 8.5 and 9.5 lost days, column (3). Thus, by linear interpolation of the ranges, days lost can be calculated as 9.36 days.

Table 6.2 Days Lost Corresponding to Precipitation Rate

Month	Midpoint of Precipitation ranges (mm) (1)	Cumulative Precipitation Rate of midpoint (mm) (2)	Cumulative Days Lost (3)	Average Total Precipitation per month (mm) (4)	Total Days Lost (5)
April	19	19	2	88.8	8.25
May	19	38	3.5	102.8	9.36
June	19	57	5.5	94	8.5
July	19	76	7	99.3	9
August	9.5	85.5	8	99.8	8.5
September	9.5	95	8.5	91.1	8.25
October	9.5	104.5	9.5	113.2	10
November	9.5	114	10	106.5	9.6

Overtime: Usually, in construction planning, there is a need to compress the project duration, or to accelerate the project execution. These policies are needed to meet contractual obligations or to benefit from incentives. There are three strategies adopted to accelerate project execution. These strategies are overtime, over-staffing, and shift work (Hanna et al. 2005). Overtime achieves schedule acceleration and productivity increase by increasing the amount of hours worked by the labor force beyond the normal working hours per day (8 hours). However, research indicates that labor productivity could be negatively impacted by overtime, causing problems such as fatigue, reduced safety, increased absenteeism, and low morale (Horner and Talhouni 1995). Hiring more workers might not be possible due to site congestion problems, and shift work might cause quality problems. Thus, management has to be prepared in which direction to go and to understand the consequences of each decision. The dynamics resulting from these strategies are included in the model. The lookup table that quantifies the impact of overtime on productivity is cited in Hanna et al. (2005).

Road Surface Condition: Rutted and soft roads that have a higher rolling resistance may affect hauling durations, and consequently affect fleet productivity (Kannan 1999). The frequency with which the manager acts to keep the road in good condition influences the productivity of the fleet. Therefore, information collected in a timely manner provides a buffer from the adverse effects of such a situation. For this case study, it is assumed that the access road condition is good for the first 16 hours of the work execution, and then road surface deteriorates slowly until the next scheduled maintenance of the access road (80 hours period). Thus, a certain amount of fleet productivity is expected to be lost.

Operator Skills and Equipment Mechanical Condition: The skills of the equipment operators are an essential factor in the productivity of the fleet. Experienced operators or seasoned ones have less trouble meeting project management objectives, while new operators need to go through trial and error phases. This inevitably creates fluctuations in the productivity of the fleet. The other issue that should be addressed is the equipment's mechanical condition. The productivity of the earthmoving fleet changes depending on the age of the equipment. Quantifying the optimum productivity of the fleet based on equipment age is achieved based on experience and historical data. For example, at Caterpillar, new equipment with a maintenance contract can hit 90- 95% of the designed productivity, while mid-age more typical equipment can reach 80% of its designed productivity. Old equipment productivity could be 60-75%. In this study, the optimum equipment productivity of the fleet is considered as 90% (FPC 1997)

Soil Type: The nature of the soil loaded and hauled influences the productivity of the fleet. Loading soil of high density is different from loading one with low density. In this project, three types of soils are used. The load factor used for moraine is 100%, for granular and excavated soil 90%, and rock 80% (FPC 1997).

The aforementioned discussion paves the way to summarize the influential units that drive the model behavior and affect its outcomes. These units of the earthmoving operations are summarized in Table 6.3. Each unit is tested against the criteria developed in Chapter 4 to select the appropriate simulation method (DES or SD).

Table 6.3 Summary of the Simulation Model Units

Operations	Operational Level Process Units	Strategic/context level Units
Excavation of riverbed	-Excavation -Loading -Hauling -Dumping -Return	Schedule pressure, road condition, operator skill, soil type, equipment age, weather condition, overtime, fatigue, and cut depth.
Rock11*	-Loading -Hauling -Dumping -Returning -Spreading -Compacting	Schedule pressure, quality, road condition, overtime, rework cycle, soil type, operator skill, equipment age, weather condition, overtime, fatigue, and depth of cut.
Granular12		
Moraine13		
Rock21		
Granular22		
Moraine23		
Rock31		
Granular32		
Moraine33		
Simulation Method	***DES***	***SD***

*** Rock 11: the first digit means phase number, while the second digit means operation order in schedule.**

6.4.2 Developing DES and SD Models

The influential units that were responsible for generating the project's real behavior have been discussed and summarized in the previous section. The next stage involves developing the corresponding simulation models (e.g., DES and SD) of those units. The DES and SD models are the ingredients of the hybrid simulation model. In the following subsections, the procedures followed to develop simulation models are presented.

(i) Developing DES Models

The case study of the earthmoving project involves developing ten DES simulation models, of which, one model simulates the tactical aspects of the riverbed excavation operation while nine models simulate the operational aspects of soil backfill operations. The excavation of the riverbed operation involves excavating the soil from the riverbed, loading the excavated soil to off-highway haulers, hauling the excavated soil, dumping the soil, and then the hauler returns to the loading site for the next cycle. The soil backfill involves processes such as loading soil to haulers, hauling soil to dumping site, soil dumping, hauler return, soil spreading, and soil compaction. Table 6.4 presents a description summary of the ten developed DES models. The list of abbreviations of the modeling elements, such as COMBI, NORM, and QUEUE that were developed using the EZstrobe software are also explained in the table. The input data to those models such as scope and durations were described in the data collection chapter. The models were parameterized as indicated by the parameters table and produced selective outputs indicated in the Results table. A sample of the parameters and results tables developed for backfill of Rock 11 operation is shown in Figure 6.2. The outcomes of simulated operations (e.g. excavating, dumping, spreading, and compaction rates) are the variables needed by the SD model to account for the operational level. Other variables can be included in the hybrid simulation process; however, in the developed hybrid model in this thesis only mentioned variables are required by the SD model. Usually, the operational variables in the SD modeling are entered as deterministic values as SD modeling technique is limited in the operational aspects. Therefore, these variables are computed by the DES as probability distributions and fed where appropriate into SD model to reflect the operational level of the operations.

Parameters Table

nTrucks	Number of trucks	8
nLoaders	Number of loaders	2
AmtOfSoil	Amount of soil in m3	192700
nBulldozer	Number of Bulldozer	3
nCompactor	Number of CompactorWt	3

Results

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

Figure 6.2 DES Model Parameters and Results

Table 6.4 DES Model Input Elements

<table>
<tr><th>Operation</th><th>Operation ID</th><th>Process ID</th><th>Description</th><th>Queue ID</th><th>Description</th></tr>
<tr><td>Excavation of the riverbed</td><td>Excavation</td><td>-Excavate
-Load
-Haul
-Dump
-Return</td><td>-Excavate soil from riverbed
-Load soil into haulers
-Haul soil to dumping site
-Dump soil
-Hauler returns to loading site</td><td>SoilAv
BulldozerWt.
SoilToLoad
LdrsWt
TruckWt
WtDump
Spotter
SoilInPlc</td><td>-Soil available for excavation(start)
-Bulldozer ready to excavate.
-Soil excavated and ready to load.
-Loader is ready to load soil.
-Hauler is ready to be loaded.
-Waiting to dump soil.
-Spotter is ready to direct hauler.
-Soil is dumped (end).</td></tr>
<tr><td>Backfill of Rock Material</td><td>Rock11*</td><td>-Load
-Haul
-Dump
-Return
-Spread
-Compact</td><td>-Load soil into haulers
-Haul soil to dumping site
-Dump soil
-Hauler returns to loading site
-Spread soil in dam site
-Compact soil to design standard</td><td>SoilToMove
LdrsWt
TrucksWt
WtDump
Spotter
SoilInPlc
BulldozerWt
WtToComp
CompactorWt
CompactedSoil</td><td>-Soil available to load (start).
-Loader is ready to load.
-Hauler is ready to be loaded.
-Hauler waiting to dump soil.
-Spotter is ready to direct hauler.
-Soil is dumped and ready for spreading.
-Bulldozer is ready to spread soil.
-Spread soil ready for compaction.
-Compactor ready to compact soil.
-Soil compacted (end).</td></tr>
<tr><td>Backfill of Granular</td><td>Granular12</td><td rowspan="8">Similar as above</td><td rowspan="8">Similar as above</td><td rowspan="8">Similar as above</td><td rowspan="8">Similar as above</td></tr>
<tr><td>Backfill of Moraine</td><td>Moraine13</td></tr>
<tr><td>Backfill of Rock</td><td>Rock21</td></tr>
<tr><td>Backfill of Granular</td><td>Granular22</td></tr>
<tr><td>Backfill of Moraine</td><td>Moraine23</td></tr>
<tr><td>Backfill of Rock</td><td>Rock31</td></tr>
<tr><td>Backfill of Granular</td><td>Granular32</td></tr>
<tr><td>Backfill of Moraine</td><td>Moraine33</td></tr>
</table>

*** Rock 11 (stage I, scope of rock 1)**

Samples of the graphical DES models developed using EZStrobe simulation software is shown in Figure 6.3. Model A shown in the Figure is the simulation model for excavating the riverbed operation, while model B is the simulation model developed for the soil backfill operations, e.g., (stageI-Rock13). The rest of the discrete simulation models for operations of Granular11, Moraine12, Granular21, Moraine22, Rock23, Granular31, Moraine32, and Rock33 are shown in Appendix A.

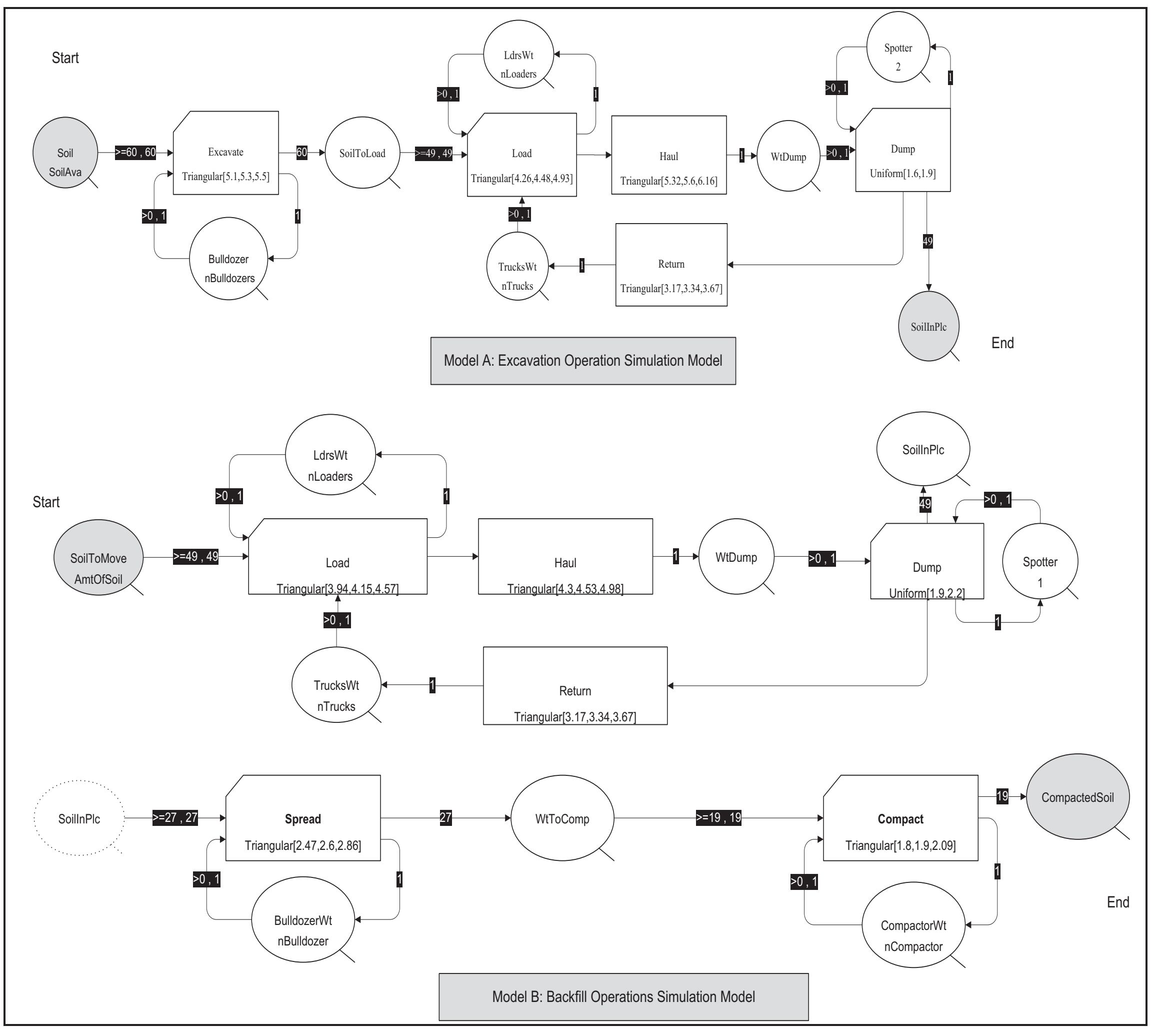


Figure 6.3 Discrete Simulation Models for Excavation and Backfilling Operations

DES Simulation Outputs

The DES models were computed using EZStrobe software. The models were run for 500 cycles to obtain output parameters. The excavation rate and dumping rate from the riverbed were calculated on an hourly basis. For the backfill soil operations, the dumping rate, spreading rate, and compaction rate were also computed on an hourly basis. The equations used to calculate these variables for operation Rock 11 were shown in Figure 6.2. The remaining operations calculations are similar to operation Rock11 and are described in Appendix A. The scope of the rock material was 5.005 million m^3 (79% of total scope), and its borrow pit is distanced at 3094m from the construction site. Therefore, the selected fleet should be fully utilized and produce higher productivity compared to the other soil types as shown in Table 6.5. The dumping, spreading, and hauling rates shown in the table are classified as sender variables and represent the operational inputs needed by the strategic level of the simulation model and will be later communicated to the SD model through interface variables. At this point, the DES model developed is accomplished and the next stage focuses on developing the SD model.

Table 6.5 DES Model Outputs

		Work Scope									
	Process	**Excavation**	**Rock 11**	**Granular 12**	**Moraine 13**	**Rock 21**	**Granular 22**	**Moraine 23**	**Rock 31**	**Granular 32**	**Moraine 33**
	Scope of Work	1038000	192700	14500	29200	3209400	286500	555900	1602900	139000	269900
	Number of Haulers	7	8	10	8	8	10	8	8	10	10
	Number of Loaders	2	2	1	1	2	1	1	2	1	1
	Number of Bulldozers	2	3	1	1	3	1	1	3	1	1
	Number of Compactors	0	3	1	1	3	1	1	3	1	1
Dumping Productivity	**Max Productivity (m3/hr)**	**1421.33*/** 1367.39**	1462.43	216.87	347.88	1462.90	221.28	349.82	1466.84	223.70	424.90
	Min Productivity (m3/hr)	**1222.65*/** 1200.56**	1323.11	157.30	260.10	1323.08	158.10	262.01	1319.26	156.76	337.55
	Average productivity (m3/hr)	**1320*/** 1284.52**	1393.21	187.21	304.55	1393	190	306.29	1393.08	190	381.49
	Standard Deviation	**34.40*/** 28.27**	35.20	15.37	22.19	35.36	16.88	22.06	37.25	17.88	22.13
Spreading Productivity	**Max Productivity (m3/hr)**	-	1462.56	216.94	343.95	1456.75	221.65	349.05	1462.94	221.81	434.98
	Min Productivity (m3/hr)	-	1323.05	157.18	264.07	1329.15	158.26	262.18	1323.15	158.47	327
	Average productivity (m3/hr)	-	1393.21	187.43	304.55	1393.21	190.32	306.29	1393.08	190.23	381.49
	Standard Deviation	-	35.16	15.275	20.32	32.16	16.67	22.06	35.25	16.88	27.13
Compaction Productivity	**Max Productivity (m3/hr)**	-	1464.98	218.26	341.79	1458.77	221.66	351.71	1466.72	221.83	430
	Min Productivity (m3/hr)	-	1321.41	155.15	266.14	1327.09	158.06	260.42	1319.79	158.11	331.00
	Average productivity (m3/hr)	-	1393.21	187.43	304.55	1393.21	190.68	306.29	1393.08	190	381.49
	Standard Deviation	-	36.17	16.25	19.32	33.67	16.63	23.06	37.25	16.88	25.13
	Duration (Hours)	808.33	138.44	77	96.79	2303.70	1054.72	1814.44	1151.85	730.42	707.63
			Total backfill duration =8504 Hours. By considering 50% overlapping, duration = 4620 hours								

*** Excavation rate of riverbed, **Dumping rate of Excavation of riverbed**

(ii) Developing SD Model

This section describes the SD model development that accounts for the dynamics generated in the earthmoving project due to management policies and the surrounding factors. The SD model development encompassed four stages: 1) identifying feedback processes; 2) identifying the model boundary; 3) modeling dynamics; and 4) model testing and validation.

1- Feedback processes

The system behavior results from the interactions of its units defined within the model boundary. Therefore, in order to investigate the earthmoving behavior that is subjected to management policies and influences of the surrounding factors, a causal-effect loops diagram was developed to depict the perceived behavior as shown in Figure 6.4. The causal-effect loops are depicting the conceptual stage of SD model development. Productivity in earthmoving operations is influenced by many factors such as high schedule pressure, operator's skills, weather, road condition, etc. When productivity of the project falls behind the perceived productivity, the anticipated completion date becomes invisible. Consequently, management has to adopt certain policies to reduce the adverse effects of productivity loss. These policies can be overtime, hiring new workers, or extending the project completion duration to foster productivity, and finish the project on time. Another aspect that is of concern is that not all work completed meets quality requirements. Rework cycle in construction is inevitable, and error correction may cause a secondary error.

Earthmoving projects are greatly vulnerable to the impact of weather conditions. This adverse impact may not only cause delays but also create unfavorable work conditions such as bad access road conditions and soil expansion. The off-highway equipment used in work execution face difficulty in operating in bad weather conditions. The dynamics generated from these factors extend the project completion duration and create reinforcing and balancing loops as shown in Figure 6.4. The shown causal loops diagram is further broken into sub causal-effect loops and discussed in detail in the following subsections. The decomposing of the causal-effect loops into sub loops does not violate the holistic principal of the system dynamics technique. It is only a step used to explain the feedback process in elaboration.

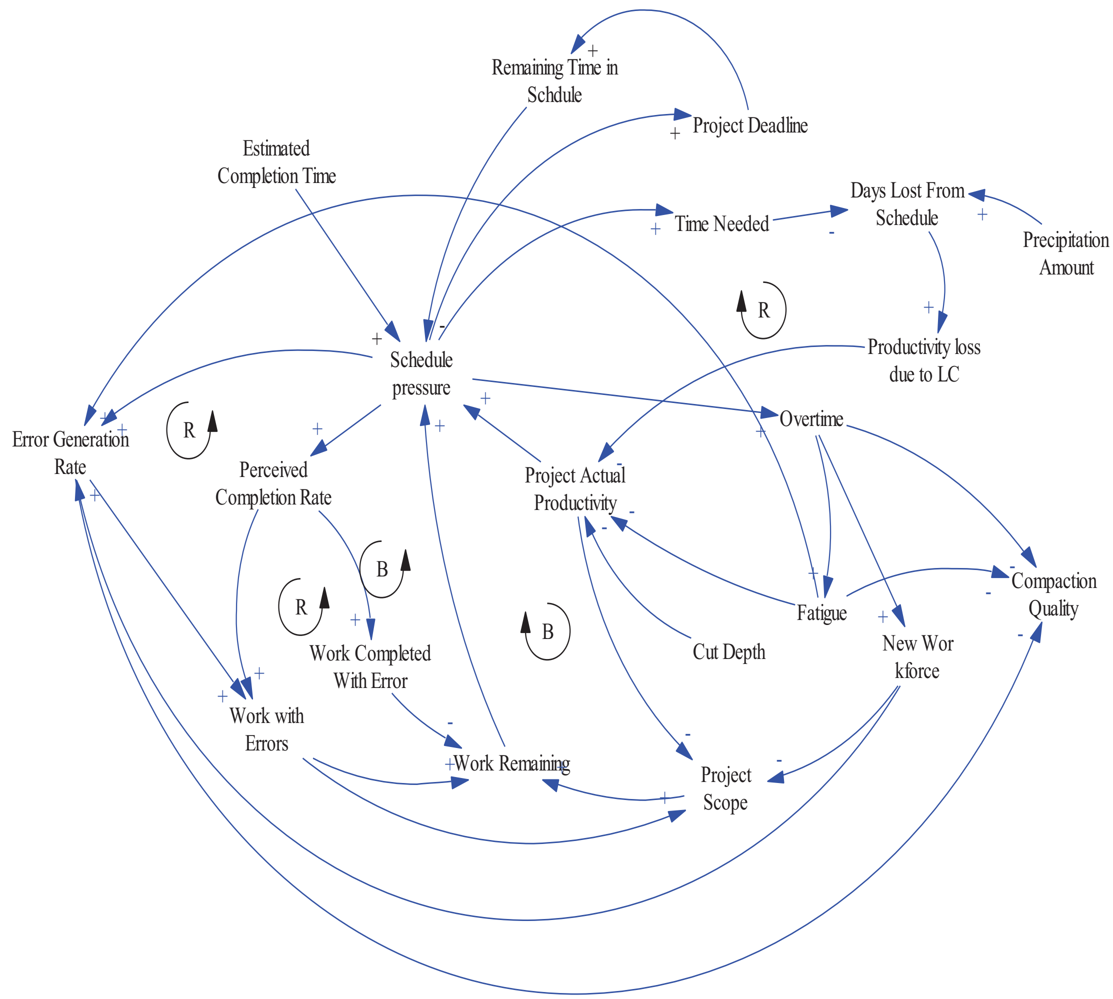


Figure 6.4 Feedback Process in Earthmoving Operations

A) Workflow and Control Loops

Workflow loops address the flow and sequence of work in the model while control loops describe policies adopted by management to control the project parameters. The schedule feedback process is common in construction projects. Slipping the schedule deadline structure in response to schedule pressure had been used in many social science models (Richardson and Pugh 1981; Abdel-Hamid 1991). The schedule causal loops structure uses a target to describe optimum required results. The target is usually the project deadline date, while in reality and based on the effect of endogenous or exogenous variables, this target drifts toward the re-estimating of project deadline in light of project dynamics. Thus, schedule pressure will be increased as a result of the difference in re-estimated and target project completion durations. Other possible policies to decrease the drift from the target can be increasing the workforce, adopting overtime, working faster or scope change. The four options are widely used in the construction sector. Each of these options has both positive and negative impacts. For example, increasing the workforce might be subjected to budget limitations and skills level, overtime might cause fatigue, working faster inevitably will result in lower quality, and finally scope change might not be possible as this is an extreme response to schedule slippage.

Figure 6.5 demonstrates typical feedback loops that can be observed in project control management (Rodrigues and Bowers 1996). The Figure shows three loops that result from management decisions and policies. The loops are called *balancing loops*, and have a negative polarity (-), as emphasized by the “B”. The *balancing loops* (-) are responsible for making the system more stable, while *reinforcing loops* (not shown in the figure) of positive polarity (+) try to drive the system out of limits. The polarity of the loops is a result of the multiplication of the variables’ signs shown on the arrows. In loop “B1”, when project progress is behind schedule, the management responds to perceived schedule slippage by either increasing the resources or extending the project completion time. The decision of increasing the resources should increase the productivity rate. As a result, perceived progress reduces the effort and eventually brings the forecasted completion date forward. Alternatively, in loop “B3”, the strategy to respond to schedule slippage is to adjust the project completion date. Balancing loops are desired in the project and establishing them is not an easy task for the manager, as there are continuous adverse influences from external variables, as emphasized in the rectangles in Figure 6.5. For instance, increasing the workforce is expected to increase the progress rate. However, on one hand, the decision of increasing the workforce is restricted by constrains such as budget limitations, availability of skilled workforce, space limitation, etc., and on the other hand, the expected increase in the productivity of the workforce is constrained by factors such as motivation, training level, and ability to work under high overtime. These kinds of mechanisms of cause and effect feedback loops are responsible for the real behavior of projects, and effort spent on understating loop evolution and their interactions mechanism will enhance the understating of the management problems of construction projects.

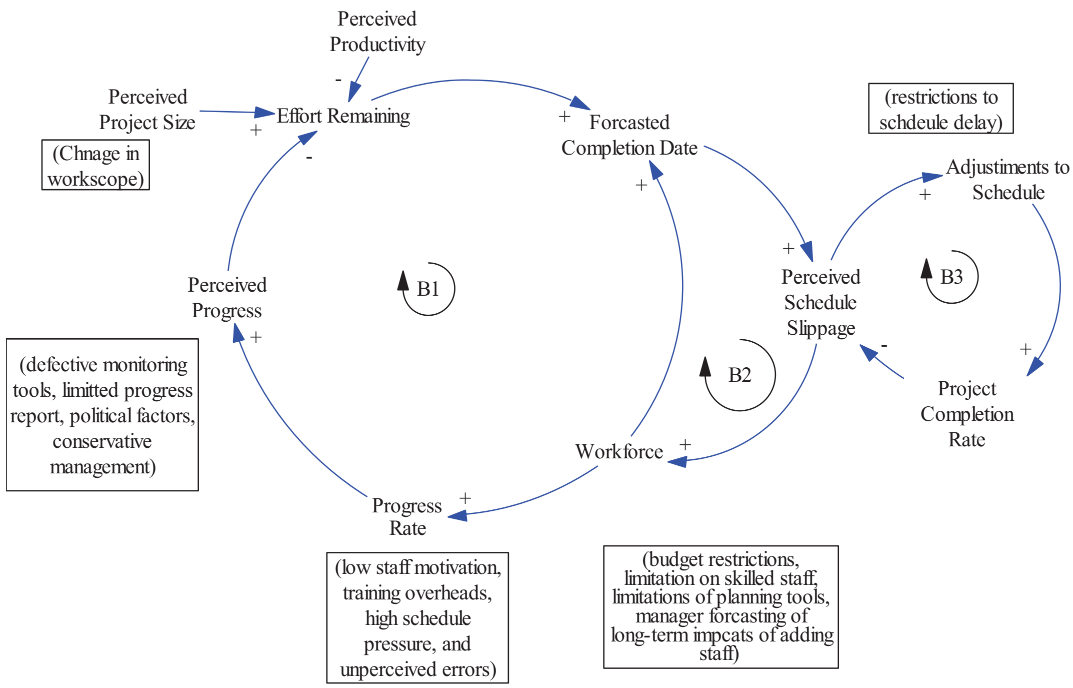


Figure 6.5 Schedule and Control Feedback Loops (Adapted from Rodrigues and Bowers 1996)

B) Rework loops

The difference between perceived and actual progress is further investigated using the rework cycle shown in Figure 6.6. The work rate is determined by the available workforce and the gross productivity. As the project progresses toward completion, scope of work is expected to get reduced. However, some work involves faults, which are detected at later times. This work needs to be reworked, and consequently work remaining decreases the anticipated project progress. The rework cycle is affected by the resources level and their productivity, the quality of the performed work and the time of error discovery. It is also essential to define whether it is a fundamental error or a minor one. Management usually pays attention to resources and productivity; however, little thought is given to time needed to discover errors. It is important to note that rework deals with error corrections, and should not be confused with quality, which has its own feedback loops. Rework loops result from the additional amount of work added to the "Work To Do", as shown in Figure 6.6. The rework cycle shows three typical loops that result from management policy. The first loop (R1) allows the schedule pressure to increase to a certain limit. High schedule pressure increases error rate and flawed work. Consequently, the remaining work or initial scope increases. This adversely impacts and increases the schedule pressure again. In the second loop (B2), the management policy response to the increase in the schedule is to increase the productivity rate, and as a result, work remaining decreases. The third and last loop shows that the increase of work rate could increase the errors, and hence increase the work remaining and schedule pressure.

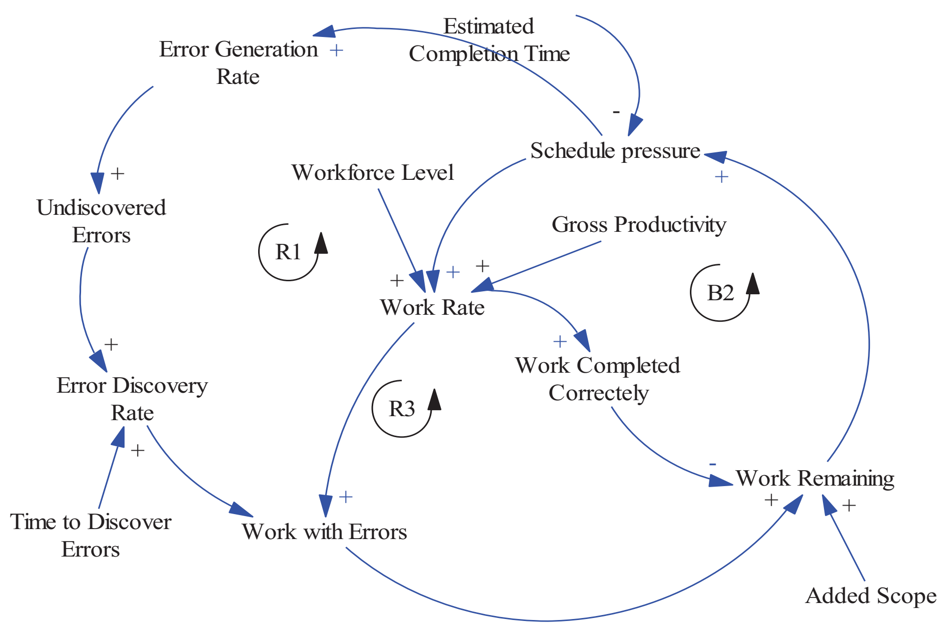


Figure 6.6 Rework Feedback Loops (Adapted from Richardson and Pugh 1981)

<u>C) Quality Loops</u>

The quality feedback loops represent the standards at which work completed is accepted and approved. The quality loops are different from the rework loops, and should be dealt with differently. Figure 6.7 demonstrates three loops. The balancing loop B1 describes the gap in quality between the standard and the actual quality. The quality gap between the two increases as the standards increase. Thus, work completed and released with the required quality is affected negatively. This results in extra pressure imposed on the schedule and on the workforce. The tendency of the workforce to decrease quality as a response to increasing pressure is positively impacted. The reinforcing loop R3 focuses on the classical behavior of the quality cycle, where actual work quality increases and faulty work decreases. This loop works toward decreasing the remaining work of the project scope.

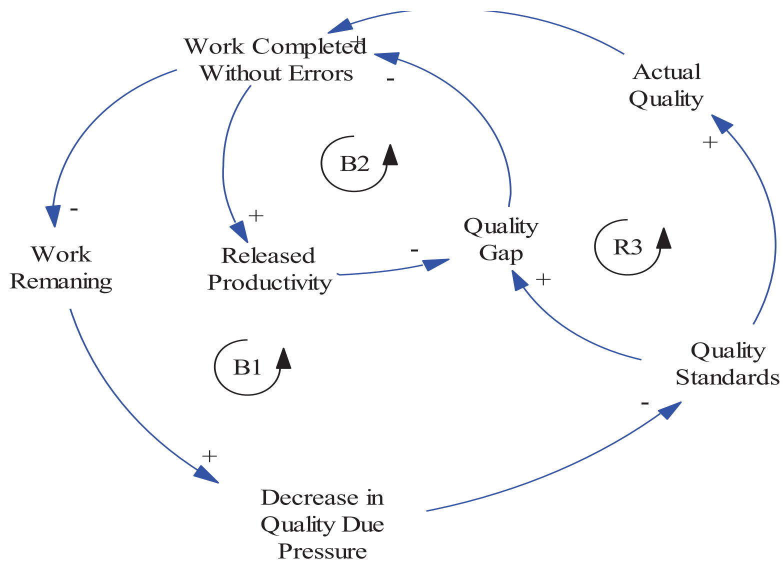


Figure 6.7 Quality Feedback Loops (Adapted from Ford 1995)

The remaining dynamic loops such as weather, soil types, road condition, and depth of the cut are addressed in the coming SD model development section.

2- Model Boundary

The feedback process in SD has a closed boundary within which the behavior of the system is generated. Defining the model boundary involves selecting the components of interactions necessary to generate the behavior of interest as specified by the model's purpose. Variables in the model are classified as *endogenous*, *exogenous*, and *excluded*. *Endogenous* variables are the main concern of all model variables. They are variables in a causal-effect structure whose value is determined by the states of other variables in the system. *Exogenous* variables come from outside of the model, and are unexplained by the model's feedbacks structure. They are involved in a causal-effect structure whose value is independent from the states of other variables in the system. Examples of exogenous variables are planned completion duration and planned productivity. Finally, variables categorized as *excluded* variables are cautiously not included in the structure of the causal-effect feedback process. The exclusion of these variables should not have a great influence on the model representation; otherwise, they have to be added to the model. The SD model boundary for the earthmoving operations is summarized in Figure 6.8.

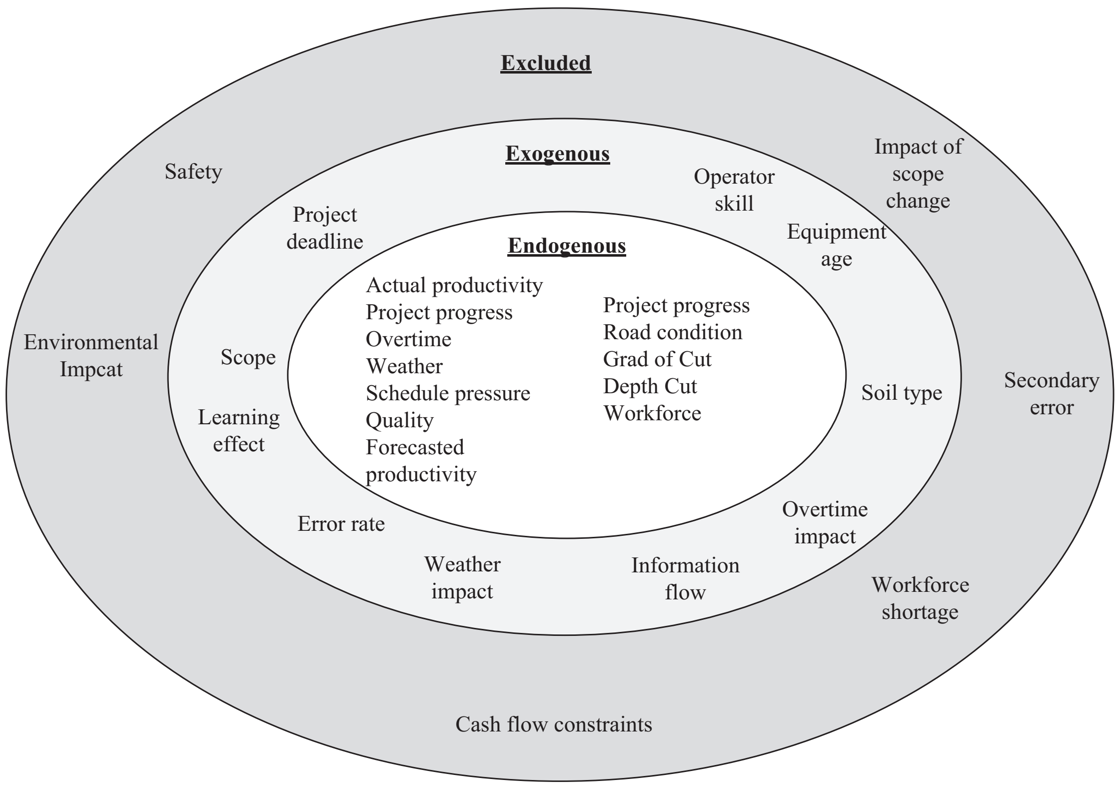


Figure 6.8 SD Model Boundary

3- Modeling Earthmoving Operations Dynamics

This stage focused only on developing and validating the mathematical SD model based on the discussed causal-effect loops structure and model boundary. The causal-effect loops were validated, as this is the main requirement before developing the SD mathematical model. The validation process is conducted on the causal-effect loops and results are shown later in this chapter. The SD model developed for the earthmoving project consisted of five modules: A) excavation of the riverbed and backfill workflow; B) forecasted productivity and schedule pressure; C) weather impact; D) impacted productivity; and E) project cost. The model was developed using Vensim software package. In order to represent multiple operations and processes in the project, the SD model structure was replicated using subscript control function in Vensim. The full list of the mathematical equations is presented in appendix B. The following section discuses in details those modules and their mathematical equations and subscripts.

A) Excavation of Riverbed and Backfill Workflow Modules

The workflow module mainly describes the workflow from the start of the project until its completion. The earthmoving project consisted of a collection of operations that were overlapping in execution. The hybrid structure adopted for the simulation model is to have the SD model as the main model, and the DES models as auxiliary that compute certain variables needed by the SD model. Therefore, the SD model is responsible for depicting the work execution pattern, and where needed, the DES model will be triggered by the SD model to supply data. In the earthmoving case study, there were two main workflows: one was excavation of the riverbed, while the other was backfilling of three types of soil on three phases. Thus, two workflow structures are required in the execution, since the two workflows involve different processes.

The model that depicted the excavation operation from the riverbed is shown in Figure 6.9. For instance, the excavation scope is stocked in “Soil to Excavate from Riverbed” stock. This represents the initial scope of the excavation work. Then, the scope flows to the

"Excavation Rate" where it is processed. The rate represents the productivity of the excavator per unit time, which can be either deterministic or stochastic. The excavated soil is stocked in the "Excavated Soil" stock. This represents the end of the excavation operation and the readiness of the soil to be hauled to the dumping site. The next sequence in the workflow involves processing the stock "Excavated Soil" through "Dumping Rate of Excavation." This stage represents the hauling and dumping of the soil. The dumped soil is finally released to stock "Soil Dumped Ex." The excavation operation is considered completed when the stock "Soil Dumped Ex" reaches 100% of the initial scope of work.

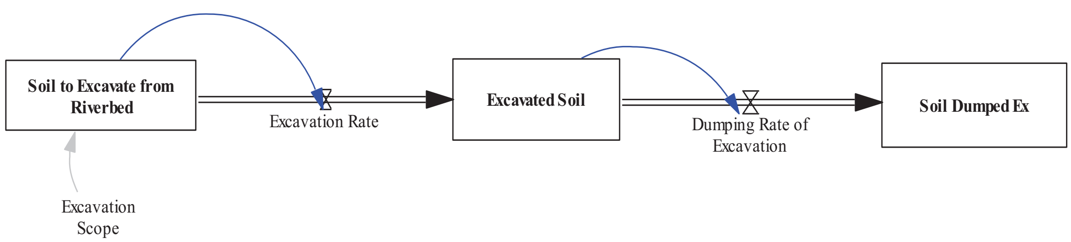


Figure 6.9 Workflow of Excavation operation

The second workflow structure of the SD model depicts the flow of backfill soil operations as shown in Figure 6.10. Initiation of the work is typically started in stock "Soil to Haul". The nine work packages are stocked at a rate of 50% (as soon as the successive operation reaches 50% of its content, the predecessor to this operation starts stocking). Each operation involves hauling-dumping, spreading, compaction, quality check, and released work. The "Soil to Haul" stock is processed at "Dumping Rate" which is then stocked in the "Soil Dumped" stock. The dumped soil is inputted into the next stock that represents the start of the soil spreading operation "Soil to Spread" stock. The soil stocked at "Soil to Spread" stock is processed by the rate "Spreading Rate," and then spread soils are stocked at the "Soil Spread" stock. The next stage involves compaction of the soil spread. The compaction is performed by the rate "Compaction Rate," and then stocked at the "Soil Compacted and

Ready for Quality Check" stock. Now, a quality check based on the design standards is ready to take place.

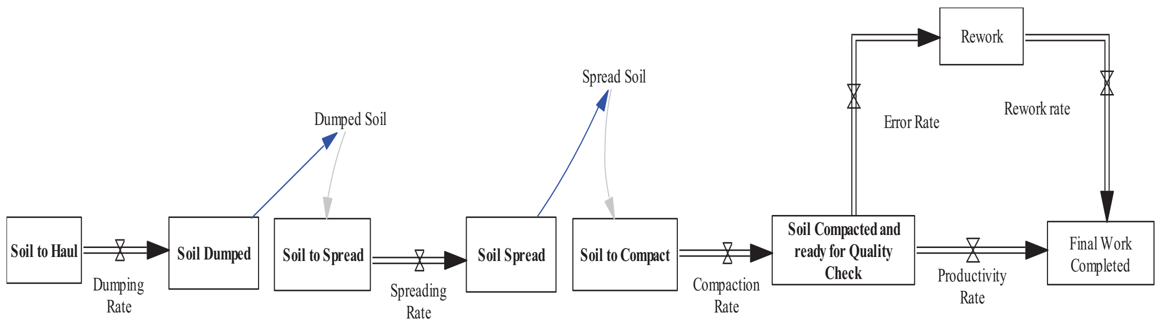


Figure 6.10 Workflow Structure of Backfilling Operations

The quality of work depends on many factors. In reality, part of the executed work contains flaws and needs to be reworked. Some errors in the work are only discovered at the quality test operation and other errors that were generated from the rework cycle are discovered later on. Therefore, the flawed work is passed to stock "Rework" where it is processed by rate "Rework Rate." The final reworked portion of the work packages is stocked in "Final Work Completed." The other portion of the work that was executed based on quality standards, and did not include faults, is passed to the "Final Work Completed" stock through the rate "Productivity Rate." The rates "Productivity Rate" and "Rework Rate" represent the actual rates at which the work packages are executed.

What has been discussed in the previous two SD structures is the flow of work packages from the start point of project execution until completion. In the following stages of the model development, the SD modeling efforts are concerned with loading the required variables on the model based on the previous discussion to generate real behavior. The first step is to load the work package variables onto the model and check their execution sequence, whether or not it is as planned. The SD structure shown in Figure 6.11 describes the loops and variables modeled to generate the workflow of the work packages for the processes of "Loading-Dumping," "Spreading," and "Compaction and Rework Cycle." The purpose at this stage is to generate the base behavior based on the ideal planning situating and to validate the model before further inclusion of variables in the modeling process. The mathematical equations responsible for generating the basic behavior of workflows are now explained.

The total scope (work packages) of soil backfill is input into the model as subscripts.

Subscript Scope: Rock11, Granular12, Moraine13, Rock21, Granular22, Moraine23, Rock31, Granular32, Moraine33

Each scope corresponds to its value, as shown in the following equation,

Scope Size[Scope]= 192700, 14500, 29200, 3.2094e+006, 286500, 555900, 1.6029e+006, 139000, 269900.

Units: m3

Since the scopes are input as individual subscripts, a function to account for the accumulation of all scopes or packages is necessary. The summation of the total scope is computed as follows:

Total Scope= SUM (Scope Size [Scope!])

Units: m3

Now, as explained in the real case study, the execution of the nine scopes overlapped by 50%. In order to represent this, a switch structure of mathematical equations is designed to allow the simulation model to depict this planning strategy. The switch formulation is as follows.

Loading-Dumping Module

1-Total Scope to haul and dump flow

Soil to Haul[Scope] = INTEG(-Dumping Rate[Scope],Scope Size[Scope])

Units: m3

The rates at which the scope is processed are discussed later in the hybrid model integration. These rates were involved in the interfacing process between the simulation models. Thus, a detailed discussion was carried out in the synchronization section. The scope of work is considered active and being processed by the rates when the following condition is true:

Scope Task is Active D[Scope]=IF THEN ELSE (:NOT: Scope Task is Done D[Scope]:AND: Start Task Flag[Scope] > 0.5, 1, 0),

Units: Dmnl

In addition, scope is considered done when:

Scope Task is Done D[Scope]=IF THEN ELSE(Soil Dumped[Scope] >=Scope Size[Scope], 1, 0)

Units: Dmnl

2- Switch Equations

Switches are a structure of equations that informs the haul-dump module when to start and when to terminate the operation. The equations of the switch are as follows.

Start Task Flag[Rock11]=1

Start Task Flag[Granular12]=SAMPLE IF TRUE(percentage of Dumped Soil[Rock11] >= Start Scope Task Percentage Dumping[Granular12], 1 , 0)

Start Task Flag[Moraine13]=SAMPLE IF TRUE(percentage of Dumped Soil[Granular12] >= Start Scope Task Percentage Dumping[Moraine13] , 1 , 0)

Start Task Flag[Rock21]=SAMPLE IF TRUE(percentage of Dumped Soil[Moraine13] >= Start Scope Task Percentage Dumping[Rock21] , 1 , 0)

Start Task Flag[Granular22]=SAMPLE IF TRUE(percentage of Dumped Soil[Rock21] >= Start Scope Task Percentage Dumping[Granular22], 1 , 0)

Start Task Flag[Moraine23]=SAMPLE IF TRUE(percentage of Dumped Soil[Granular22] >= Start Scope Task Percentage Dumping[Moraine23] , 1 , 0)

Start Task Flag[Rock31]=SAMPLE IF TRUE(percentage of Dumped Soil[Moraine23] >= Start Scope Task Percentage Dumping[Rock31] , 1 , 0)

Start Task Flag[Granular32]=SAMPLE IF TRUE(percentage of Dumped Soil[Rock31] >= Start Scope Task Percentage Dumping[Granular32], 1 , 0)

Start Task Flag[Moraine33]=SAMPLE IF TRUE(percentage of Dumped Soil[Granular32] >= Start Scope Task Percentage Dumping[Moraine33] , 1 , 0)

Units: Dmnl

The rate at which the soils are dumped is formulated as follow:

Dumping Rate[Scope]=MIN(IF THEN ELSE(Scope Task is Active D[Scope], Max Dumping Rate[Scope],0), Soil to Haul[Scope]/ Average time[Scope])

Units: m3/hr

3- Final Dumped Soil

In this module, the final stage involved stocking the executed material in stock "Soil Dumped." In this stock, the nine soil types are stocked separately, thus the output of this stock will show the quantities of each soil and not the accumulation.

Soil Dumped[Scope]= INTEG (Dumping Rate[Scope],0)

Units: m3

Since there is a need to calculate the overall accomplished scopes of the nine operations in a single figure, the model uses the following equation:

Total Soil Dumped=SUM(Soil Dumped[Scope!])

Units: m3

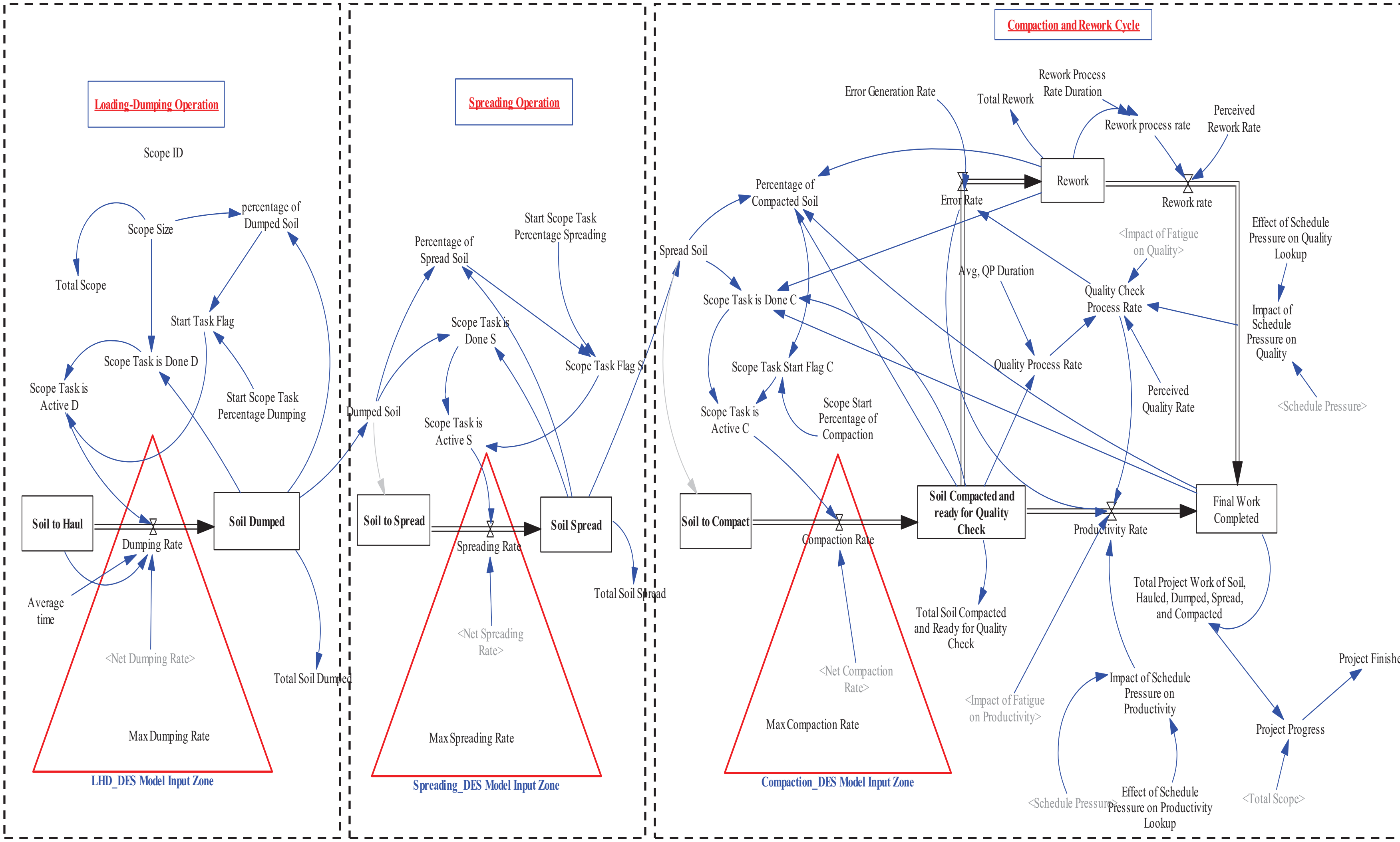


Figure 6.11 Final SD Model of Workflow in Backfilling Operations

Spreading Sub-Module

The soil dumped at stock "Soil Dumped" is used as input for the spreading process. The equations used are described next.

a- Total spread scope discharge.

Soil to Spread[Scope]= INTEG (Spreading Rate[Scope],Dumped Soil[Scope])

Units: m3

Scope Task is Active S[Scope]=IF THEN ELSE (:NOT: Scope Task is Done S[Scope]:AND: Scope Task Flag S[Scope] > 0.5, 1, 0)

Scope Task is Done S[Scope]=IF THEN ELSE(Soil Spread[Scope] >= Dumped Soil[Scope], 1, 0)

Units: Dmnl

b- Switch Equations that control the spreading flow

Scope Task is Active S[Scope]=IF THEN ELSE (:NOT: Scope Task is Done S[Scope]:AND: Scope Task Flag S[Scope] > 0.5, 1, 0)

Scope Task Flag S[Rock11]=1

Scope Task Flag S[Granular12]=SAMPLE IF TRUE(Percentage of Spread Soil[Rock11] >= Start Scope Task Percentage Spreading[Granular12] , 1 , 0)

Scope Task Flag S[Moraine13]=SAMPLE IF TRUE(Percentage of Spread Soil[Granular12] >= Start Scope Task Percentage Spreading[Moraine13], 1 , 0)

Scope Task Flag S[Rock21]=SAMPLE IF TRUE(Percentage of Spread Soil[Moraine13] >= Start Scope Task Percentage Spreading[Rock21] , 1 , 0)

Scope Task Flag S[Granular22]=SAMPLE IF TRUE(Percentage of Spread Soil[Rock21] >= Start Scope Task Percentage Spreading[Granular22], 1 , 0)

Scope Task Flag S[Moraine23]=SAMPLE IF TRUE(Percentage of Spread Soil[Granular22] >= Start Scope Task Percentage Spreading[Moraine23] , 1 , 0)

Scope Task Flag S[Rock31]=SAMPLE IF TRUE(Percentage of Spread Soil[Moraine23] >= Start Scope Task Percentage Spreading[Rock31] , 1 , 0)

Scope Task Flag S[Granular32]=SAMPLE IF TRUE(Percentage of Spread Soil[Rock31] >= Start Scope Task Percentage Spreading[Granular32], 1 , 0)

Scope Task Flag S[Moraine33]=SAMPLE IF TRUE(Percentage of Spread Soil[Granular32] >= Start Scope Task Percentage Spreading[Moraine33], 1 , 0)

Units: Dmnl

The rate at which the soils are spread is formulated as follows:

Spreading Rate[Scope]=IF THEN ELSE(Scope Task is Active S[Scope],Max Spreading Rate[Scope],0)

Units: m3/hr

c- Final Soil Spread

Soil Spread[Scope]= INTEG (Spreading Rate[Scope],0)

Units: m3

Total Soil Spread=SUM(Soil Spread[Scope!])

Units: m3

Compaction Sub-Module

a- Total Compaction Scope

Soil to Compact[Scope]= INTEG (Compaction Rate[Scope],Spread Soil[Scope])

Units: m3

Scope Task is Active C[Scope]=IF THEN ELSE (:NOT: Scope Task is Done C[Scope]:AND: Scope Task Start Flag C[Scope] > 0.5

Units: Dmnl

Scope Task is Done C[Scope]=IF THEN ELSE((Soil Compacted and ready for Quality Check[Scope]+Rework[Scope]+Final Work Completed[Scope])>= Spread Soil[Scope], 1, 0)

Units: Dmnl

b- Switch Equations that control the compaction flow

Scope Task Start Flag C[Rock11]=1

Scope Task Start Flag C[Granular12]=SAMPLE IF TRUE(Percentage of Compacted Soil[Rock11] >= Scope Start Percentage of Compaction[Granular12], 1 , 0)

Scope Task Start Flag C[Moraine13]=SAMPLE IF TRUE(Percentage of Compacted Soil[Granular12] >= Scope Start Percentage of Compaction[Moraine13], 1 , 0)

Scope Task Start Flag C[Rock21]=SAMPLE IF TRUE(Percentage of Compacted Soil[Moraine13] >= Scope Start Percentage of Compaction[Rock21], 1 , 0)

Scope Task Start Flag C[Granular22]=SAMPLE IF TRUE(Percentage of Compacted Soil[Rock21] >= Scope Start Percentage of Compaction [Granular22], 1 , 0)

Scope Task Start Flag C[Moraine23]=SAMPLE IF TRUE(Percentage of Compacted Soil[Granular22] >= Scope Start Percentage of Compaction [Moraine23], 1 , 0)

Scope Task Start Flag C[Rock31]=SAMPLE IF TRUE(Percentage of Compacted Soil[Moraine23] >= Scope Start Percentage of Compaction [Rock31], 1 , 0)

Scope Task Start Flag C[Granular32]=SAMPLE IF TRUE(Percentage of Compacted Soil[Rock31] >= Scope Start Percentage of Compaction [Granular32], 1 , 0)

Scope Task Start Flag C[Moraine33]=SAMPLE IF TRUE(Percentage of Compacted Soil[Granular32] >= Scope Start Percentage of Compaction [Moraine33], 1 , 0)

Units: Dmnl

The rate at which the work packages are compacted is formulated as follows:

Compaction Rate[Scope]=IF THEN ELSE(Scope Task is Active C[Scope],Max Compaction Rate[Scope],0)

Units: m3/hr

c- Final Work Done

Productivity Rate[Scope]+Rework rate[Scope],0)
Units: m3

Total Work Not Done=Total Scope-(Total Rework +Total Soil Compacted and Ready for Quality Check+"Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted")

"Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted"=SUM(Final Work Completed[Scope!])
Units: m3

Simulating the SD model based on the developed switch equations resulted in scheduling the nine operations as demonstrated in Figure 6.12. The figure shows an overlap of 50% between two successive operations. This represents the base case of the model as planned. Thus, this stage of the SD model development is accomplished successfully and validated. The next step involves loading the SD model with variables that are responsible for generating the real behavior of the earthmoving operations as discussed at the commencement of the SD model development.

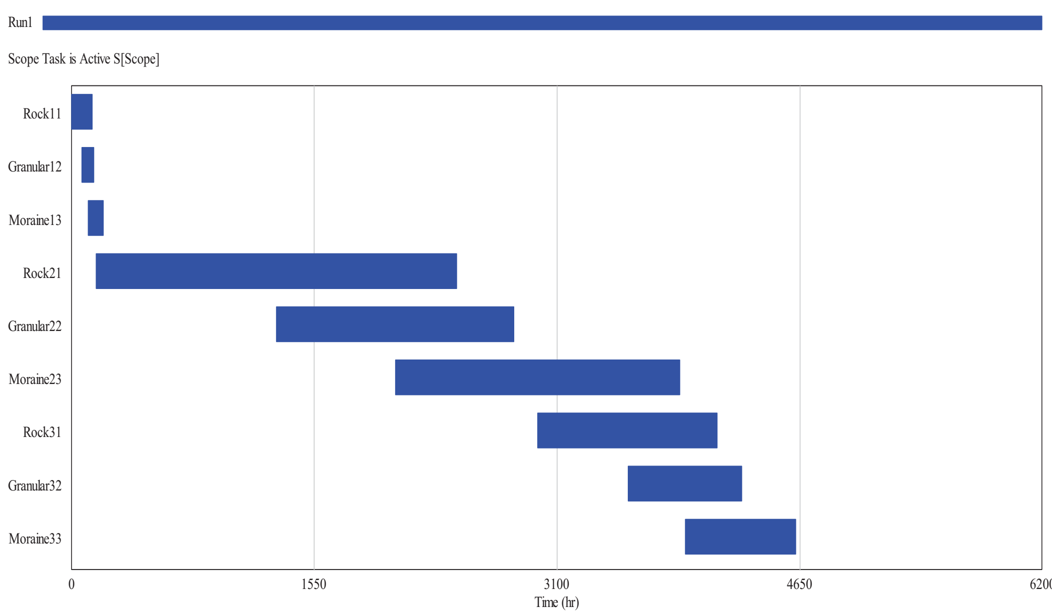


Figure 6.12 Generated Gantt chart for Backfilling Operations

B-<u>Forecasted Productivity and Schedule Pressure Module</u>

Productivity and schedule pressure are interrelated variables; both influence the project outcomes since they represent the link between perceived and actual parameters. The productivity mentioned here is the planned and forecasted, not the actual factor-impacted productivity. The real productivity is discussed in the coming sections. Management strategic targets (e.g., productivity and project completion duration) are molded and discussed in this loop structure as shown in Figure 6.13. These are not the only strategic targets; however, they represent the most essential ones. The total soil compacted and checked quality is divided by the cumulative simulation time to give an average of actual productivity. Project duration and planned productivity are set by manager at the start of the project. Planned productivity can be easily estimated by dividing the total work scope by the project duration. The outcome represents the management's desired level of productivity. Nevertheless, fluctuation of productivity is normal in construction due to dynamics, and can occur at any time during project execution. In this context, corrective actions to meet project targets can be taken based on the forecasted productivity. The forecasted productivity allows calculating the actual required completion duration for the project. The re-estimated completion duration is calculated based on the dynamics resulting in productivity fluctuation.

Schedule pressure percentage is calculated by dividing the actual time required to complete the project by the remaining time from planned schedule. Many aspects arise from the schedule pressure such as overtime required, impact on productivity, impact on quality and fatigue of operators. These aspects are computed in this structure and their effect is induced in the productivity and the quality parameters as illustrated in the figure above. The full equations of this structure are shown in Appendix B.

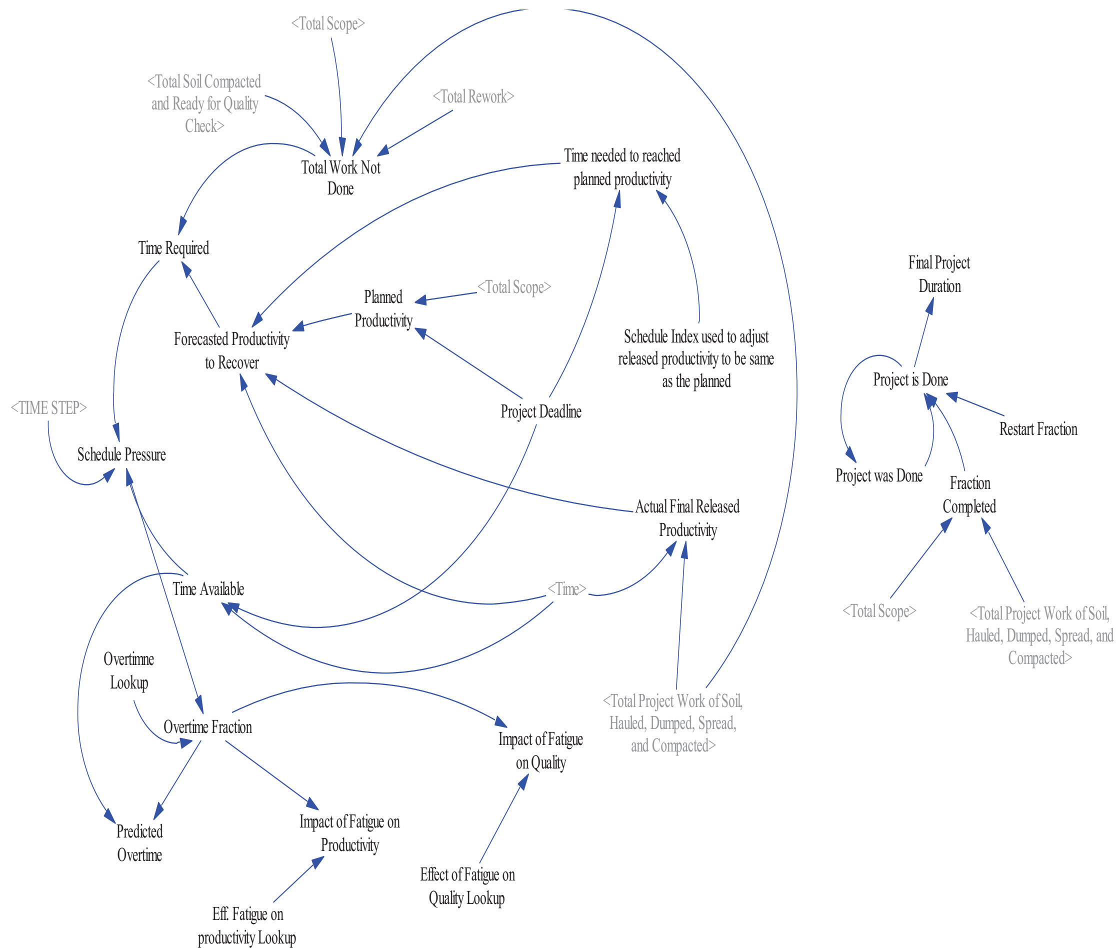


Figure 6.13 Productivity and Schedule Pressure

C-<u>Weather Impact Module</u>

Earthmoving projects are highly impacted by the weather condition. The explanation of the weather feedback process is a continuation to what was discussed in the identification of modeling units. Figure 6.14 demonstrates the dynamics that account for weather impact. As stated earlier, due to the weather condition, the project execution was planned from April to November, using two shifts with a total of 16 working hours per day. The total number of hours lost per month is calculated and prepared for the simulation model as shown in Table 6.6. “Start At Simulation Time” column shown in the table is the point in the simulation time where the delays is accounted for. For instance, in April there are 344 working hours and 68 working hours lost due rain. The point of time (80 hours) in the simulation length is selected arbitrary. This means starting from the point 80 hours the model will witness 68 hours delay until reaching 148 (80 hours plus 68 hours). Then starting from 147 hours the productivity computed by the simulation model will be normal and no effect of weather will be shown for the month of April.

In order to create an interruption in the project schedule, a subscript of events (e.g., event e1) that controls the interruption process is created in the SD model as follows:

Event: e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12,e13, e14, e15, e16, e17, e18, e19, e20, e21, e22, e 23, e24

Each event must correspond to a start point in the simulation time to create the interruption process, as shown in the following equations (e.g., event e1 occurs when simulation time reaches 80 hours).

Interruption Start Time in schedule[Event]=80, 400, 900, 1200, 1600, 1900, 2200, 2500, 2900, 3200, 3600, 4000, 4200, 4700, 5000, 5350, 5700, 5900, 6400, 6700, 7000, 7350, 7800, 8100

Is Interrupted[Event]=PULSE(Interruption Start Time in schedule[Event] , Total Interrupt Duration of Work in Hours for Single Month Reference Model[Event])

Units: Dmnl

Production is Interrupted=IF THEN ELSE(VMAX(Is Interrupted[Event!]) >0 ,1 ,0)

Units: Dmnl

Dumping Rate considering Rainfall effect[Scope]=IF THEN ELSE(Production is Interrupted , 0 , Max Dumping Rate[Scope])

Units: m3/hr

The above equations consider the weather impact on schedule and productivity. Only *Dumping Rate* variable equations were shown above. The other rates such as excavation, spreading and compaction have similar equations and shown in appendix B.

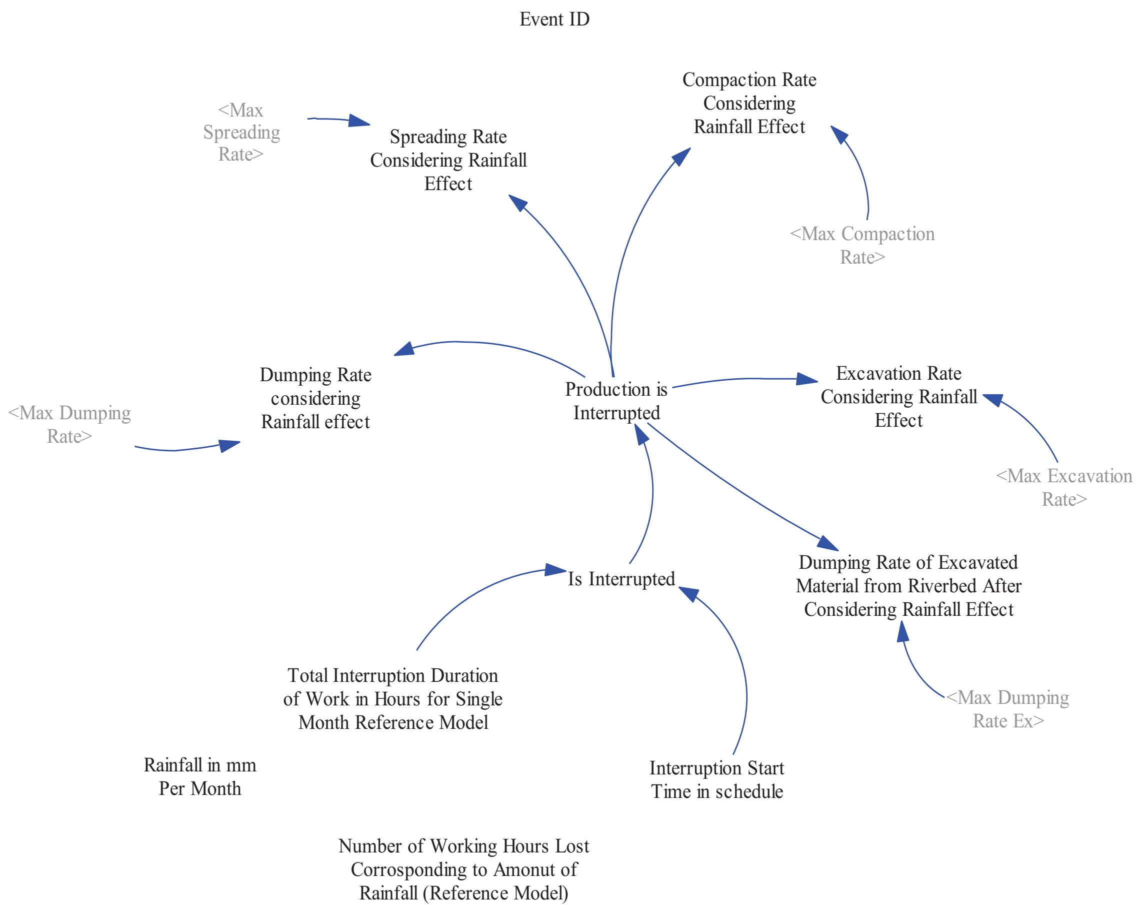


Figure 6.14 Weather Impact Module

Table 6.6 Hours Lost Due to Precipitation

S. No	Months	Working Hours Per Week for Two Shits	Working Hours Per Month	Total Cumulative Hours	Days lost	Hours Lost	Start At Simulation Time
1	April	80	344	344	8.25	66	80
2	May	80	344	688	9.36	74.88	400
3	June	80	344	1032	8.5	68	900
4	July	80	344	1376	9	72	1200
5	August	80	344	1720	8.5	68	1600
6	September	80	344	2064	8.25	66	1900
7	October	80	344	2408	10	80	2200
8	November	80	344	2752	9.6	76.8	2500
9	April	80	344	3096	8.25	66	2900
10	May	80	344	3440	9.36	74.88	3200
11	June	80	344	3784	8.5	68	3600
12	July	80	344	4128	9	72	4000
13	August	80	344	4472	8.5	68	4200
14	September	80	344	4816	8.25	66	4700
15	October	80	344	5160	10	80	5000
16	November	80	344	5504	9.6	76.8	5350
17	April	80	344	5848	8.25	66	5700
18	May	80	344	6192	9.36	74.88	5900
19	June	80	344	6536	8.5	68	6400
20	July	80	344	6880	9	72	6700
21	August	80	344	7224	8.5	68	7000
22	September	80	344	7568	8.25	66	7350
23	October	80	344	7912	10	80	7800
24	November	80	344	8256	9.6	76.8	8100

D-Impacted Productivity

This module computes the productivity impacted by the factors considered. The impacted productivity means the net productivity computed after considering the factors' influence on the operations within the model boundary. The feedback process of the impacted productivity is shown in Figure 6.15. The loss in productivity due to weather, depth of cut, road condition, operator skills, and equipment age are accounted for to calculate the net productivity. It should be mentioned that other factors such as schedule pressure and fatigue resulted from overtime are not involved in this feedback process, but accounted for in the Forecasted Productivity and Schedule Pressure module.

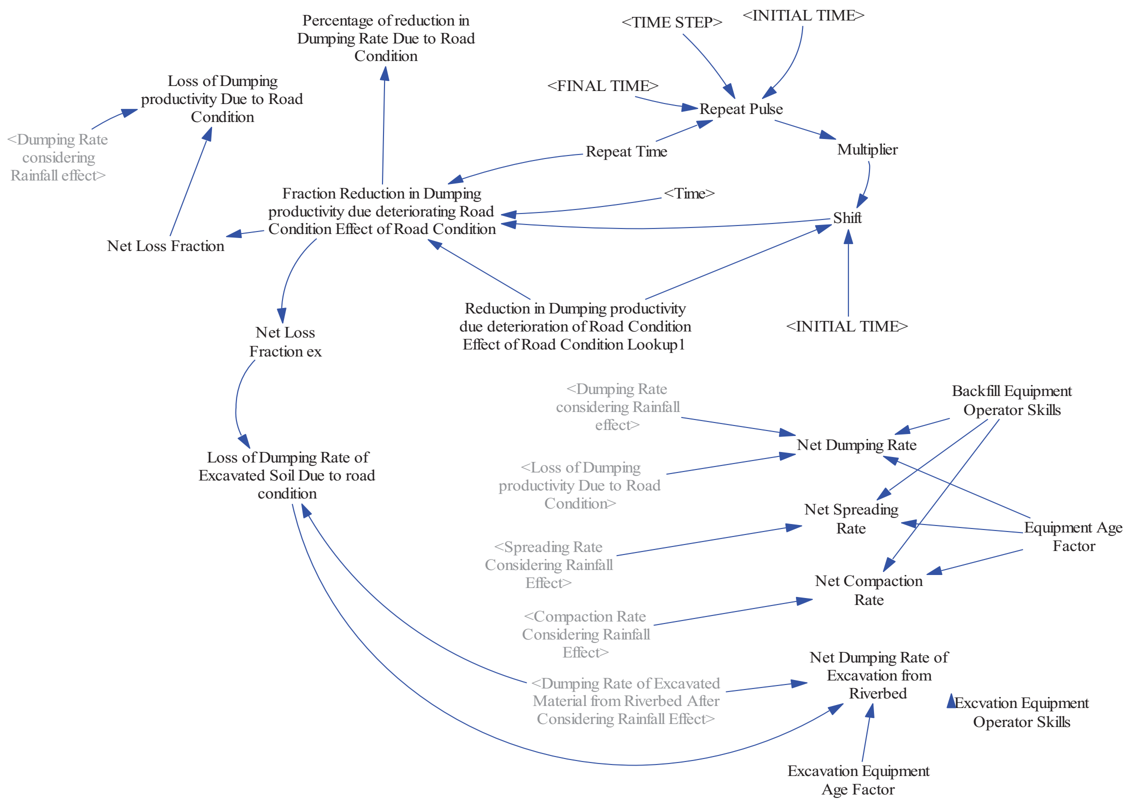


Figure 6.15 Impacted Productivity

E-<u>Cost Module</u>

The cost module shown in Figure 6.16 calculates the total project cost (direct and indirect), and predicts the needed funds to complete the project. Since four classes of workforce are involved heavily in the project, it is necessary to model these four classes using the subscript function. The subscript is defined as follows.

Workforce: Managers, Engineers, Equipment Operators, Labor

The module is composed of four stocks: 1) "Soil Cost"; 2) "Workforce Cost"; 3) "Equipment Cost"; and 4) "Overhead Cost." Each stock accumulates a specific cost category. The sum of all costs is accumulated in *"Total Accumulated Project Cost."*

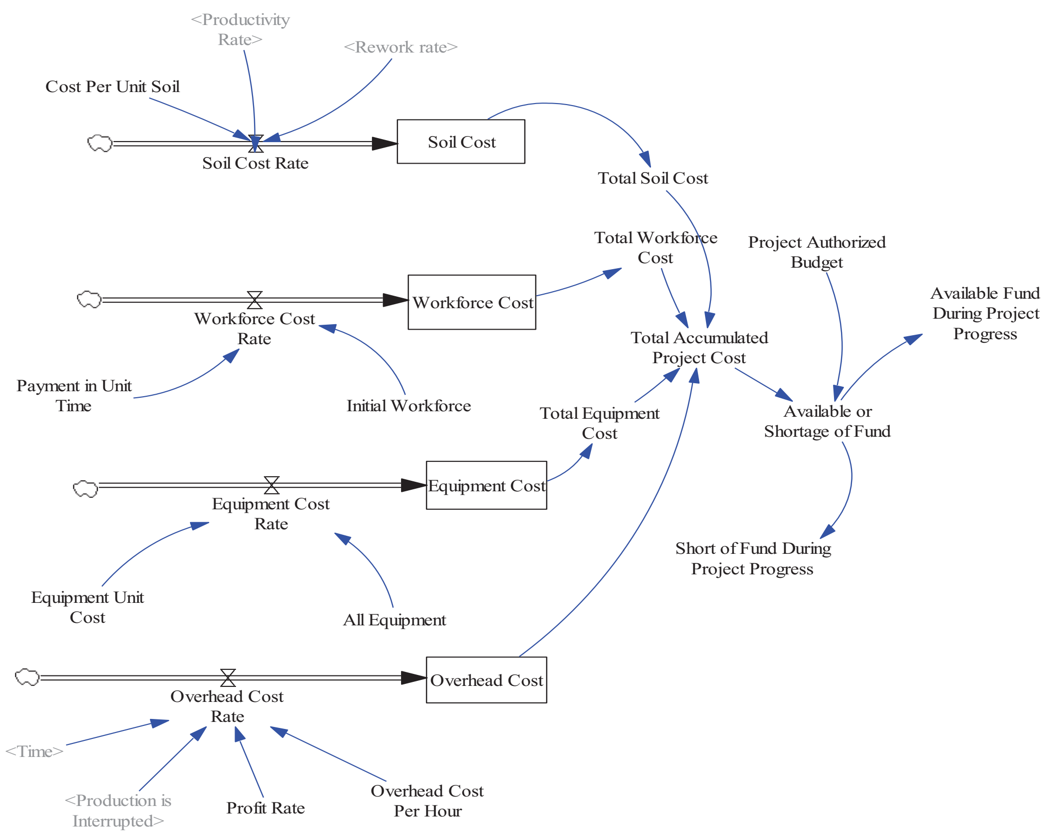


Figure 6.16 Project Cost Module

4- SD Model Testing and Validation

The first step in a normal modeling process is the issue of formulation. The essence of modeling relies heavily on its ability to represent the causal relationship in a real system (Law and Kelton 2000). Developing a credible conceptual model is a prerequisite to any validation endeavor. The SD model mainly depends on the conceptual understating of the system. The majority of policy models such as the SD type are developed for policy, testing different scenarios, and management purposes (Sterman 2000). The concern here is the inability of the SD model to mimic the real behavior of the system. Therefore, in order to assess the conceptual content of the SD model, it is imperative to examine the model and validate it. The first aspect that should be dealt with is to identify how the structure and policy generates behavior patterns. This allows the identification of the most appropriate structure(s) that mimic real behavior. It should be noted that identifying the most relevant structure is an iterative process and effort consuming. Sterman (2000) summarized a wide variety of specific standard tests to uncover flaws and improve SD models, as shown in Table 6.7. These standards were followed closely to validate the SD model of the earthmoving case study being used to develop the hybrid simulation model. The SD model was validated successfully and the results are shown in Table 6.7.

The developed SD model must be capable of generating base-case results similar to the results of the DES models, given that no project dynamics are included in the model. This means assuming 100% quality, relaxing project duration, no precipitation impact, and no impact of the other variables included in the model. The expected results of production rates and operation durations computed by the SD model should be identical to the results computed by the DES models. In order to show this, the nine soil backfill operations were simulated in the SD workflow module structure. The result of this testing process is shown in Table 6.8. The table presents a comparison between the duration of operations simulated by the DES models and the SD model. The ratio between the two durations of a single operation is approximately one. In Figure 6.17, outputs of certain parameters form the SD model are demonstrated, which shows that productivity is steady and follows a linear pattern. This indicates that SD successfully mimicked the outputs of the DES model, which points to a successful process of model validation.

Table 6.7 Summary of Applied Tests on Earthmoving SD model

Test Name	Purpose of the Test	Tools and Procedures
Boundary Adequacy	To ensure the model's purpose has been included in the model boundary	The model was developed based on project documents and records, industry practices, interviews with experts, literature reviews, and personal experiences. This has been done to check if a significant feedback process was neglected in the conceptual model. The model was tested at the sub-model and feedback loops levels, then the equations involved were checked to ensure the model boundary developed is respected.
Structure Assessment	To test whether the model structure is consistent with relevant descriptive knowledge of the system under consideration	The model evolved from well-proven model structures (Richarson and Pugh, 1981, Ford and Sterman, 1998, and Lyneis et al., 2001). Iterative procedures were followed to develop the model. Additionally, the structure was developed in accordance with theory describing the dynamics in earthmoving projects.
Dimensional Consistency	To ensure all variables in the model's equations are dimensionally consistent and correspond to the real system	The automated dimensional analysis tool available in Vensim (System dynamics simulation package) was used to perform unit checks. It was performed successfully. The meaning of the variables and their roles in the real system have been discussed.
Parameter Assessment	To check whether the parameters included in the model are consistent with relevant knowledge	This part involved two steps. The first was to use the available numerical data (e.g., 90 mm perception rate per day results in two days loss from project duration. The second was to make informed estimations of unavailable numerical data (e.g., adverse impact of road used by trucks is considered as deteriorating as of 16 hours from project commencement and continuing until scheduled maintenance at 80 hours).
Extreme Condition	To ensure the model behaves in a realistic manner despite being subjected to extreme values of equation inputs or policies.	The work quality was set at 100%, and the schedule was relaxed to prevent any schedule pressure effect. The simulated duration was similar to the one computed by the DES models. This is because DES does not consider the adverse impact of policies or context factors of the project in the modeling process.
Integration Error	To ensure an adequate time step was selected and applied.	Different time steps (e.g., 0.05, 0.1, 0.25, and 0.5) were set and the model was run based on these time steps. The model did not show any significant changes in the output based on different time steps.
Behavior Reproduction	To test whether the model can produce behavior of interest based on the same structure, and to see if the model is capable of generating various model behaviors	The output of the model variables was successfully compared with actual productivity variable behavior (e.g., S-curve) and real project progress.

Table 6.8 Comparison between DES and SD models duration Computations

Items	DES computed Duration (1)	SD computed Duration (base-case) (2)	Ratio (DES Duration/SD Duration)
Excavation of Riverbed	808.33	811	1.00
Rock11	138.44	139	1.00
Granular 12	77.5	78	0.99
Moraine 13	96.79	96	1.01
Rock21	2303.7	2302	1.00
Granular 22	1504.72	1509	1.00
Moraine 23	1814.44	1818	1.00
Rock31	1151.85	1151	1.00
Granular32	730.42	731	1.00
Moraine 33	707.63	708	1.00

Figure 6.17 Results of SD Model without Impact

6.4.3 Identifying Hybrid Model Structure and Interface Variables

The hybrid simulation structure is the arrangement between the DES and SD models that governs data mapping among models or variables. The process of data mapping is carried out based on a designed protocol as shown in Table 6.9. In Chapter 4, three possible hybrid structures were identified. The hybrid simulation model developed in this thesis uses the strategic-operation structure. The SD model acts as the main model in the hybrid simulation environment, while the DES models act as auxiliary gears in the global SD model. The rationale of selecting the strategic-operation structure is due to the problem under investigation and the objectives of the hybrid model. It is an issue to be judged by the modeler, depending on the problem on hand. The other important aspect at this stage is to identify the interface variables that will serve as linkages between the hybrid model variables (e.g., sender and receiver variables). The effective integration of the hybrid simulation models is achieved through a careful selection of these variables; otherwise, the hybrid simulation outcomes will be questionable.

Table 6.9 The Interface Variables Map in the Formalism Form

Sl. No.	Sender Module Type and Name m_{DESs}	Receiver Module Type and Name m_{SDr}	Interface Variable v_{inter}	Sender Variable in DES Module v_s	Receiver Variable in SD Module after Impacting the Variable v_r	Time Bucket Size (Hour)
1	DES_Exavation	SD_Excavation	Max_Excavation_Rate	COMBI_Excavate	Excavation Rate	1
			Max_Dumping_Rate_Ex	COMBI_Dump_Ex	Dumping Rate_Ex	1
2	DES_Rock11	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate11	COMBI_Dump_11	Dumping Rate[Scope11]	1
			Max_Spreading_Rate11	COMBI_Spread_11	Spreading Rate[Scope11]	
			Max_Compaction_Rate11	COMBI_Compcat_11	Compaction Rate[Scope11]	
3	DES_Granular12	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate12	COMBI_Dump12	Dumping Rate[Scope12]	1
			Max_Spreading_Rate12	COMBI_Spread_12	Spreading Rate[Scope12]	
			Max_Compaction_Rate12	COMBI_Compcat_12	Compaction Rate[Scope12]	
4	DES_Moraine13	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate13	COMBI_Dump_13	Dumping Rate[Scope13]	1
			Max_Spreading_Rate13	COMBI_Spread_13	Spreading Rate[Scope13]	
			Max_Compaction_Rate13	COMBI_Compcat_13	Compaction Rate[Scope13]	
5	DES_Rock21	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate21	COMBI_Dump_21	Dumping Rate[Scope21]	1
			Max_Spreading_Rate21	COMBI_Spread_21	Spreading Rate[Scope21]	
			Max_Compaction_Rate21	COMBI_Compcat_21	Compaction Rate[Scope21]	
6	DES_Granular22	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate22	COMBI_Dump22	Dumping Rate[Scope22]	1
			Max_Spreading_Rate22	COMBI_Spread_22	Spreading Rate[Scope22]	
			Max_Compaction_Rate22	COMBI_Compcat_22	Compaction Rate[Scope22]	
7	DES_Moraine23	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate23	COMBI_Dump23	Dumping Rate[Scope23]	1
			Max_Spreading_Rate23	COMBI_Spread_23	Spreading Rate[Scope23]	
			Max_Compaction_Rate23	COMBI_Compcat_23	Compaction Rate[Scope23]	
8	DES_Rock31	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate31	COMBI_Dump_31	Dumping Rate[Scope31]	1
			Max_Spreading_Rate31	COMBI_Spread_31	Spreading Rate[Scope31]	
			Max_Compaction_Rate31	COMBI_Compcat_31	Compaction Rate[Scope31]	
9	DES_Granular32	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate32	COMBI_Dump_32	Dumping Rate[Scope32]	1
			Max_Spreading_Rate32	COMBI_Spread_32	Spreading Rate[Scope32]	
			Max_Compaction_Rate32	COMBI_Compcat_32	Compaction Rate[Scope32]	
10	DES_Moraine33	SD (Dumping-Hauling, Spreading, & Compaction)	Max_Dumping_Rate33	COMBI_Dump_32	Dumping Rate[Scope33]	1
			Max_Spreading_Rate33	COMBI_Spread_32	Spreading Rate[Scope33]	
			Max_Compaction_Rate33	COMBI_Compcat_32	Compaction Rate[Scope33]	

m_{DESs}: sender simulation model, m_{SDr}: receiver simulation model, v_{inter}: interface variable, v_{ou} :sender variable, v_{in}: receiver variable.

The DES model developed for the riverbed excavation operation involved excavation and dumping processes. In the discussed SD model, there were two variables called *excavation rate* and *dumping rate*. These two variables represent productivity parameters at the micro level of the project. Alternatively, these two variables arise from the operational aspects of the project. Computing the value of these rates involves modeling tasks at the tactical level. SD modeling fails to calculate these rates, as illustrated earlier in the discussions. Therefore, these variables should be modeled using the DES model, and are called sender variables. Similarly, in the backfill operations, each soil type has a *dumping rate*, a *spreading rate*, and a *compaction rate*. These variables arise from the operational level of the backfill operation. Therefore, these variables will be computed by DES models. In total, since nine backfill operations are modeled, there are twenty-seven sender variables, in which each backfill operation has three sender variables. These twenty-seven sender variables have twenty-seven counterparts, receiver variables, in the SD model. The partnerships between senders and receivers are accomplished through interface variables. The interface variables are defined in the SD model. A map demonstrating the sender, receiver, and interface variables is given in Table 6.9. The shown protocol used to achieve the integration of the DES and SD models constitutes the hybrid simulation model.

In the developed hybrid simulation model, the interface variables are designed in such a way to receive data from DES sender variables and deliver these data to SD receiver variables. For example, the DES model for the excavation operation from the riverbed was utilized to compute the productivity of the excavation and dumping processes. These two variables named *COMBI_Excavate* and *COMBI_Dump_Ex* in the DES model are the data sender variables for receiver variables named *Excavation Rate* and *Dumping Rate_Ex* in the

SD model. The data was mapped through the interface variables *Max_Excavation_Rate* and *Max_Dumping_Rate_Ex* in the SD model.

Similarly, the DES models of the backfill operations were used to compute the productivity of dumping, spreading, and compaction processes. Since there are twenty-seven sender variables in the nine operations, only *DES_Rock11* is illustrated as an example. The three variables in the *DES_Rock11* model named *COMBI_Dump_11*, *COMBI_Spread_11* and *COMBI_Compcat_11* are the data sender variables for the receiver variables in the SD model named *Dumping Rate[Scope11]*, *Spreading Rate[Scope11]* and *Compaction Rate[Scope11]*. The data is transferred via the interface variables *Max_Dumping_Rate11*, *Max_Spreading_Rate11* and *Max_Compaction_Rate11* in the SD model.

6.4.4 Formalism and Synchronizing of Variables

The simulation models and their variables should be described in a format that can be easily understood and computed by the Executer. The developed hybrid simulation formalism in Chapter 4 requires defining module type, output variables (senders), input variables (receivers), interface variables, and the synchronization time function.

The developed formalism is applied to the hybrid simulation model, as demonstrated in Table 6.9. For instance, item number 1 shown in Table 6.9, the module type (m_t), is defined as SD_*Excavation* (m_{SD}). A question may arise here as to why the module definition is in the SD model and not the DES model. The direct answer is that all interfacing operations are conducted in the SD environment; in addition, the hybrid simulation model has an SD control structure. Therefore, the defined model type should be in the SD environment, not the DES environment. The interface variable *Max_Excavation_Rate* (v_{inter}) is defined in the *SD_Excavation* module. The sender or output variable that is imported from the DES module is defined as *COMBI_Excavate* (v_{ou}) and the receiver or input variable is defined as *Excavation Rate* (v_{in}). The rest of the items shown in the tables are incorporated in the formalism along the same procedure. The final step is to define the point in the simulation run-time to synchronize the variables based on the method proposed in Chapter 4. The time bucket was selected to be one hour; this means that every one hour, the interface variables in the SD model receive the outputs of the DES models and input these outputs to receiver variables in the SD model.

The SD model of workflow shown in Figure 6.18 demonstrates the interface variables in the red triangles. Each interface variable in the backfill operation module named *Max_Dumping_Rate*, *Max_Spreading_Rate* and *Max_Compaction_Rate"* has nine embedded variables controlled using the subscript function in Vensim. In the excavation operation module, two variables are defined as interface variables: *Max_Excavation_Rate* and *Max_Dumping_Rate_Ex*. For instance, the interface variable *Max_Dumping_Rate*, shown in the backfill module of Figure 6.19, receives *COMBI_Dump_11, COMBI_Dump12, COMBI_Dump_13, COMBI_Dump_21, COMBI_Dump22, COMBI_Dump23, COMBI_Dump_31, COMBI_Dump_32* and *COMBI_Dump_32* from the DES models. The final full loaded model with all variables is shown in Figure 6.19

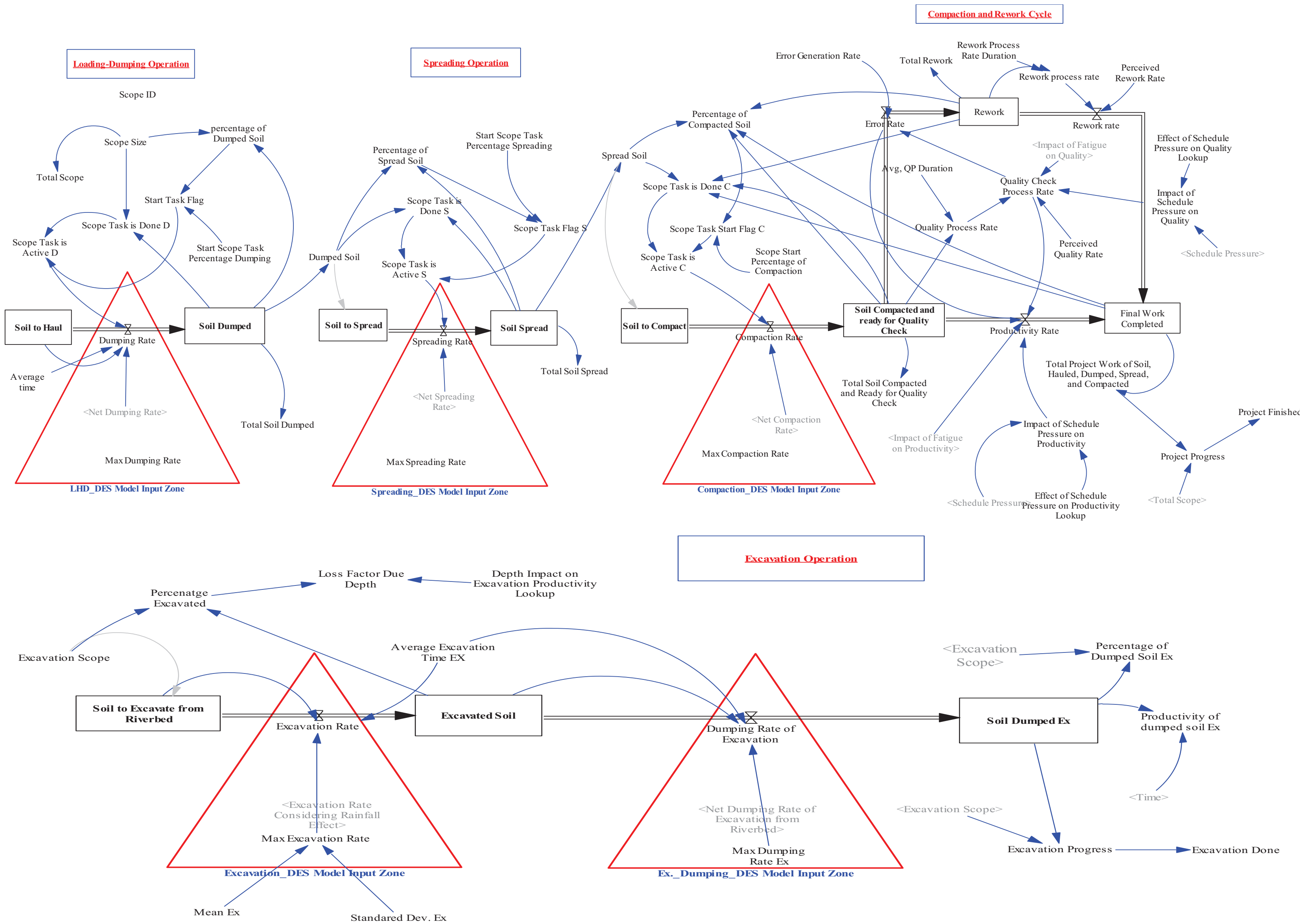


Figure 6.18 SD-Workflow Module with Interface Variables

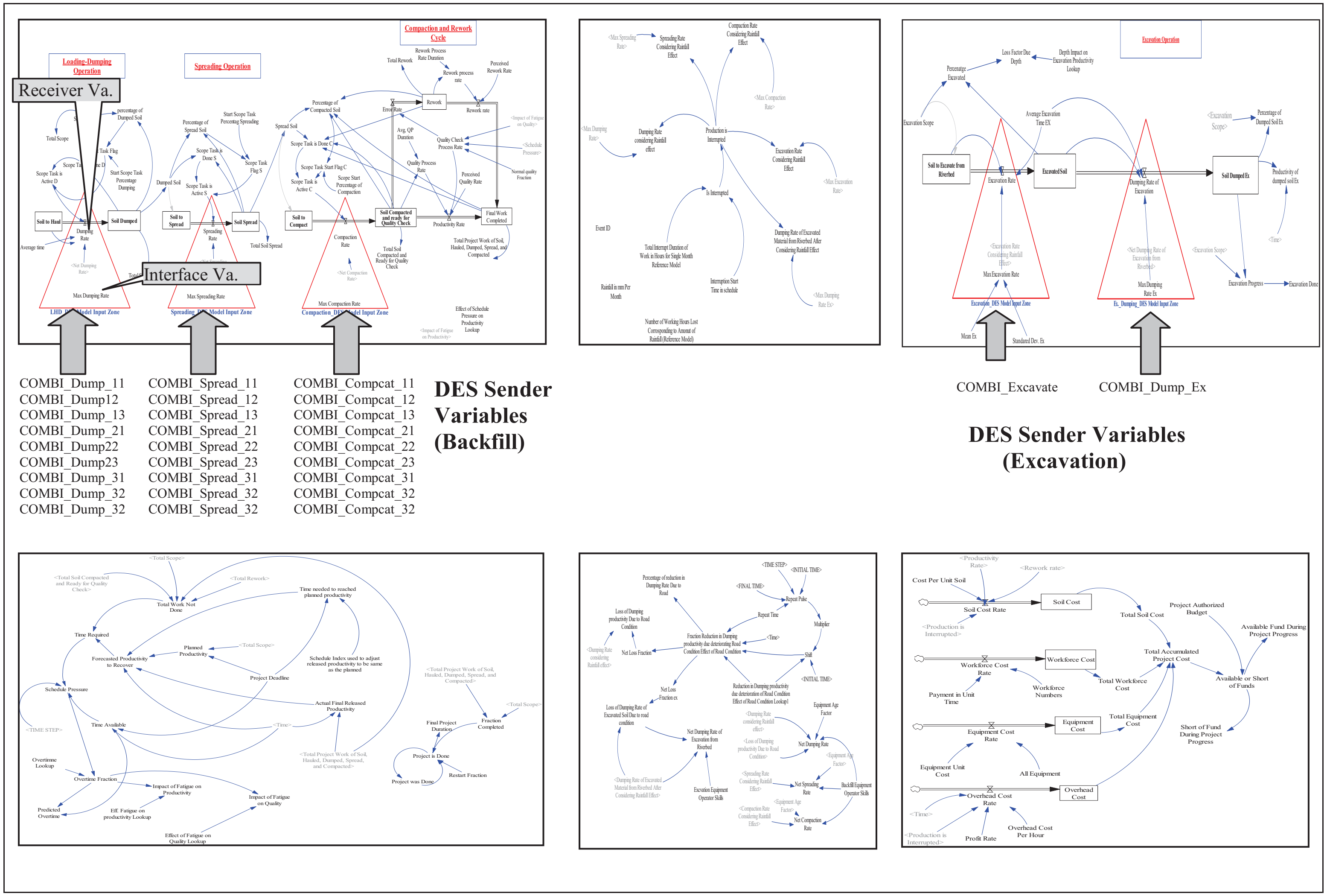


Figure 6.19 Fully Loaded Hybrid Simulation Model

6.5 Results and Analysis

Since the case study is composed of two separated operations (excavation and backfill), two different simulation models were developed. The results of the excavation and backfill were discussed and analyzed separately in this section. However, later on in the discussion, wherever is needed, the data are combined to have a global view of the project. The other reason to separate the analysis is that the results of the models cannot be mixed to avoid confusion. Prior to applying the feedback loops on the hybrid simulation model, it is essential to generate a base-case of the hybrid simulation model in order to compare the results of the different execution scenarios. The base case does not account for outstanding construction characteristics, feedback loops, and policies. Thus, it represents work execution in ideal situations. This scenario is similar to the results generated by the DES models. In the model testing stage, three scenarios were examined. A summary of the scenarios and their data is shown in Table 6.10. Scenario (A) represents the base case, while the other two scenarios address the project behavior under different factors and management policies.

Table 6.10 Project Execution Scenarios

	Scenario A	Scenario B (Key Policy Options)	Scenario C (Key Policy Options)
Backfill	-Backfill duration =4620 hours. -Quality Error = 0% -No weather impact. -No schedule pressure impact. -No overtime or fatigue impact. -No depth of cut impact. -No adverse road condition. -Operator skills =1 -Equipment age factor =1	-Backfill duration =4620 hours. -Quality Error =5% -Weather impact. -Schedule pressure impact. -Overtime and fatigue impact. -Adverse road condition. -Operator skills =0.96 -Equipment age factor =0.9 (middle age)	-Backfill duration =6680 hours. -Quality Error =5%. -Weather impact. -Schedule pressure impact. -Overtime and fatigue impact. -Adverse road condition. -Operator skills =0.96 -Equipment age factor=0.9 (middle age)
Excavation	-Excavation duration =811 hours. -No weather impact. -No schedule pressure impact. -No overtime and fatigue impact. -No depth of cut impact. -No adverse road condition. -Operator skills =1 -Equipment age factor =1	-Excavation duration=1145hours. -Weather impact. -Schedule pressure impact. -Overtime and fatigue impact. -Depth of cut impact. -Adverse road condition. -Operator skills =0.96 -Equipment age factor=0.9	N/A

6.5.1 Backfill Operation Results and Analysis

Hybrid Simulation Base Case (Scenario A)

Results generated from scenario A such as durations and productivities must be identical to the results generated by DES models. This is because the hybrid base case is not subjected to policies and influential factors. The benchmarking of the SD model to DES results has been successfully executed, as demonstrated in Figure 6.12 and Table 6.8. Part of the base case results are shown in Figure 6.20. The overall project completion duration of soil backfill as simulated by the DES model with 50% overlapping between the nine operations was 4616.56 hours (Table 6.8) while project completion duration generated by the hybrid simulation model was 4620 hours. This is can be considered a verification of the soundness of the model and the ability to generate simulated results similar to the DES models if policies and feedback loops are excluded from the modeling process. Figure 6.20 demonstrates the Gantt chart of the nine activities as they were planned, in addition to the individual productivity and accumulated productivity of all activities, schedule pressure, and overtime requirements. All simulated results depicted in Figure 6.20 show an ideal behavior of the simulation model. This allows moving to the next stage of verification that involves testing under dynamics and policies effects.

The base case generated by the hybrid simulation model demonstrates the process of planning and simulating construction projects using traditional methods (e.g., CPM and DES). Commonly, the traditional techniques simply describe the project as a top-to-bottom hierarchy through the decomposition of project elements into the smallest acceptable level, where work packages can be easily described by activities. Thereafter, cost, duration, and resources are estimated, mainly from experience, as deterministic numbers. Then, the project's job logic is described as a network of activities connected based on work sequence and logic.

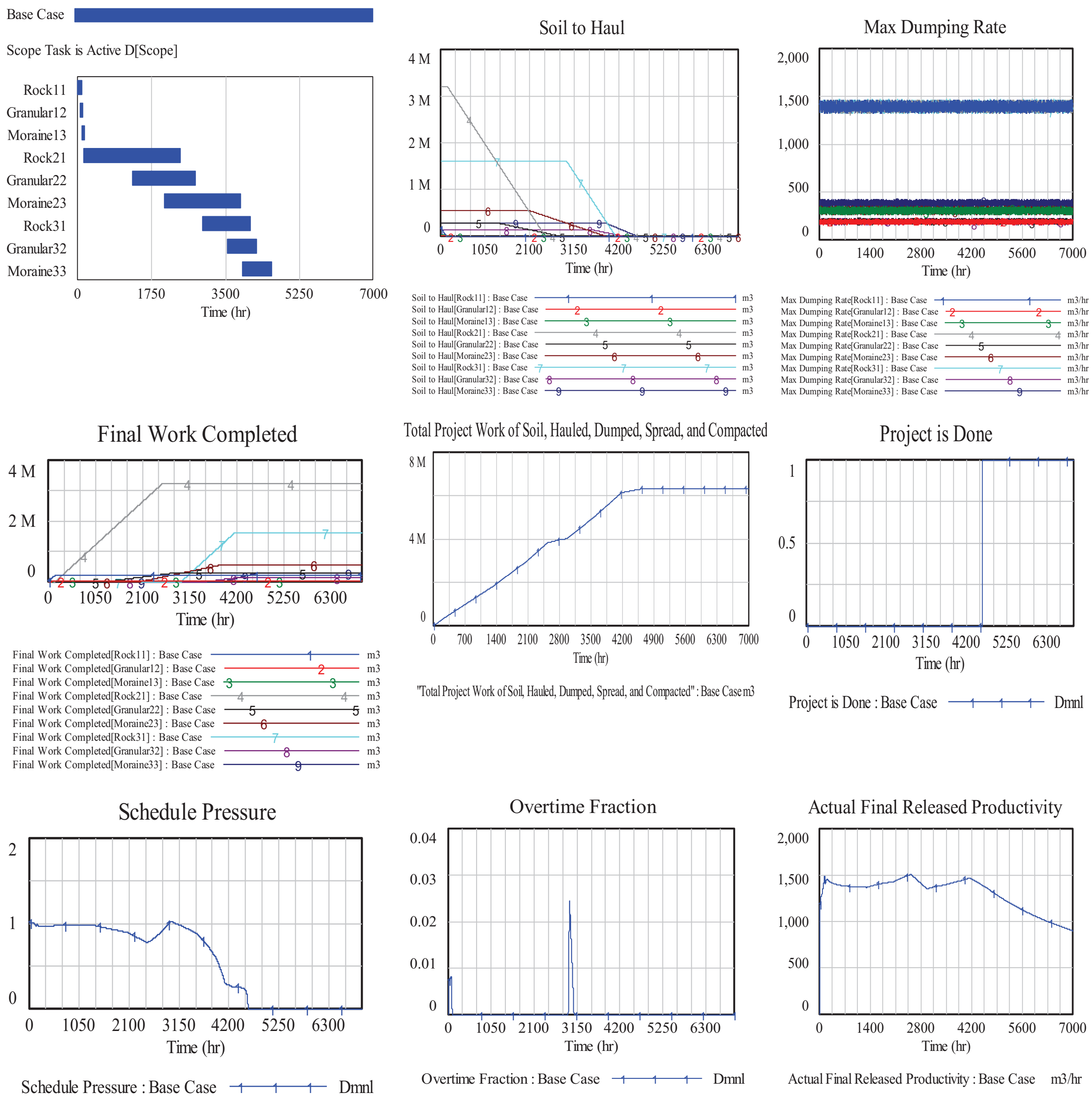


Figure 6.20 Base Case Results (Scenario A)

The apparent purpose of this process is to generate the actual project behavior. One of the main concerns in such a static and linear philosophy of addressing the issues of planning and controlling lies in the ability of the restructured activities of the network from bottom-to-top to behave based on the assumptions at the project decomposition stage, which does not happen in reality. Therefore, the missing link is explained through project dynamics.

Furthermore, management in reality behaves dynamically to adhere to project plans, and responds to new updates. These plans are targets or baselines of the management, and when these targets are endangered, actions are triggered to correct the drift of project behavior from targets. Thus, traditional methods are used as baselines implemented within a dynamic environment of causal-effect feedback loops. This shortcoming of the traditional methods is demonstrated through scenarios B and C.

Hybrid Simulation (Scenario B)

In scenario B, the hybrid simulation model of the earthmoving operations was simulated considering outstanding operation characteristics such as the feedback process and factors shown in Table 6.10. This scenario mimics what has happened in the real project implementation. The simulated project completion duration in scenario B became 6685 hours, as denoted by (a) and (d) in Figure 6.21, which represents a 44.6 % increase from the hybrid base case. The actual simulated productivity also dropped from approximately 1350 ton/hour in scenario A to 1000 ton/hour (35 % drop) as denoted by (f) in Figure 6.21. This difference in the project completion duration and actual released productivity could be explained by the feedback processes caused by a number of variables such as weather, skills, and others.

The fleet of equipment processing the “total soil to be hauled and dumped” is showing a productivity of dumping process as denoted by (b) in Figure 6.21. The pattern shows a fluctuation of productivity due to the influence of the feedback process. This rate represents the impacted dumping productivity. The final dumped soil pattern in the haul-dumping process is shown in (c). Similar pattern of behavior is noticed for spreading and compaction processes. Consequently, productivity fluctuation has extended the original project completion duration from 4620 hours in the ideal situation to 6685 hours in the current situation. Increasing the actual project completion duration beyond the planned duration due to the project's surrounding environment has increased the schedule pressure, as can be seen in the figure denoted by (e). This has in turn triggered the other predefined policies in the model to recover from the schedule slippage, such as extending the schedule beyond the planned completion date, hiring more workers and equipment, and using overtime. In this model, only the first option is valid as in the real project implementation, the management was already using the maximum equipment possible without creating site congestion, and a policy of overtime is also not applicable since two shifts were already used (16 working hours a day).

Follow-up reviews and an investigation of publications confirmed the obtained results. The model successfully predicted what really happened in execution. The earthmoving operations of the dam construction were designed to be completed in three years (with one year being equal to seven work months). This makes the total project completion duration 7224 hours. The hybrid simulation model predicted the total project duration as 6685 hours. The difference between the two figures was 7.46%. Therefore, it can be said that the hybrid simulation model could predict the project completion duration and productivity with an accuracy of 92.54%.

In order to investigate the sensitivity of certain factors, the weather impact has been neglected from the model. The new simulated completion duration became 5410 hours as illustrated in Figure 6.22, which represents a 17% increase in project completion duration from the base case (Scenario A), and a 19% reduction in project completion duration from the duration computed with inclusion of weather impact (Scenario B). The 17% increase is attributed to other factors included in the hybrid simulation model such as skills and schedule pressure.

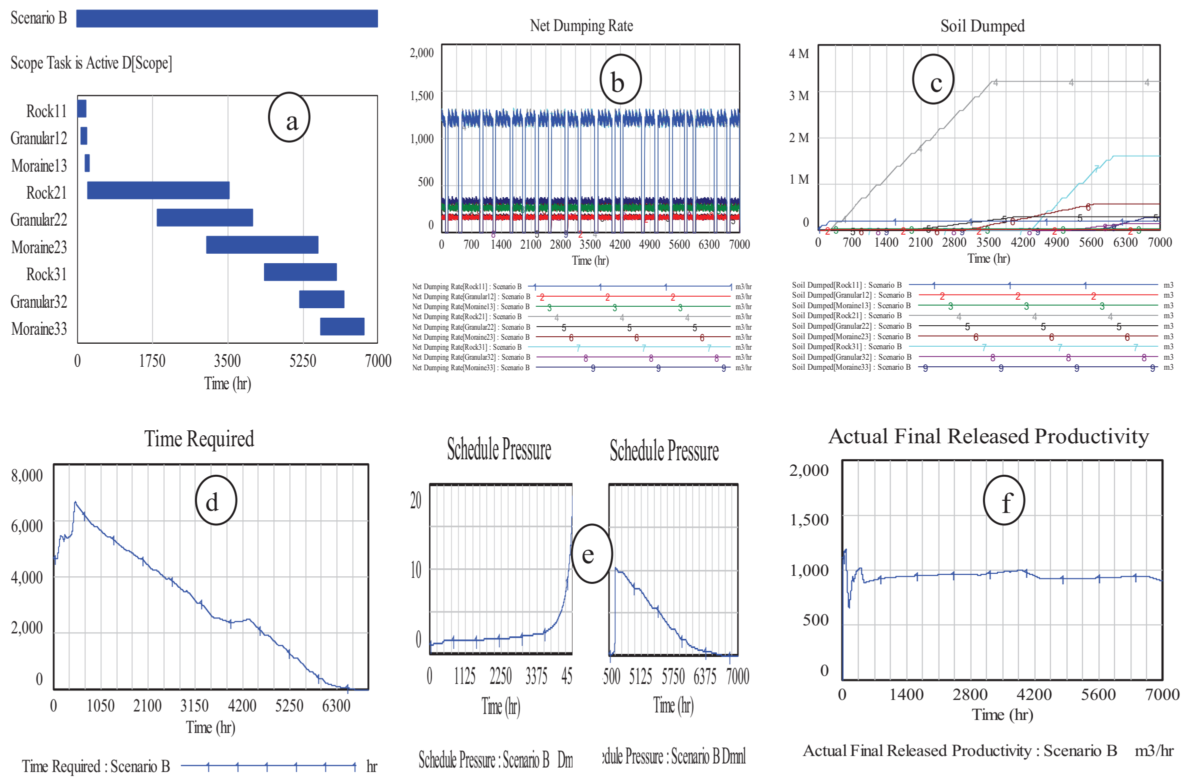


Figure 6.21 Scenario B Results

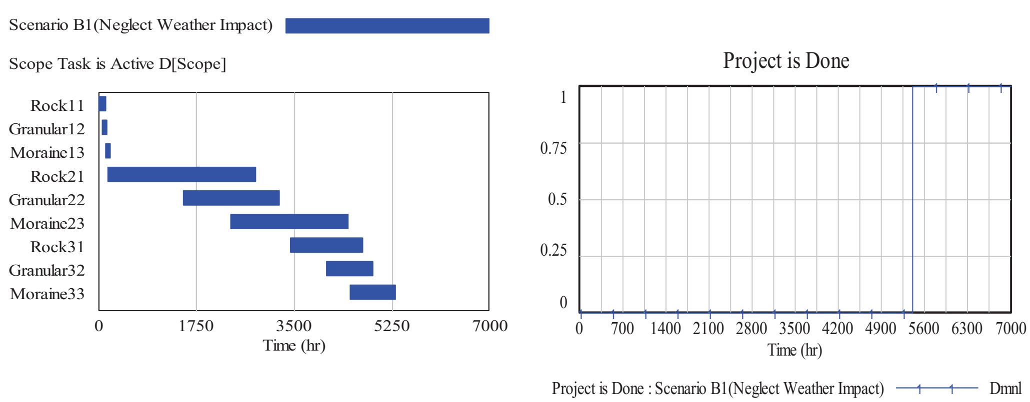


Figure 6.22 Simulated Results after Neglecting Weather Impact

Hybrid Simulation (Scenario C)

In scenario B, the schedule pressure and overtime were high (quality is deteriorating and rework is increasing). The planned project duration is set as 4620 hour (ideal time) while impacts of the factors surrounding the operations and reworks are reducing the fleet productivity, and hence increasing the actual project completion duration. In project execution scenario C, the planned project completion duration has been relaxed and set as actual project duration computed in scenario B (6685 hours). Relaxation is achieved by balancing the planned and actual project completion duration. The purpose of Scenario C is to show how the simulation model is affected under reduced project schedule pressure and overtime. Relaxing the project completion duration eventually result in planned productivity that approximately matches the fleet productivity as explained in the productivity causal-effect feedback loops. Realistic project completion duration has reduced the project schedule pressure, as demonstrated in Figure 6.23. Furthermore, Scenario C had resulted in a 5980 hour project completion duration, which represents 13.14 % reduction in the project completion duration computed by Scenario B. This 13.14% of project time saving results from safeguarding the crew and equipment from high schedule pressure, fatigue, quality deterioration, high rate of equipment breakdown and overtime. This is essential consideration for management in order to optimize the outcomes of the project parameters, as a better understating of how the dynamics are generated and how they affect the project will provide insight to addressing construction planning and management issues.

When comparing Scenario C with Scenario A, it was found that project completion duration has increased by 29% because of subjecting the model to the surrounding environmental effects (e.g., weather, equipment age and operator skills). In terms of actual hourly productivity, Scenario C is showing a better productivity than Scenario B with approximately 1190 ton/hour compared to approximately 1000 ton/hour. This represents a 19% difference in productivity.

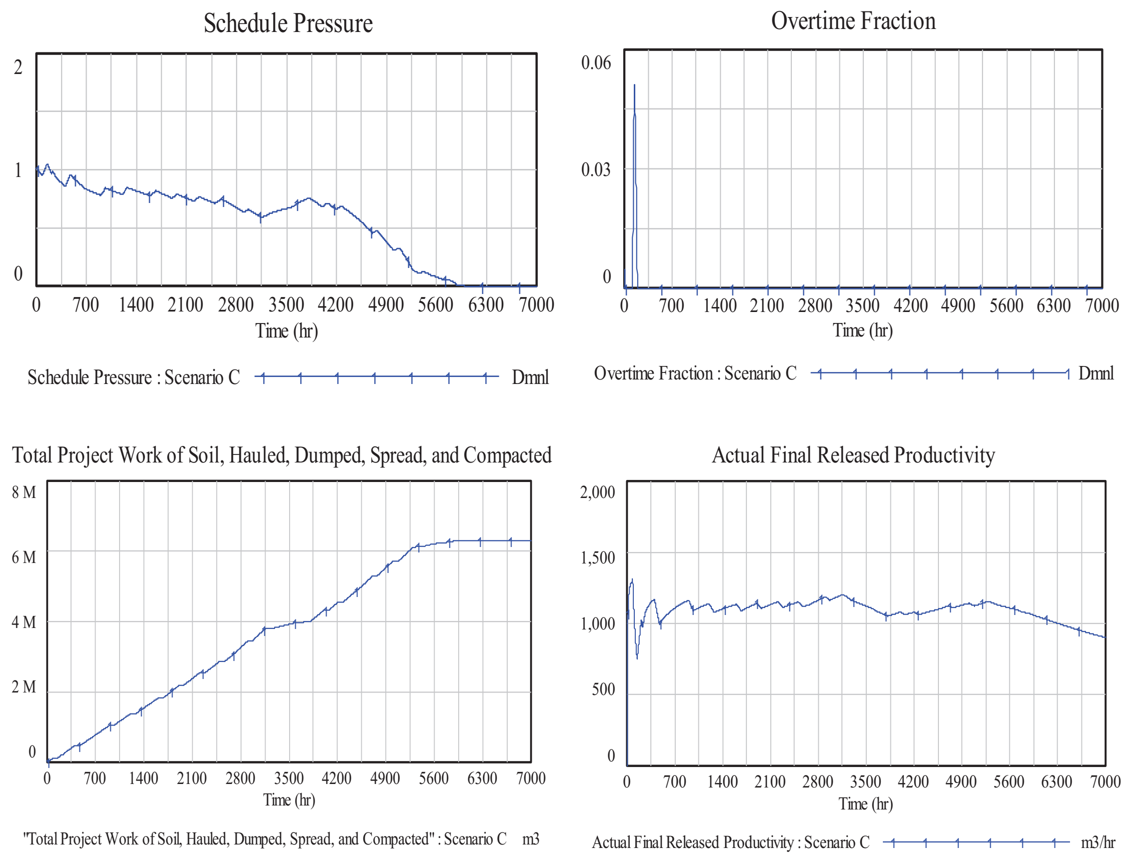


Figure 6.23 Scenario C Outputs

6.5.2 Excavation Operation Results and Analysis

The excavation operation was simulated using three scenarios, as shown in Table 6.10. The base case scenario for the excavation operation was developed based on the ideal situation where no impact of policies or the surrounding environment was considered. The simulated excavation completion duration was 811 hours, compared to 808.33 hours simulated by DES, as demonstrated in Figure 6.24. Thus, at this stage, the excavation part of the hybrid simulation model behaved similarly to the DES model. The next stage involved exposing the excavation hybrid model to policies, factors and feedback loops. The simulated project duration became 1145 hours, which represents a 29.10 % increase. Similarly, the simulated actual productivity has shown a drop of 25 % from the anticipated one in the ideal situation.

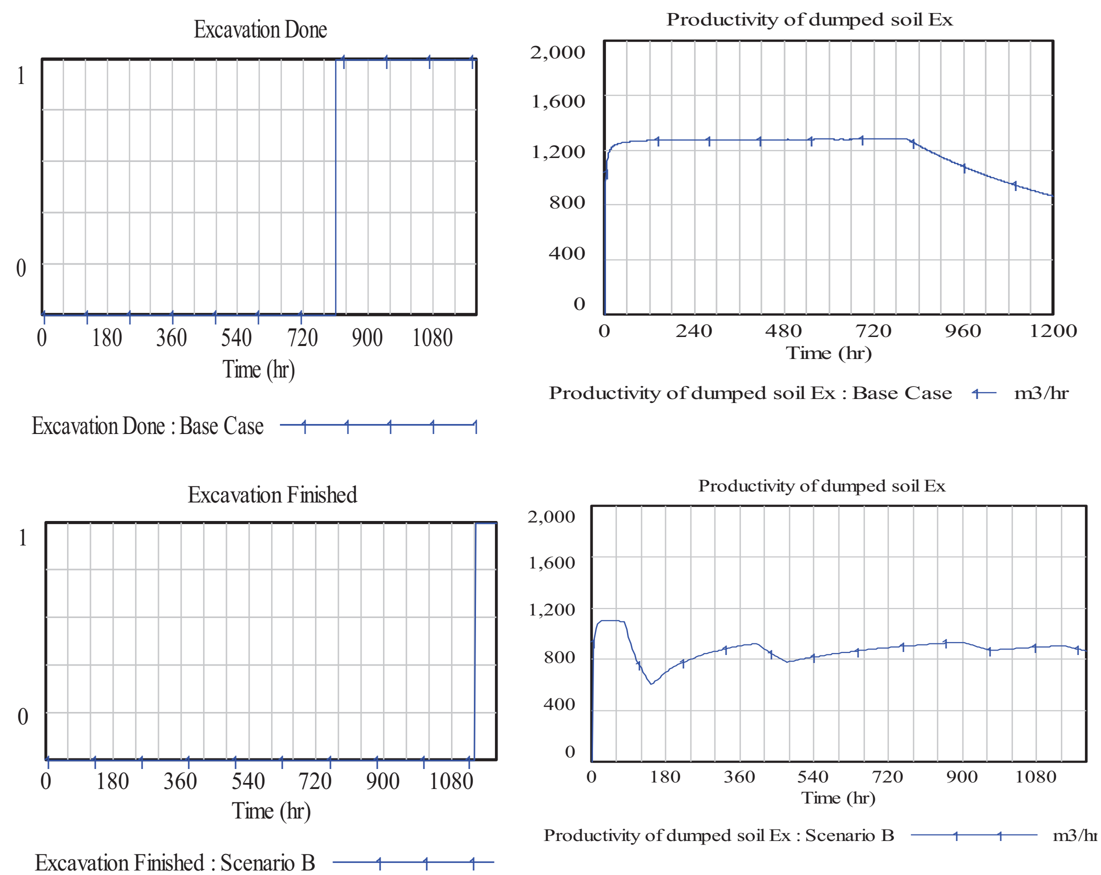


Figure 6.24 Simulated Outputs of Excavation Model

6.5.3 Added Value of Hybrid Simulation Modelling

The benefits of hybrid simulation in modeling and simulating construction operations have been discussed in the three implementation scenarios. The argument presented at the commencement of this thesis focused on developing simulation models using hybrid simulation approach that are capable of generating near-real behavior. This argument has previously been validated and emphasized in this section through a comparison between results generated using DES and SD models. Table 6.11 presents a comparison between completion durations for each operation computed by the DES model and the hybrid model scenarios B and C. The results show a significant difference between the duration computed by the DES model, where no impact of the surrounding conditions are considered, and the duration computed using the hybrid model that considers these impacts. The results of the hybrid model (Scenario B) were compared with the actual project data and were found to be 94 % precise. The second level of comparison of the simulation model results is conducted at the productivity level, as shown in Figure 6.25. The productivity of the DES model is compared with the productivity of the hybrid simulation model. The DES models always generate a higher productivity since they compute parameters considering no influence of the surrounding environment, and that is one of the main pitfalls of the DES simulation method. On the other hand, hybrid simulation computed productivity shows a fluctuation based on the occurrence of surrounding events. These events are captured as they occur and influence the workflow pattern as shown in the productivity figure.

Table 6.11 Comparison between DES Model and Hybrid Simulation Model Outputs

Operations	Simulated Durations by DES Model (Hours)	Simulated Durations by Hybrid Model Scenario B (Hours)	Simulated Durations by Hybrid Model Scenario C (Hours)
Excavation	808.33	1145	-
[Rock11]	138.44	226	207
[Granular12]	77	156	144
[Moraine13]	96.79	112	99
[Rock21]	2303.7	3309	2920
[Granular22]	1054.72	2244	1899
[Moraine23]	1814.44	2608	2271
[Rock31]	1151.85	1694	1461
[Granular32]	730.42	1065	970
[Moraine33]	707.63	1026	1022

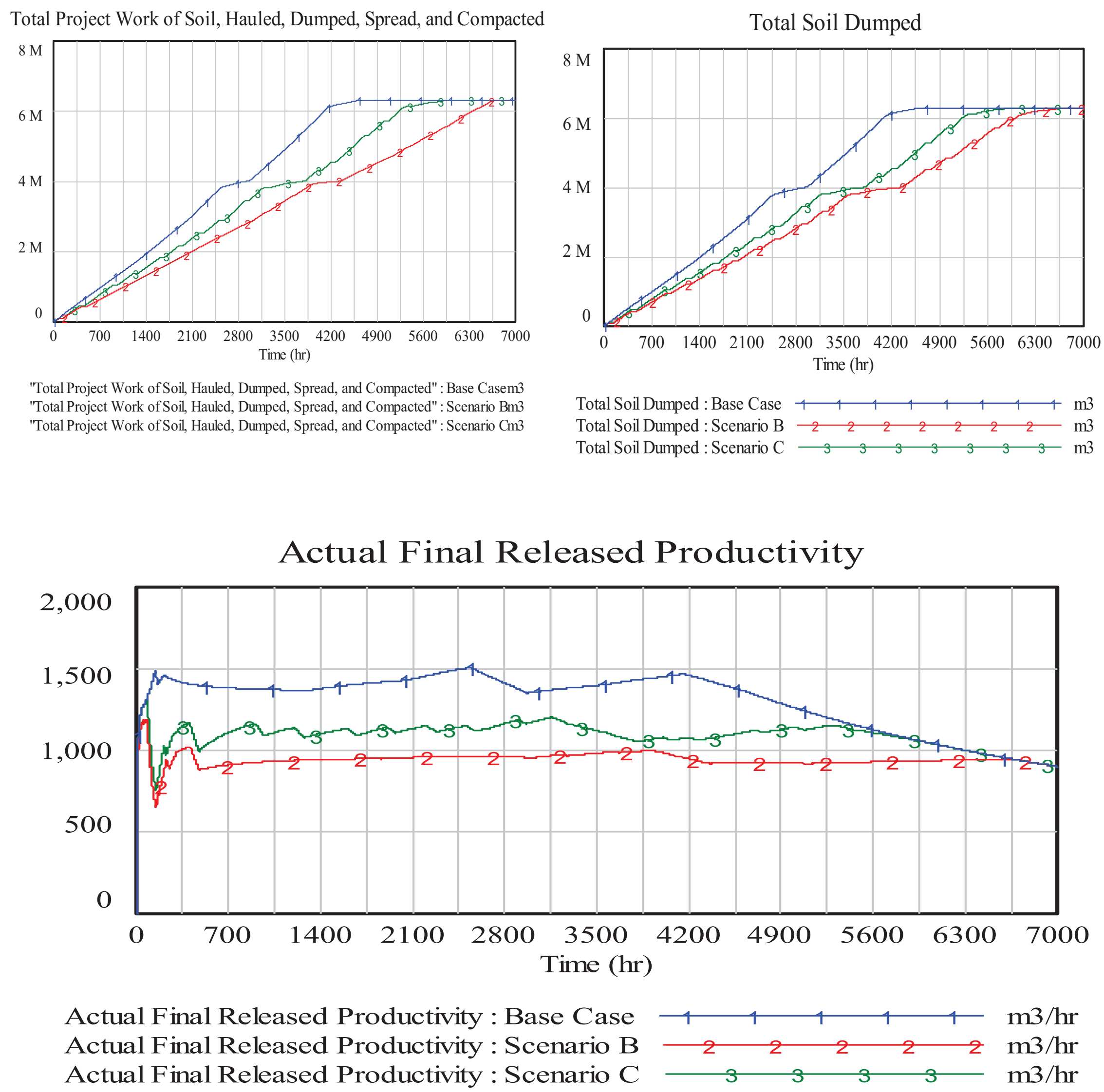


Figure 6.25 Productivity of Simulation Models

6.6 Chapter Summary

This chapter has discussed the stages of hybrid simulation development, validation of the model, and testing of the three scenarios. The operational aspects of the excavation and backfill operations were modeled using the DES method (EZStrobe software) with 50% overlapping between operations. The management policies and context aspects were modeled using the SD method. The feedback processes resulting from the interactions of the variables that are main causes of project dynamics was developed and validated. The validation process has been conducted using several approaches. Then, the SD model was developed using the Vensim software package and validated.

The next stage involved synchronizing the DES and SD models. The hybrid model was designed with the SD model in a controlling position, while the DES models were triggered by the SD to compute certain variables needed by the SD model (e.g., excavation rate, dumping rate, spreading rate and compaction rate). The synchronization protocol divided the variables involved in the synchronization process into sender variables initiated in the DES models, interface variables initiated in the SD model, and receiver variables initiated in the SD model. Twenty-nine variables were identified in the DES models as necessary sender variables to achieve the objectives of the hybrid simulation model. These variables are interfaced with another twenty-nine variables in the SD model that act as conveyors of data to the receiver variables.

Finally, the hybrid simulation model was tested using three scenarios that represented the ideal, extreme, and moderate scenarios. The output parameters of the simulation model such as productivity and project completion duration were illustrated and discussed. The simulation model showed a significant variation in outputs when subjected to altered management policy factors. The hybrid model, when subjected to the actual conditions surrounding the earthmoving operations, successfully predicted the project's actual duration with 92 % accuracy.

CHAPTER 7

AUTOMATED TOOL: HISIM

7.1 Overview

This chapter presents an automation tool developed to provide a hybrid simulation computation environment. The developed simulation tool is called HiSim, which stands for **H**ybrid **S**imulati**o**n application for construction proj**e**cts. The application was built based on the hybrid simulation method developed in Chapter 4. HiSim is an integration of a discrete event simulation language and continuous simulation languages (system dynamics) on a single computational platform. The integration process was developed using the Microsoft environment VB.NET. The Executer, one component of the hybrid simulation method, is responsible for controlling the integration process. The final section of this chapter demonstrates a case study to illustrate the application procedure and the validation process of HiSim.

7.2 HiSim Development

The hybrid simulation method developed in this thesis has been implemented through integrating different software packages such as STROBOSCOPE for DES model simulation, Vensim for SD model simulation, Microsoft Visio, and Microsoft Excel. The organization of the software packages involved is shown in Figure 7.1. The shown hybrid application architecture has four main components: 1) a graphical user interface (GUI); 2) a hybrid simulation computation; 3) databases; and 4) reporting. The GUI is the layer where the user first interacts with HiSim. It allows the user to create simulation models, export data and import data. The second layer involves the computation process of the hybrid simulation model. The components of the hybrid simulation developed in Chapter 4 are used in this layer. The third layer deals with storing and importing data from the software packages databases. The final layer specifies the types of reports that are necessary for the user. Further insight into the procedures of devolving a hybrid simulation model through interacting with HiSim is presented in the following subsections.

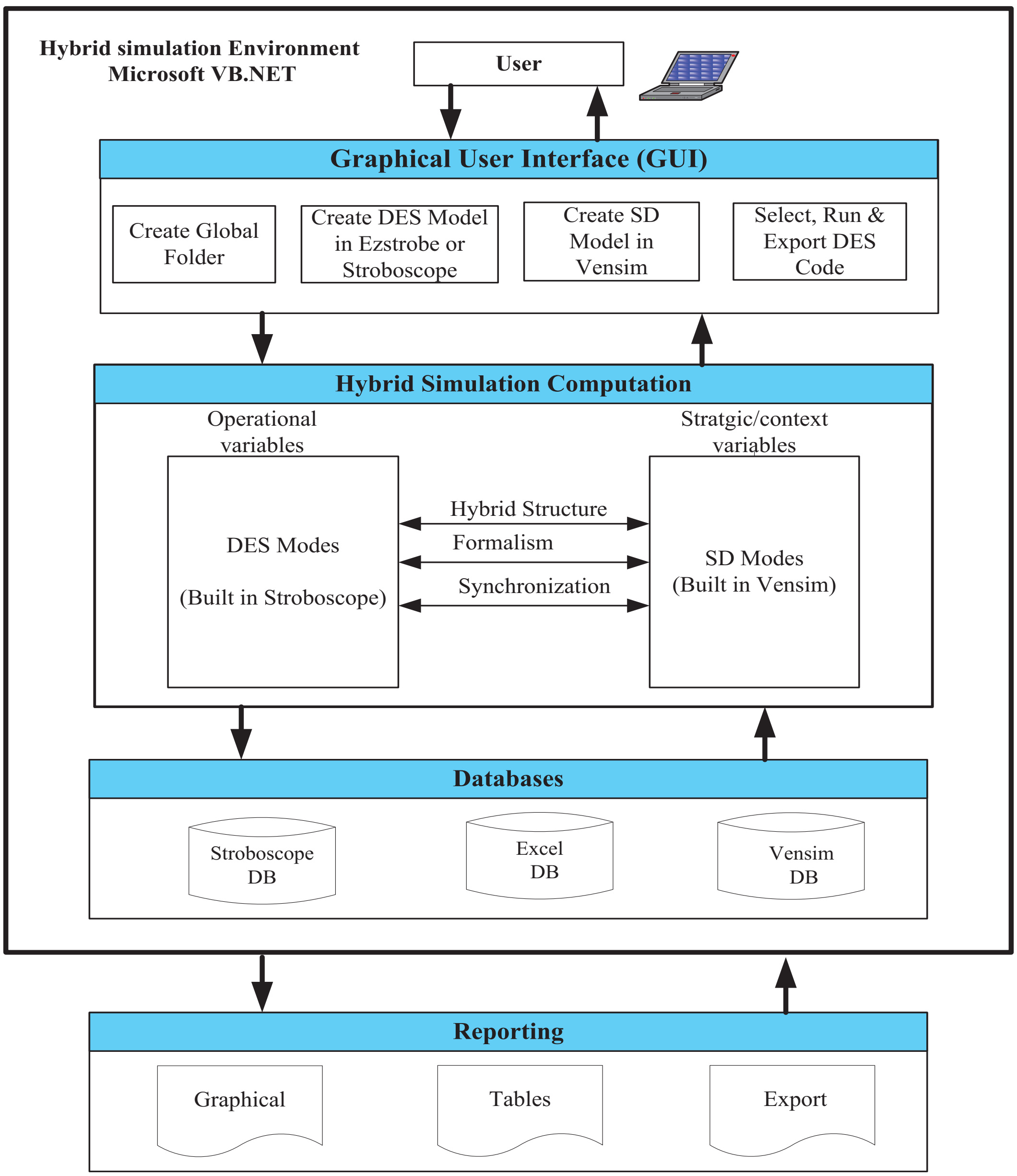


Figure 7.1 HiSim System Architecture

7.2.1 Graphical User Interface (GUI)

The GUI is the modeler's gate to interact with the hybrid simulation model. In order to create a model, the user of HiSim has to create a folder on the computer's hard drive. The folder is the central location of simulation model files. Misplacing any of these files causes the application not to function in the specified manner. The first stage is to create the DES simulation model. The user interacts with EZStrobe to create the DES model. Alternatively, the user can create the DES model in the STROBOSCOPE simulation language using coding format instead of the graphical. The second stage involves developing the SD model using Vensim. The user interacts with Vensim and develops the required model. The file has to be placed in the created folder for the hybrid model. The next stage involves running the simulation model and exporting the necessary files. The GUI gives the user a control over the simulation and supporting software packages. A snap shot of the GUI is shown in Figure 7.2. The following sections provide insight into the steps and the mechanism of interactions among the software packages.

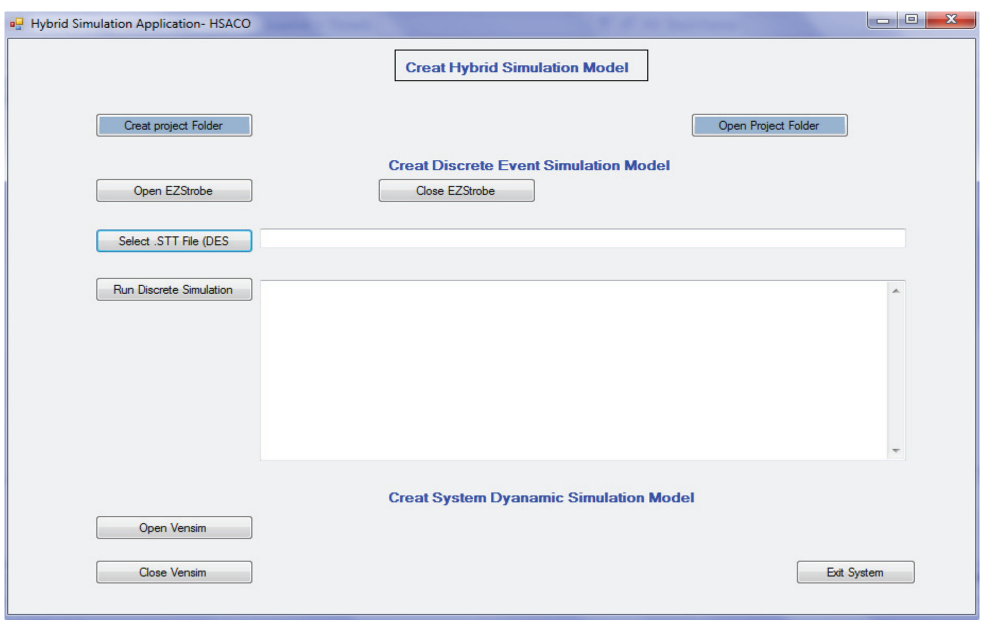
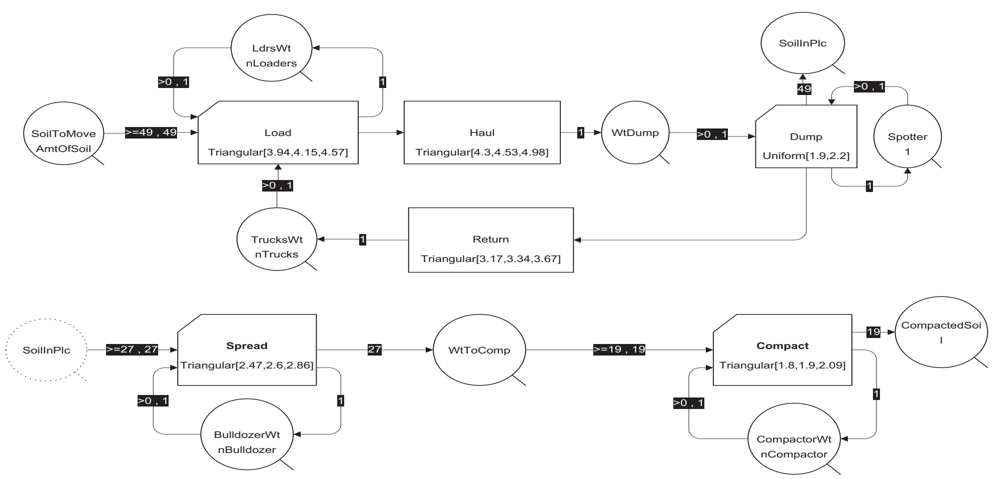


Figure 7.2 Snapshot of GUI in HiSim

7.2.2 Create DES Model

From GUI of HiSim, the user creates the folder that will contain all the simulation data files of the hybrid simulation model. The DES model is created in the EZStrobe graphical format or STROBOSCOPE code format. The variables (sender variables) that need to be computed by the discrete simulation model and exported to the SD are defined in the discrete simulation environment. The sender variables are defined in the equations format that is supported by EZStrobe. Figure 7.3 demonstrates a DES model for the soil backfill operations. The figure shows the graphic model and the equations used to compute the output parameters. The sender variables in the DES model are demonstrated in yellow in Figure 7.3.



nTrucks	Number of trucks	8
nLoaders	Number of loaders	2
AmtOfSoil	Amount of soil in m3	192700
nBulldozer	Number of Bulldozer	3
nCompactor	Number of CompactorWt	3

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

Figure 7.3 Defining Sender Variables in DES Model

When the process of the DES model is accomplished, the next step involves running the DES model from EZStrobe. It is important before running the DES model to select the option "Produce a trace output for debugging," as shown in Figure 7.4. This allows the STROBOSCOPE to convert the graphical DES model into a code format in the STROBOSCOPE simulation language. If the DES model is directly developed in the STROBOSCOPE code editor, then there is no need for this step. A sample for the code that is generated in STROBOSCOPE is shown in Figure 7.5. The code is stored in the hybrid simulation folder as an .STT file extension format. This code can be easily accessed and computed using STROBOSCOPE simulation language, and the process is shown in Figure 7.6.

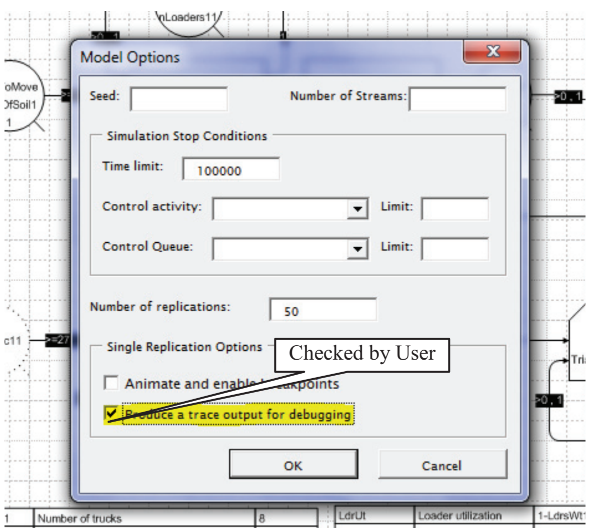


Figure 7.4 Converting the DES Graphical into STROBOSCOPE Code

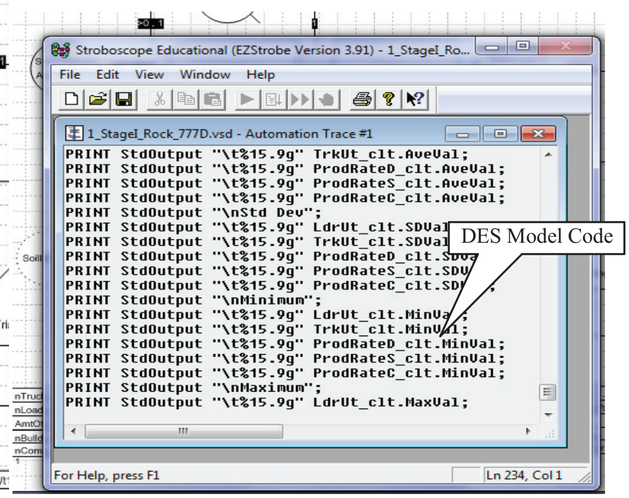


Figure 7.5 Simulation Code Generated in STROBOSCOPE Simulation Language

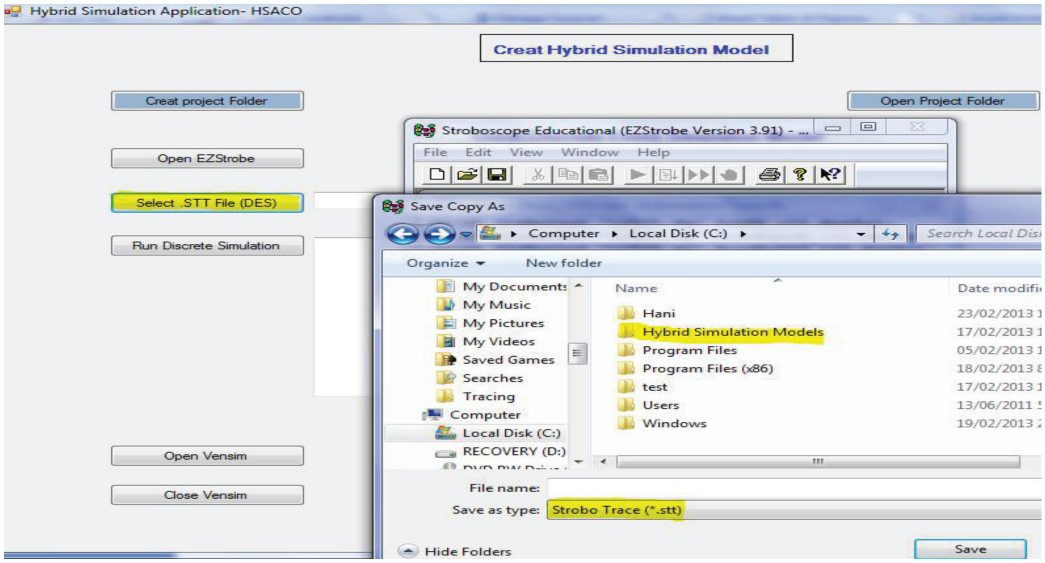


Figure 7.6 Storing DES Model in .STT Format

7.2.3 Executer: Computing and Exporting Outputs of DES Model

Now, the DES model is developed and the sender variables that will be used by SD are defined and known to the modeler. The next step involves importing the .STT file into the HiSim application and running it. To achieve this, a code is developed in the VB.NET environment. Initially, variables should be declared within the module as follows:

```
Public StroboApp As Object
    Public File_content As String
    Public variable_names(20) As String
    Public var_count As Integer
```

The function that calls and releases the STROBOSCOPE, `GetStrobo()`and `ReleaseStrobo()` respectively are coded as follow:

```
Public Function GetStrobo() As Object
        On Error Resume Next
        If (StroboApp Is Nothing) Then
            StroboApp = _
              CreateObject("Stroboscope.Document")
        End If
        GetStrobo = StroboApp
    End Function
```

```
&

'releases Stroboscope
    Public Sub ReleaseStrobo()
        StroboApp = Nothing
    End Sub
```

Now, the .STT file that contains the DES simulation is imported from its location using the following code:

```
Private Sub OpenFileDialog1_FileOk(sender As System.Object, e As
System.ComponentModel.CancelEventArgs) Handles
OpenFileDialog1.FileOk

        Dim FILE_NAME As String =
OpenFileDialog1.FileName.ToString()
        TextBox1.Text = FILE_NAME

        Dim objReader As New System.IO.StreamReader(FILE_NAME)
        Dim lineStr As String
        Dim i As Integer = 0

        lineStr = objReader.ReadLine
        Do Until lineStr Is Nothing
            If lineStr.Contains("COLLECTOR") Then
                Dim lineParts As String() = lineStr.Split(" ")
                variable_names(i) = lineParts(1).Substring(0,
lineParts(1).Length - 6)
                i += 1
            End If
            lineStr = objReader.ReadLine
        Loop

        var_count = i
        objReader.Close()

        Dim objReader2 As New System.IO.StreamReader(FILE_NAME)
        File_content = objReader2.ReadToEnd
        objReader2.Close()

        TextBox2.Text = File_content


    End Sub
```

The next and final stage involved is running the simulation code by calling STROBOSCOPE and then exporting the outputs into the Microsoft Excel Application as an .XLS file. The code used is as follows:

```
    Private Sub Run Discrete Simulation_Click(sender As System.Object, e As
System.EventArgs) Handles Run Discrete Simulation.Click

        Dim nResult As Integer
        Dim oExcel As Object
        Dim oBook As Object
        Dim oSheet As Object
        'On Error Resume Next
        'set name of client running Stroboscope

        oExcel = CreateObject("Excel.Application")
        oBook = oExcel.Workbooks.Add
        'Add data to cells of the first worksheet in the new workbook
        oSheet = oBook.Worksheets(1)
        oSheet.Range("A" & 1).Value = "Variable"
        oSheet.Range("B" & 1).Value = "Average"
        oSheet.Range("C" & 1).Value = "Standard Dev"
        oSheet.Range("D" & 1).Value = "Minimum"
        oSheet.Range("E" & 1).Value = "Maximum"

        GetStrobo()
        StroboApp.ClientVersion("test Strobo")

        nResult = StroboApp.RunStatements(File_content)

        For i = 0 To var_count - 1
            oSheet.Range("A" & i + 2).Value = variable_names(i)
            oSheet.Range("B" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.AveVal")
            oSheet.Range("C" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.SDVal")
            oSheet.Range("D" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.MinVal")
            oSheet.Range("E" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.MaxVal")
        Next

        StroboApp.CloseAllOutputs()
        StroboApp.EndModel()

        StroboApp.SetAskRelease(False)

        Dim filename As String = Microsoft.VisualBasic.DateAndTime.Day(Now) & "_"
& Month(Now) & "_" & Year(Now) & "$" & Hour(Now) & "_" & Minute(Now) & "_" &
Second(Now)

        'Save the Workbook and Quit Excel

        oBook.SaveAs("C:\Hybrid Simulation Models\" & filename & "result.xls")
        oExcel.Quit()
```

```
        ReleaseStrobo()
    End Sub
```

After the .STT file is selected, the code of the DES model is imported into the dialogue box. Now, the DES model code is ready to be run and computed. The instance of loading the DES model code is shown in Figure 7.7. When the simulation run is finished, the results are stored in a Microsoft Excel file, as shown in Figure 7.8, and exported directly to the hybrid simulation folder, where it is interfaced with the interface variables at every time step of the simulation length of the SD model.

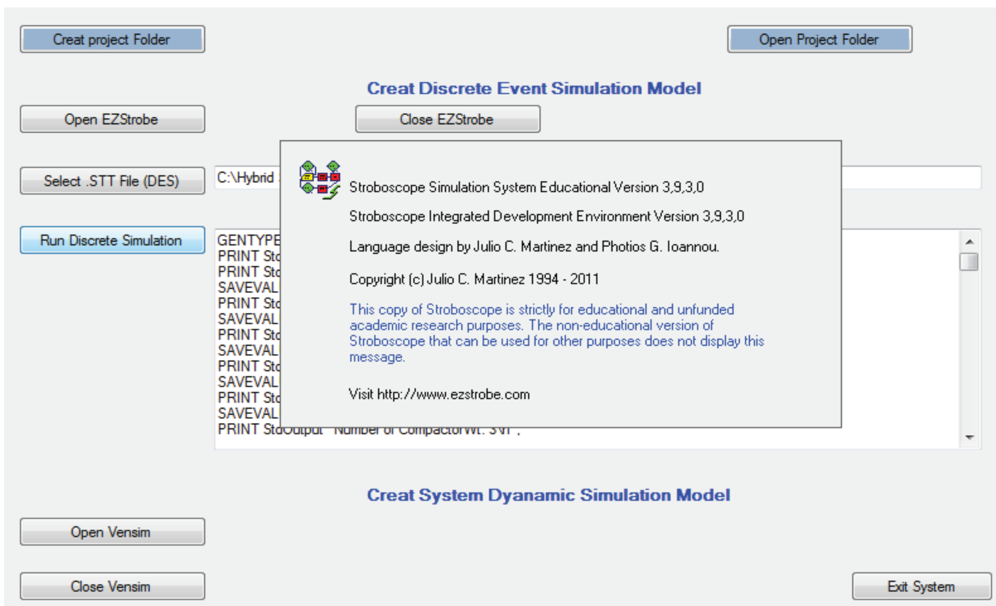
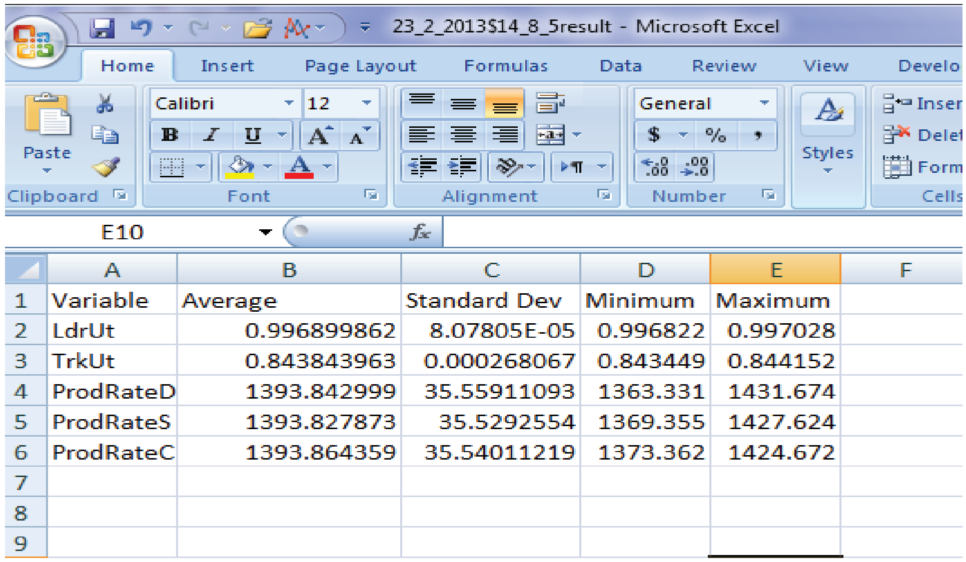


Figure 7.7 Loading DES Code and Running Simulation



	A	B	C	D	E	F
1	Variable	Average	Standard Dev	Minimum	Maximum	
2	LdrUt	0.996899862	8.07805E-05	0.996822	0.997028	
3	TrkUt	0.843843963	0.000268067	0.843449	0.844152	
4	ProdRateD	1393.842999	35.55911093	1363.331	1431.674	
5	ProdRateS	1393.827873	35.5292554	1369.355	1427.624	
6	ProdRateC	1393.864359	35.54011219	1373.362	1424.672	
7						
8						
9						

Figure 7.8 DES Model Results Stored in Microsoft Excel

7.2.4 Create SD model

The SD Model is created using Vensim. Several functions are embedded in Vensim such as subscript control, data export and data import, which allows interfacing and mapping of variables of different simulation models. The interface variables are defined in the Vensim environment. The DES models' sender variables are called by the interface variables and exported from that point to the receiver variables that were defined in the SD model.

7.2.5 Reporting

The hybrid simulation application is capable of reporting the simulation results in tabular, graphical and spreadsheet format. This makes the analysis an easy and simple task. The powerful reporting capabilities of Vensim are utilized to publish the simulation model results.

7.3 Applying the Developed Hybrid Simulation Application on a Case Study

A hybrid simulation model was developed as part of the process of testing HiSim. The case study data presented in Chapter 5 were used in the model development. The steps of developing a hybrid simulation model using HiSim are shown in Figure 7.9. In Step 1, the file folder of the hybrid simulation model is defined. The next stage in Step 2 involves developing the DES model using EZStrobe and defining the sender variables in the DES model. In total, ten DES models were developed for the excavation and backfilling operations. The ten DES models were responsible for computing twenty-nine sender variables such as excavation rate (one variable), dumping of excavated soil rate (one variable), backfill dumping rate (nine variables), backfill spreading rate (nine variables), and backfill compaction rate (nine variables). The values of these variables at each time bucket (e.g., 1 hour) were calculated. Step 3 involves running the DES model to generate the STROBOSCOPE Code in an .STT file format. All graphical DES models developed using EZStobe were converted into STROBOSCOPE simulation code as shown in Step 4, and stored in the hybrid simulation folder as an .STT file. In Step 5, the DES .STT file is called and imported into HiSim; thereafter, the DES model is run and computed, which is Step 6. HiSim then exports the outputs (the operational variables) of the DES code into an Excel spreadsheet, as shown in Step 7. The DES simulation process is thus accomplished, and the variables are ready to be synchronized.

The process of developing the SD model starts at Step 8. Stocks, flows, and auxiliary variables were developed based on the case study feedback process. The process of developing the SD model involved defining the receiver variables and interface variables. This step primarily depends on the SD model structure and the purpose of the hybrid simulation model. However, usually the receiver variables should be variables that the SD model is limited in its capability of computing. This is the purpose behind deploying hybrid simulation. In the illustrative case study used, the excavation rate and excavated soil dumping rate are defined as receiver variables in the excavation simulation model and the rates of dumping, spreading, and compacting are defined as receiver variables in the backfill model. The last issue is to establish the data conveyor (interface variables) between the sender and receiver variables. The interface variables are defined in the SD model to establish the link between the sender and receiver. The number of each of the sender, receiver, and interface variables should be equal. Otherwise, an error message pops up.

At this point, all the elements necessary to start the synchronization of DES and SD model variables are developed. The synchronization operation starts by defining the time bucket size. A time bucket of one-hour size was selected and defined in the Vensim settings. This means that at the end of each one-hour time period, HiSim will request that the values of the sender variables from the DES models be sent to the interface variables, then on to receiver variables in the SD model.

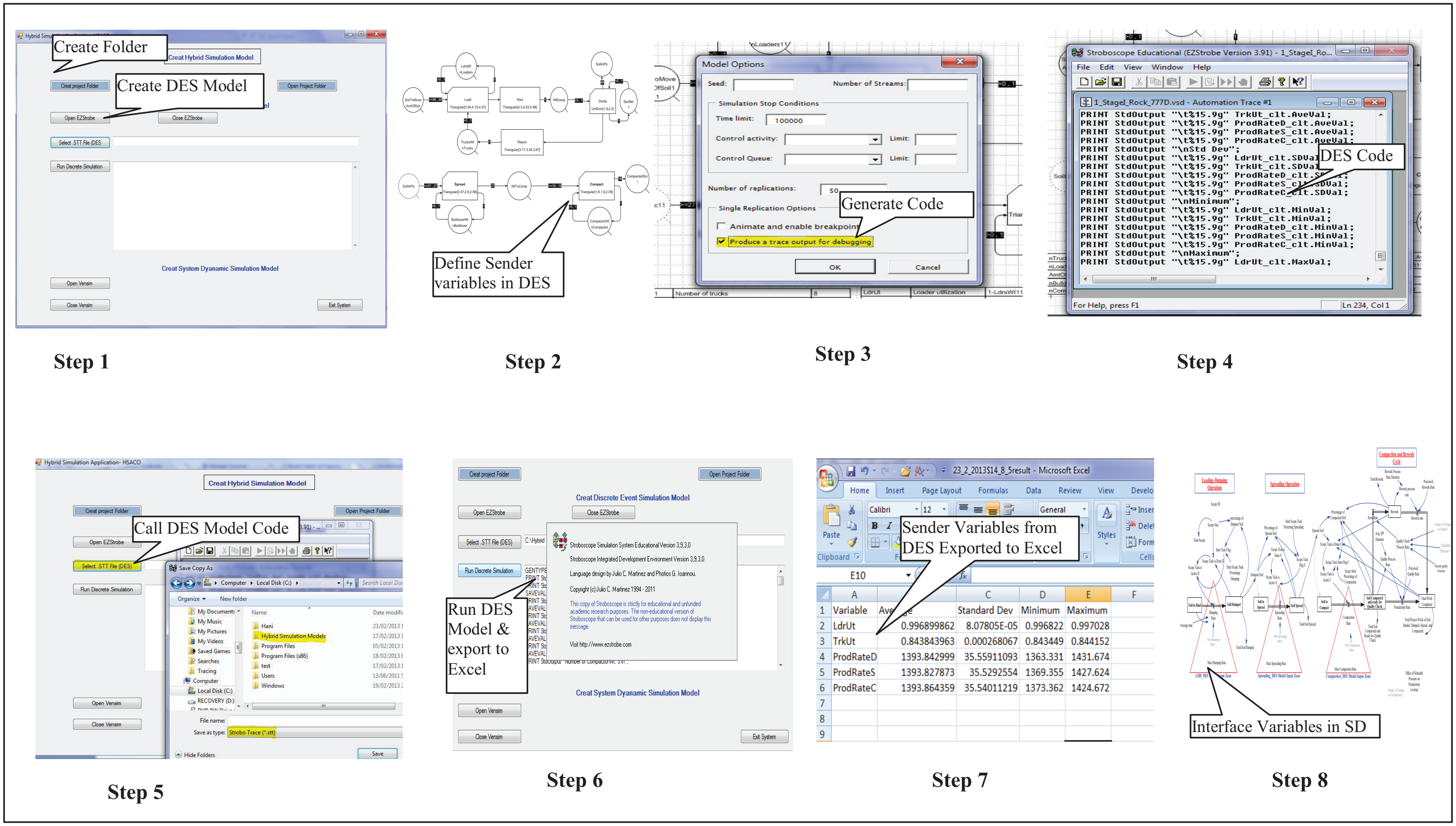


Figure 7.9 Steps of Applying HiSim to a Case Study

7.3 Results and Reports

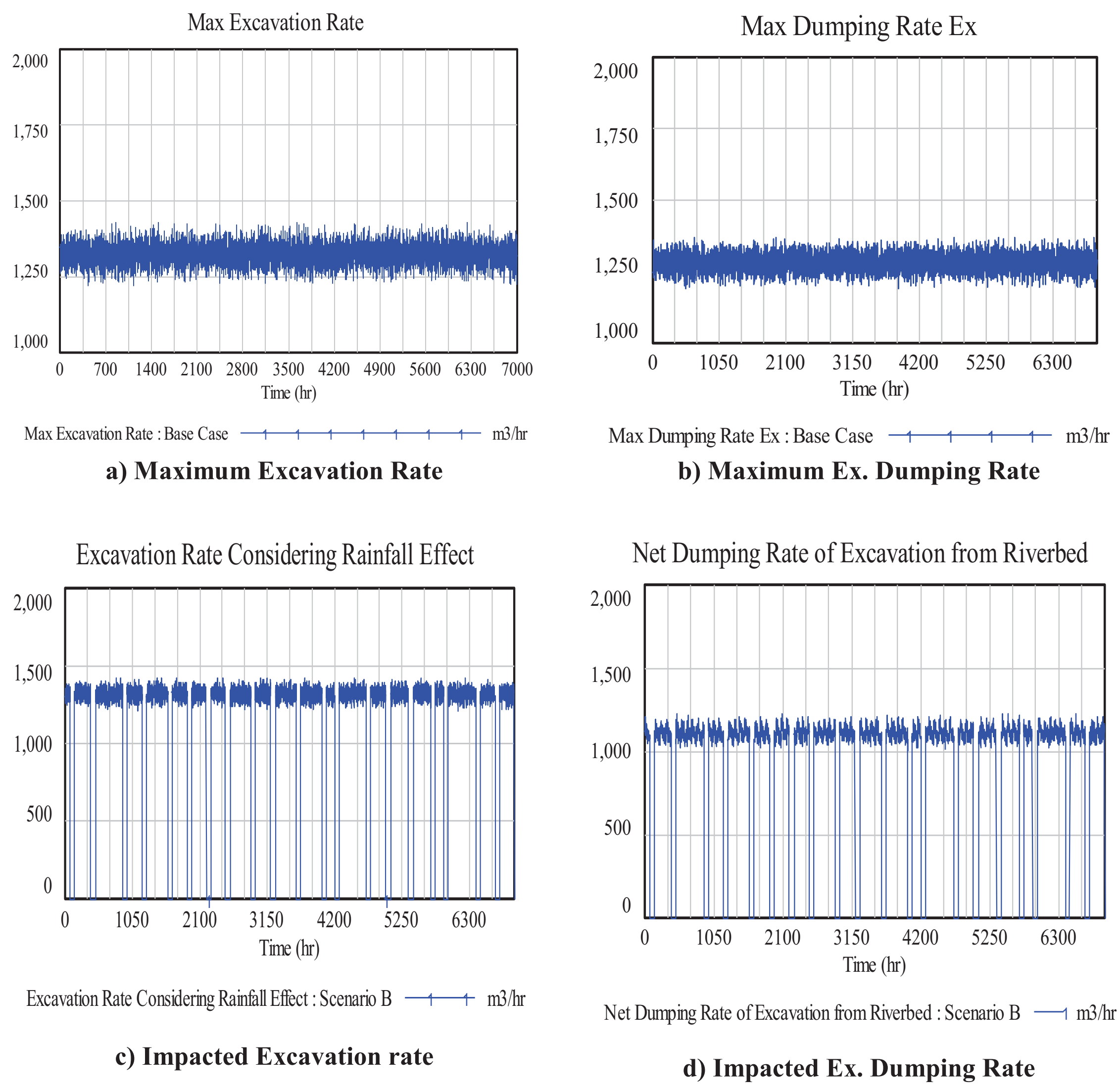
This section presents the results of the hybrid simulation model for the earthmoving operations used to demonstrate the use of HiSim. The same assumptions discussed in Chapter 6 are also applied here. Three types of results are presented. The first is the data of the sender variables, the second is the data of the impacted sender variables, and third is the productivity outcome of the simulation model.

7.3.1 Sender Variables of Excavation Operation

This section presents and discuses the hybrid simulation steps and results of HiSim. The process of checking whether the interface variables receive data from sender variables as was designed is investigated. This is part of the validation process of HiSim. Figure 7.10 demonstrates the behavior of two types of variables that were defined as senders and receivers in the excavation operation. Figure 7.10 (a-b) presents the first type of variables data that were computed by the excavation DES model. These variables were defined as the sender variables in the hybrid simulation model. The charts demonstrate a random behavior, where the value of the variable at every time step follows a stochastic behavior and picks up different values (random variant) every time the state updates occur. The charts show that these variables were interfaced successfully. For instance, the *excavation rate* and *excavation-dumping rate* defined in the SD model picks up different values every time the SD system updates the states of the variables. These values are imported from the random variant generated by the DES model.

7.3.2 Receiver Variables of Excavation Operation

The second type of variable behavior is shown in Figure 7.10 (c-d). These variables are the receiver variables. The values of these variables are computed by impacting the sender variables by the feedback process of the SD model loops. The impact is based on the policy and the structure of the SD model. For instance, the interruption in the behavior shown in the graph is due to a stoppage of the work execution due to a weather impact. There is a difference of approximately 25% between rates (DES results) and impacts (Hybrid results). The resulting impact rates shown in the figure represent a realistic behavior of the variables.



a) Maximum Excavation Rate

b) Maximum Ex. Dumping Rate

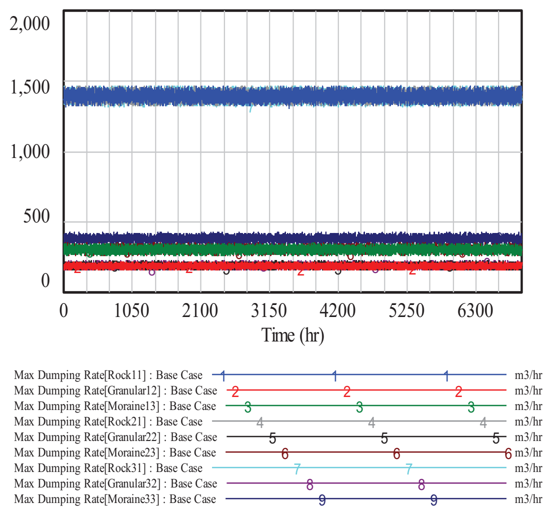
c) Impacted Excavation rate

d) Impacted Ex. Dumping Rate

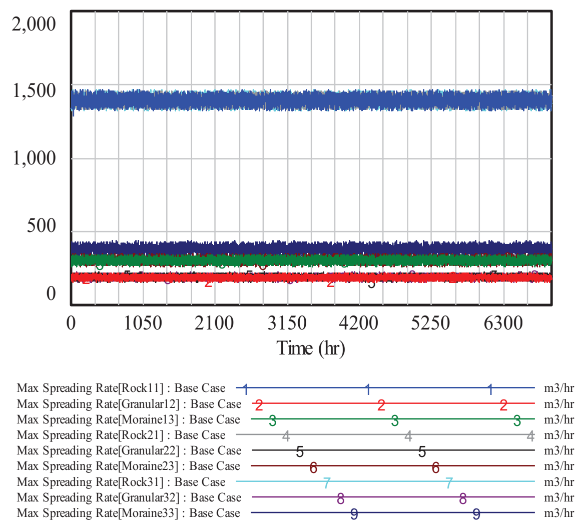
Figure 7.10 Sender and Receiver Variables in Excavation Operation

7.3.3 Sender Variables of Backfill Operation

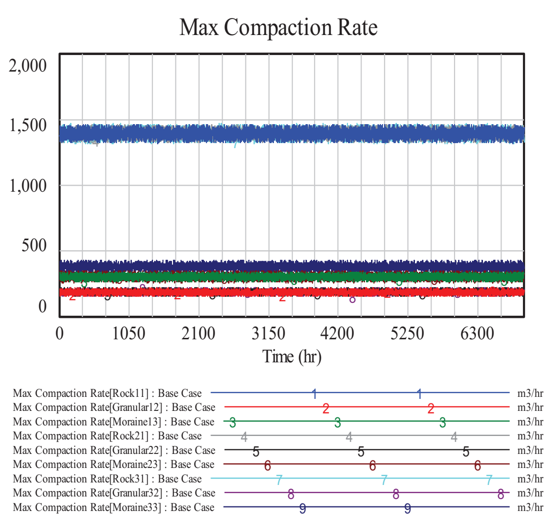
The second aspect of the hybrid simulation model presented here is related to the backfill operations results. Figure 7.11 (a, b, & c) shows the behaviors of the twenty-seven sender variables as computed by the backfill DES models. The graphs clearly show a stochastic behavior of the variables being imported. The Gantt chart that represents the sequence of the activities is shown in Figure 7.11 (d).



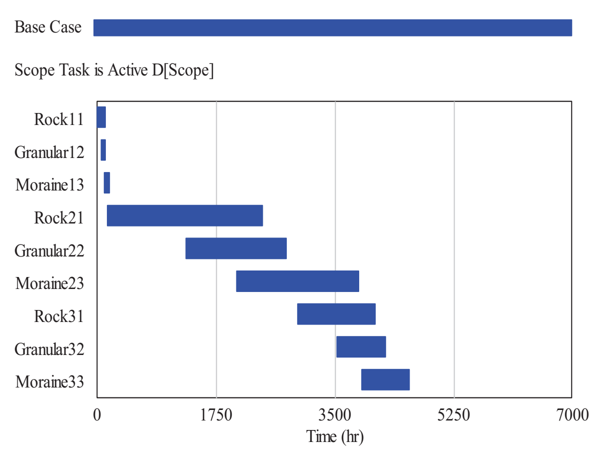
a) Max Dumping Rate



b) Max Spreading Rate



c) Max Compaction Rate

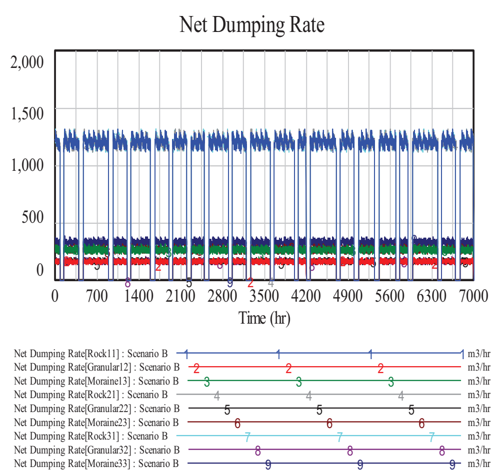


c) Gantt Chart of Operations

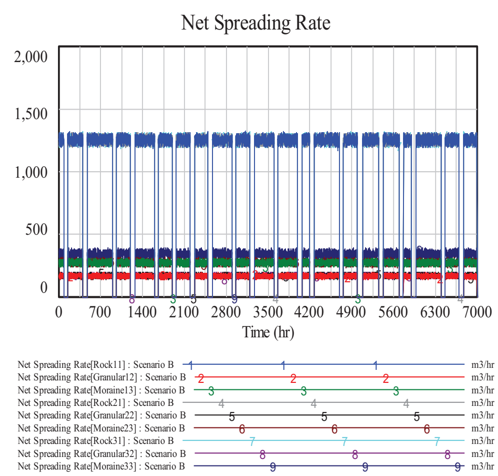
Figure 7.11 Sender Variables in Backfill Operations

7.3.4 Receiver Variables in Backfill Operations

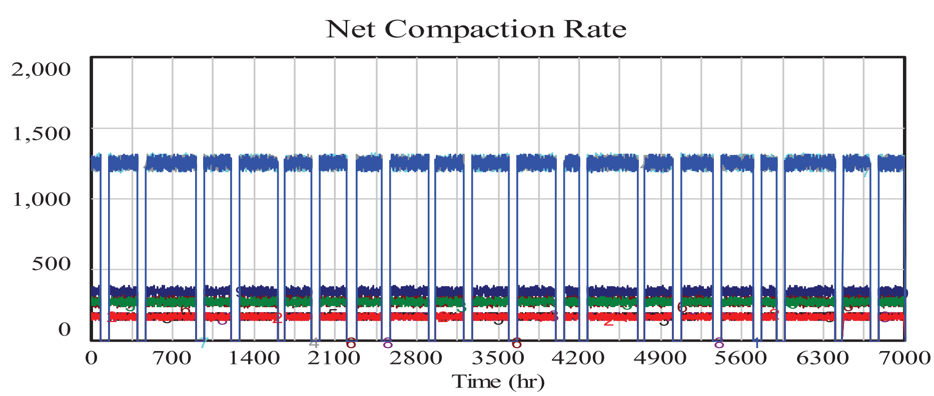
The outcomes of variables defined in the SD model such as *Net Dumping Rate*, *Net Spreading Rate* and *Net Compaction Rate* that receive inputs from interface variables are shown in Figure 7.12. The behavior of these three variables demonstrates the effects of the feedback loops and policies modeled in SD. The interruption of rates shown in the graphs is due to the impact of weather that resulted in stoppage of work execution. The difference between behaviors shown in Figure 7.11 and Figure 7.12 depicts the difference between the ideal and real simulation mode.



a) Impacted Net Dumping



b) Impacted Net Spreading



c) Impacted Net Compaction

Figure 7.12 Receiver Impacted Variables in Backfill Operations

7.3.5 Project Parameters Outputs

Certain productivity outputs such as actual released productivity, forecasted productivity, time required to complete, and total accumulated work are shown in Figure 7.13. For instance, the sender variables' productivity range was 1400 m^3/hr. However, the productivity indicators shown below are demonstrating behavior lower than what was anticipated. This is mainly due to the impact of policies and feedback loops. Time required is showing a fluctuating behavior as the model was designed to predict project duration and productivity based on policy and operational parameters together.

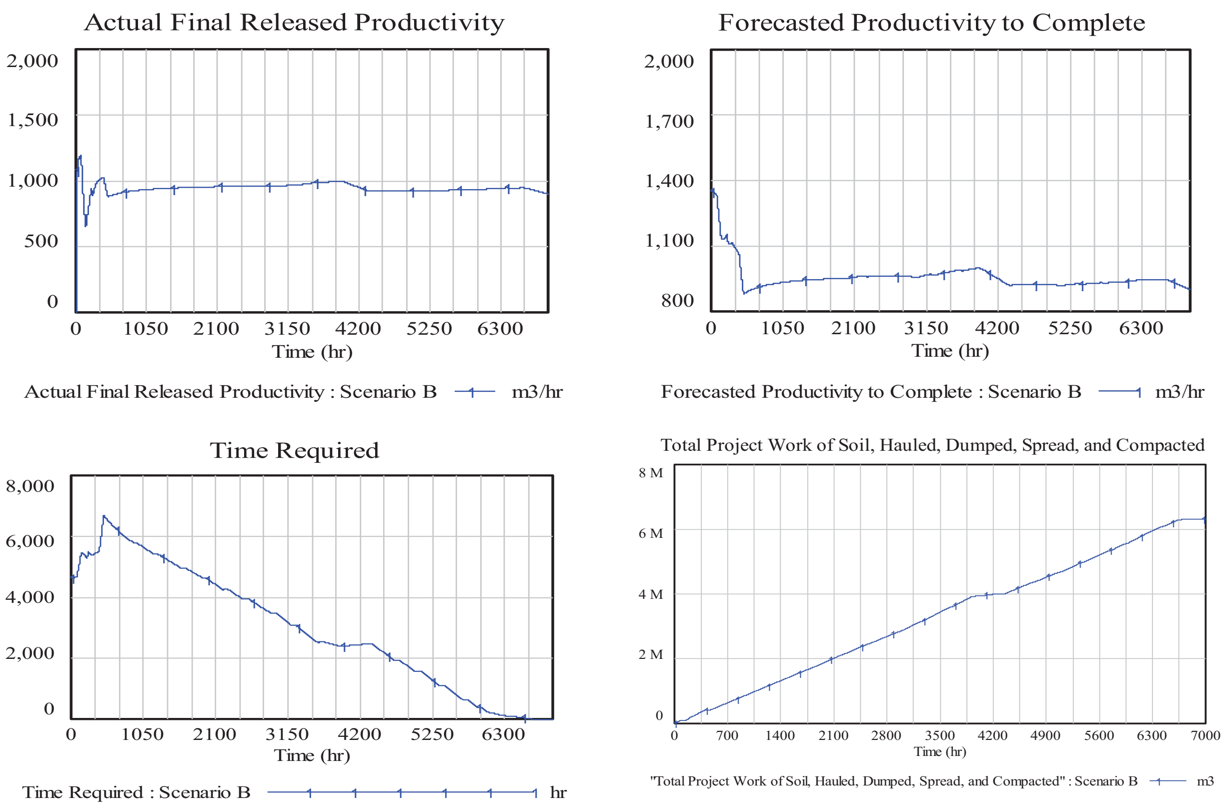


Figure 7.13 Outputs of Project Productivity Parameters

7.4 Validation of HiSim

Simulation is a computer-oriented research and it is almost exclusively a computer-based process. Simulation validation is different from validating other scientific tools. The issue of precision in simulation tools relies mainly on the correct implementation of the tool and the accuracy of data. These two conditions of precision work in parallel and compromising either of them results in a poor simulation model. For instance, conceptual mental models are believed to be accurate and representative. However, when those models are implemented and carefully examined using simulation tools, it can be found that such models are inaccurate. Thus, it is indispensable, before the deployment of a simulation model, to verify the conceptual model of the phenomenon under study. Validating the hybrid simulation model developed using HiSim has to be conducted at three levels: 1) the SD model level; 2) synchronization of variables; and 3) model behavior level.

7.4.1 SD Model Validation Level

The validation of the SD model is essential as it represents the conceptual model design of the problem under study. This validation process has been exhaustively performed on the earthmoving project model in Chapter 6. The model proved to be sound and representative.

7.4.2 Validation of Synchronization Process

The second level of the validation process involves conducting tests on the interfacing process among sender, interface, and receiver variables in the hybrid simulation model. This test is required to ensure the synchronization process is executed according to the developed protocol. As per the designed synchronization method, three types of variables were involved in the interactions process between the DES and SD models. These variables were classified earlier as sender variables that send state updates from DES model, interface variables that convey state updates to receiver variables defined in the SD model, and lastly the receiver variables that receive state updates from interface variables. As mentioned, twenty-nine variables of each category were involved in the synchronization protocol. This makes the total number of variables involved in the synchronization process equal to eighty-seven variables. The synchronization process was successfully performed and validated on all mentioned variables. The validation process is considered successful only if "*the interface variables developed in the SD model share similar behavior with sender variables developed in the DES models*." Figure 7.14 demonstrates comparisons between sender variables in DES model shown in Figure 7.14 (a) and interface variables shown in Figure 7.14 (b, c, d, and e). Through examining the variables' behavior illustrated in the figure, it is clear that both sender and interface variables share similar behaviors.

An example of validating eight variables in the hybrid simulation model (four sender variables and four interface variables) is carried out to validate the synchronization process. The illustration is limited to only eight variables due to the large number of variables involved (87 variables). The variables selected for the illustration are excavation rate, dumping rock11, spreading moraine23, and compaction granular32. The results of the simulation models pertaining to those variables are shown in Figure 7.14. The behaviors of DES model variables computed in the discrete simulation environment are shown in Figure 7.14 (a). The callouts shown in the figure illustrate the counterpart of variables behavior in the system's dynamic simulation environment, Figure 7.14 (b, c, d, & e). For instance, the callout containing the text "Compare with (b)" is pointing to the behavior of the excavation rate (sender-discrete variable) and comparing it to the behavior of the maximum excavation rate (interface-SD variable) as shown in Figure 7.14 (b). By examining the variable behaviors shown in Figure 7.14, it can be said that both sender and interface variables demonstrate identical behavior. Thus, it can be concluded that the synchronization process is successfully achieved.

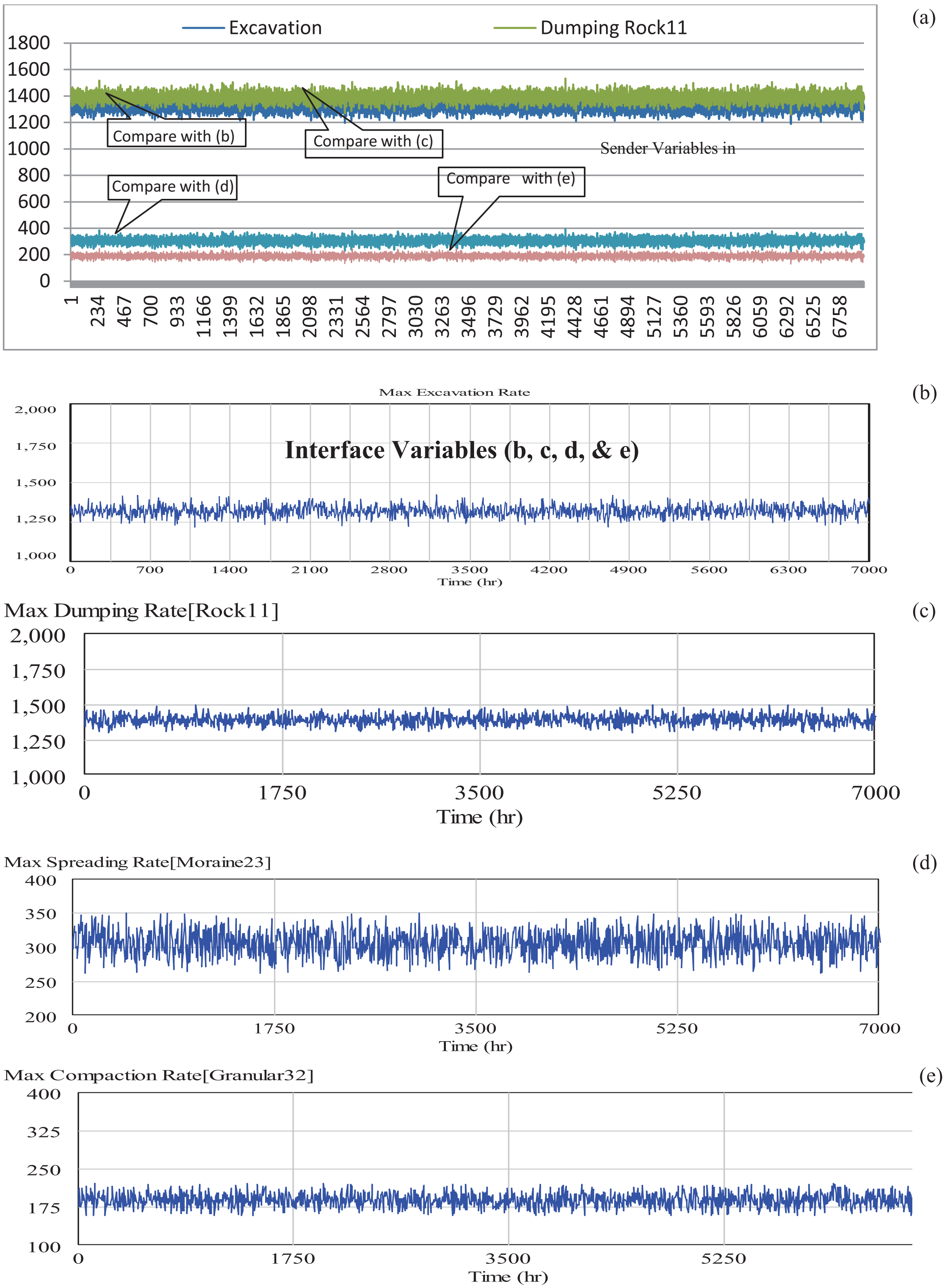


Figure 7.14 Comparison between Sender Variables (DES) and Interface Variables (SD)

7.4.3 Validating Model Behavior

The model's behavior and outcomes demonstrated in Figure 7.13 depict the behavior of the model discussed in Chapter 6. The outcomes are considered representative of the real behavior of the earthmoving project. For instance, when the impact variables and policies were excluded from the model structure, the model behaved similarly to the DES model. Furthermore, when the variables responsible for the dynamics of the project were loaded on the model, the productivity behavior followed an s-curve behavior. Therefore, the methodology, the model, and the automated tool have all demonstrated a potential for use in hybrid simulation applications for construction projects.

7.5 Limitation of HiSim

The hybrid simulation method proposed in Chapter 4 is a generic method and can be utilized in different ways to develop hybrid simulation applications that serve the needs of managers. As emphasized on in the development stage of the hybrid simulation method, the problem that requires deploying hybrid simulation dictates selection single hybrid model structure from the defined three structures in (section 4.2.3). The hybrid simulation model in this thesis is developed based on Strategic-Operation Structure. While the hybrid simulation method developed in this thesis is generic, the HiSim application is limited to hybrid simulation models that require Strategic-Operation Structure.

7.6 Chapter Summary

This chapter has presented the devolved Hybrid Simulation (HiSim) application for simulating construction operations. The application has been developed using the VB.NET Microsoft environment. The Executer code, which was the last component of the developed hybrid simulation methodology in Chapter 4 is the tool that was used to integrate the software packages used. The DES model was developed using STROBOSCOPE simulation language while the SD model was developed using Vensim. Microsoft Excel was used to export the DES model simulation output. The sender variables in DES were imported using a function in Vensim and mapped into the interface variables. The receiver variables in the SD model were connected to the interface variables to execute the mapping of data between the simulation models. A case study was utilized to demonstrate the implementation of the hybrid simulation application. Finally, the validation of the hybrid simulation application was conducted successfully at three levels.

CHAPTER 8

CONCLUSIONS AND FUTURE WORKS

8.1 Chapter Overview

This research has developed a hybrid simulation method and a hybrid simulation application (HiSim) to model and simulate construction operations. The method provided the roadmap to build hybrid simulation models and applications. The application means the software computational platform capable of computing hybrid simulation models. Six components were identified as essential elements for integrating DES and SD methods. These components were developed and successfully tested. Additionally, the developed method illustrated the challenging aspects that are indispensable to consider when developing hybrid simulation models. The integration process between DES and SD models was achieved through a new synchronization method, which has been developed in this thesis. Finally, the lessons learned from the research conducted in the field of construction project modeling are outlined, in addition to conclusions and future works.

8.2 Conclusion

Construction projects are composed of heterogeneous aspects. When a simulation tool is deployed, it must deal with those aspects; otherwise, model outcomes will be questionable. The DES method has been a useful tool to model and simulate construction operations. However, the underlying model of the DES method often seems to fail in representing real operations, as it tends to assume no interrelationship between project components. In reality, project components have a complex dynamic feedback process that requires modeling of inherent dynamics in simulation models. Nevertheless, this dynamic nature has not been explicitly addressed by the DES method. Project failure can be attributed to a poor representation of the inner and outer aspects of operations that are responsible for project dynamics. Uncontrollable external forces are often cited but the real cause may be internal such as the feedback process among components of the project. The DES technique simply describes the project as a top-to-bottom hierarchy through the decomposition of project elements to the smallest acceptable level called tasks. Then, costs, durations, and resources are estimated, mainly from experience, and loaded onto the tasks. The project's job logic is described as a network of tasks connected based on the work sequence and logic. The apparent purpose of this process is to describe the actual project behavior generated in reality. One of the main concerns with such a static philosophy of addressing the dynamics issues of planning and controlling lies in the ability of the restructured tasks of the network from bottom-to-top to behave based on the assumptions at the project decomposition stage. On the other hand, SD is a modeling and simulation method with a wide range of applications in different fields, used mainly to model strategic aspects. One of the strengths of SD method is modeling the whole system within a predetermined boundary. This allows understanding the system's behavior. However, SD method fails to compute the operational aspects at the tactical level. Thus, this research identified an opportunity to benefit from the capabilities of both DES and SD methods.

The purpose of any project model, whether it is an SD model or a DES model, is to strive to deliver an unbiased model that captures the likely behavior of project related parameters and their dynamic impact on project execution. The research has identified a persistent need for a hybrid simulation tool that responds to the increased complexity of construction operations, and at the same time, account for the project management decision levels (strategic and operational). The change is needed to shift from the fragmented modeling and simulation approach of construction operations to a more holistic integrative environment that accounts for heterogeneous aspects. For instance, DES simulates the operational aspects (e.g. server unit rate, auxiliary unit rate, and capacity) for the purpose of understanding system behavior, managing resource interactions and estimating productivities. Nevertheless, these factors are not the only factors responsible for generating the operation's real behavior. There is another important side of the operation that arises from the policy/context level (e.g., weather condition, labor skill, and fatigue). Furthermore, the DES method allows the resource interactions and model computations to be conducted in an ideal environment where the impacts of dynamic forces and surrounding factors that drive model behavior are neglected.

This research indicated that SD is well suited to address the dynamic nature of the project's interrelated parameters at the strategic level, while the DES method is well suited for modeling parameters at the tactical level. Since DES and SD methods were developed from different backgrounds, the process of integrating them is a challenging task. The main challenges are: 1) lack of hybrid simulation framework to build the hybrid simulation model; 2) synchronizing DES and SD simulation clocks, as the first updates states based on event occurrence and the second updates states based on elapsed time intervals; and 3) developing the Executer that integrates all these components. A method to integrate the construction project's strategic and operational management decision levels has been presented in this thesis. Two broad components of the proposed hybrid simulation method have been identified in this thesis: 1) a hybrid simulation method to develop hybrid simulation models; and 2) a hybrid simulation computation platform. In order to achieve these two broad components, a hybrid simulation method that encompassed six components was developed and presented in Chapter 4, and then the hybrid method was tested using a full real case study. The most important contributions of this research that have been presented and published are: 1) hybrid simulation framework (Alzraiee et al. 2012a); 2) dynamic planning of earthmoving operations (Alzraiee et. al. 2012b); and 3) synchronization of simulation clocks for the DES and SD methods (Alzraiee et al. 2012c). Since DES and SD update states differently, the synchronization method plays the main role in updating the states of the variables in both simulation models. The other important contribution was the developed formalism, which is needed to describe the model variables.

A real case study of a dam construction was used to implement the developed hybrid simulation method. The earthmoving operations involved in the dam project were modeled and simulated. The scope of work was excavating 1.038 million m^3 of soil from the riverbed and backfilling 6.3 million m^3 of three types of materials in three phases each. Modeling the case study using a hybrid simulation method took two directions. The first was utilizing the DES method to model operational aspects. To do so, ten DES models where developed using EZStrobe software, in which, one model was dedicated to the excavation operation, and nine models were dedicated to the backfilling operations. The second was to use the SD method to model the dynamics and policies inherent in the project. The feedback process of earthmoving operations variables were carefully prepared in causal-effect loops based on the conceptual model and boundary. Subsequently to the examination of the causal-effect loops to guarantee their conformity with reality, the mathematical SD model was developed using the Vensim simulation software. The concurrency issue among operations was addressed in the SD model using switches. The ten operations were successively planned with a 50% scope overlap between any two successive operations. Prior to integrating simulation models to provide the hybrid simulation environment, the SD model was successfully tested using several popular validation tests in the system dynamics modeling field.

The next major stage in developing the hybrid simulation model was to establish data mapping between the fragmented simulation models (synchronization/interfacing of variables). The hybrid model was designed to have the SD model in a controlling position, while the DES models were to be triggered by SD to compute certain variables needed in the SD model (e.g., excavation rate, dumping rate, spreading rate and compaction rate). The synchronization protocol divided the variables involved in the synchronization process into sender variables initiated in DES models, interface variables initiated in the SD model, and receiver variables also initiated in the SD model. Twenty-nine variables were identified in the DES models as necessary sender variables to achieve the objectives of the hybrid simulation model. These variables were interfaced with another twenty-nine interface variables in the SD model, which acted as conveyors of data to the receiver variables. In total, there were 87 variables involved in the synchronization process. The hybrid simulation model was tested using three scenarios for the case study: the base case, the extreme case, and a moderate case. The hybrid model offered a simulated duration of the real situation with 92 % accuracy. The most important feature that was offered by the hybrid simulation model was insight into the interaction among the project elements. In reality, this allows management to understand project behavior problems and, consequently, change policies using informed predictions of future outcomes.

The hybrid simulation method was implemented in the context of tool development. The Hybrid Simulation (HiSim) is the tool developed to automate the process; it integrates EZStrobe (DES simulation software), Vensim (SD simulation software), and Microsoft Excel. These software systems were integrated using the Executer developed in VB.net and shown in Appendix C. However, hybrid simulation software that is more sophisticated can be developed for hybrid simulation problems based on the developed method. This will require a tremendous amount of high and skilled programming work.

8.3 Lessons Learned and Important Issues

This section is intended for future researchers to benefit from the author's lessons learned during his Ph.D research, primarily in the area of management decision level, feedback process, discrete simulation, system dynamics, and hybrid simulation. Based on the exhaustive literature review conducted in this thesis, and the application of the developed method to a real-world case project, the author would like to highlight issues of concern. These issues are believed to be capable of providing insight into construction modeling problems, and could be the subject of future research in the field of simulation and modeling of construction projects.

8.3.1 Rework Cycle

Errors are very likely to occur in construction operations, and significantly affect the entire prepared action plan. Recognizing the existence of errors in construction work is important in noticing the impediments of the available traditional planning and simulation tools. When planning construction operations, errors generated in the work execution are typically not accounted for. Instead, contingencies that account for time and cost losses due to errors are added. The rework cycle contributes to cost and time overrun, and it even generates secondary and sometime tertiary errors. Problems occur in construction due to the interactions of operations' exogenous and endogenous variables. Excluding the rework cycle from simulation models has proven to affect the model outcomes. Analyses of the currently used discrete simulation and planning (e.g., CPM, PERT) demonstrate that these tools are not powerful enough to capture the rework cycle, as the SD can. Understanding the rework cycle evolution along with the causal-effect loops will enhance the predicted certainty of cost and time of the project as well as improve the response action plan.

8.3.2 Interactions of Construction Operation Parameters

Construction projects are of heterogeneous nature and have diverse characteristics. Using simulation tools in the construction field requires the modeler to deal with: 1) decision level; 2) system complexity; 3) types of variables; and 4) relationship among variables (Alzraiee et al. 2012a). A typical construction project system involves strategic and operational decisions made at different management levels. This also involves dealing with discrete and continuous variables. The relationships among these variables are in the form of cause-effect relationships. The system behavior is mainly generated based on the interactions of these aspects. However, modeling and simulating all these aspects at the same time is a major challenge and successfully modeling most influential aspects is a necessity to generate the real behavior of the system in the virtual world. In DES, one management decision level of construction operations is modeled to represent the behavior of the system, which means considering only one side of the issues (e.g., operational aspects). The outcomes of such models have failed to portray the real system behavior and the reasons of such behavior. The underlying models of the DES simulation approach tend to assume no interrelationships between project components. In reality, project components demonstrate a complex dynamic feedback process that should be involved in the simulation process. Nevertheless, this dynamic nature and uncertainty have not been explicitly addressed by the DES simulation method.

8.3.3 Holistic and Fragmented Perspectives

Construction operations can be modeled from two perspectives, holistic or fragmented. With an increase in the complexity of construction operations and the need to increase the certainty of the project's cost and time, more focus is needed to incorporate influential factors in a single simulation model. Recently, the efforts in simulation of construction operations have focused on exploring tools that are capable of setting a main framework (e.g., strategic or policy) that allows the elements of the modeled project to interact within the developed boundary. This prevents assumptions that hinder the model outcomes. For instance, consider the need to have process productivity at a level of 1000 units/hour. The DES model developed for this process would attempt to sequence the tasks involved in the process to establish job logic. Then, it would compute the model based on the resources, costs, and durations involved in each task. The model's recourses and durations are configured to have a productivity of 1000 units/hour. Now, the question is whether this model is capable of producing 1000 units/hour based on the static inputs. The answer is definitely no, as productivity fluctuates during the course of the project due to the results of interactions among discrete and continuous variables (Pritsker et al. 1997). Project dynamics and changes in management policies also play a role in maintaining or hindering productivity. Therefore, the problem should addressed from the opposite side by setting the model productivity target as 1000 units/hour and letting the model use resources based on dynamics or policies to keep the productivity rate as desired. This strategy allows management to have a better estimation of project parameters.

8.3.4 System Dynamics

SD, as stated earlier, is the process of studying the model behavior over time. It is an excellent technique to understand how system components interact. There are many aspects in construction management that can be well understood and quantified by using system dynamics. For instance, construction operations constituted of complex inter-relationships among their variables can be reflected in a form of cause-effect loops. Trying to gain insight into a system's behavior evolution needs a thorough understanding of the feedback process, and its roots at the evolution stage. In this case, modeling the construction system structure using SD helps managers understand the projects in depth. SD is a powerful technique and is well suited for enhancing how construction operations are perceived by management at the planning and execution stages. Nevertheless, applying SD to real construction operations for understanding the system behavior presents several hurdles, as SD lacks details of the lowest level of operations. Such impediments were conquered in this thesis through utilizing the DES method. Through this research, it can be affirmed that SD represents a promising area of research for applications in construction management, particularly in understanding system evolution, interaction, and behavior. Further, understating cause-effect loops generation and root causes inevitably enhances identifying problems early and promotes proactive strategies.

8.3.5 Hybrid Simulation in Other Fields

Management domains in other science fields such as manufacturing, supply chain, software projects and healthcare have explored hybrid simulation's benefits and uses. These fields are rich in rigorous research and special purpose models that can be studied and adapted to solve modeling problems in the construction field. Furthermore, exploring hybrid simulation applications in other fields provides a background to understating the need to use hybrid simulation and the structure type of the hybrid simulation models.

8.4 Research Contribution

This research presents the following contributions to the field of computer simulation application in construction management.

1. Developed a strategy to integrate diverse construction operation elements into a single model that assists management in understating behavior evolution and the impact of policies/decisions on project outcomes.
2. Developed a hybrid simulation system that is capable of providing guidelines for hybrid simulation modeling.
3. Integrated strategic and operational decision levels on a single computational platform. The developed method integrates SD models used to model continuous variables and DES models to model discrete variables.
4. Developed a mathematical formalism that describes elements of the DES and SD models in the synchronization process.
5. Developed a synchronization method that facilitates the time advancement and state update mechanism in DES and SD models. The developed method preserves the distinct features of each state update mechanism.
6. Developed a generic system dynamic model for earthmoving operations.
7. Developed the Executer responsible for integrating hybrid simulation components.

8.5 Limitations

The developed hybrid simulation method is a relatively new simulation method applied to construction operations modeling. The limitations associated with the developed method are listed below.

1. Restriction of the hybrid developed system to integrate DES and SD only. Other simulation methods exist, such as agent based modeling (ABM), which provide a potential for modeling heterogeneous systems like construction projects.

2. Developing and validating the SD model is time consuming and requires modeling skills. Furthermore, the SD model relies on the conceptual model and availability of detailed data.
3. The developed method requires that the modeler be knowledgeable in discrete and continuous simulation methods.
4. The synchronization method developed using the Time Bucket concept can have a high computation overhead for large models.

8.6 Future Work

The thesis has presented a method for enhancing the modeling and simulation of construction operations through integrating DES and SD models. The presented simulation method can be enhanced through improving and extending the existing research. The following subsections discuss the potential areas where the hybrid simulation method could be improved or extended.

8.6.1 Improving Existing Research

(i) Hybrid Simulation Structure

Through studying the previously developed hybrid simulation model in different fields, three hybrid simulation structures were used. The issue of selecting a hybrid simulation is mainly dependent on the nature of the problem and the purpose of the model. In the implementation chapter, only one type of hybrid simulation structure was tested. The SD model was designed as the controlling structure driving the simulation model. However, the SD simulation did not dominate the simulation behavior. The DES models were used as auxiliary units in the global simulation model, without rendering DES as secondary models. While the developed hybrid simulation method is generic and can be applied to different hybrid model structures, this implementation is limited to the strategic-operation hybrid simulation structure. Hence, there is a need to test other hybrid simulation structures identified in Chapter 4.

(ii) Formalism

The developed simulation focused on two directly involved variables (sender and receiver) and the synchronization function. The direct impacts of the interface variables from either side are monitored. If the formalism is extended to involve secondary impacted variables, this inevitability improves the understanding of the system under investigation. Sometimes it is necessary to know how sensitive the secondary variables are to interface variables. For instance, the dumping rate (sender) from the DES model is directly mapped into the dumping rate (receiver) in the SD model. It is not clear how this rate could affect the loading or the hauling process. Such impact propagation among variables needs to be traced and quantified. Furthermore, the formalism could be split into two parts, in which one part addresses the interface variables and the other part addresses the synchronization process. This is expected to generate a more meticulous integration process.

(iii) Enhancement of Developed Hybrid Simulation System

The developed hybrid simulation method is the first of its type in simulating and modeling construction operations. Its elements are extracted from the need of construction projects for a hybrid simulation. The method involved six components that are necessary for developing a hybrid simulation model. This framework could be expanded to include other simulation philosophies. Further enhancements can be introduced through increasing the number of criteria that decides which simulation method to use for a particular element of the project. The research has adopted only six criteria. Adding more criteria would allow a better simulation method selection process.

The proposed hybrid simulation method could be implemented using the powerful capabilities of Java programming language. Such programming work should develop new discrete and continuous simulation engines, and integrate them based on the developed hybrid simulation components. The essence of the hybrid method was presented, and the rest requires professional programming work.

8.6.2 Extension to Existing Research

(i) Integrating Agent-Based (AB) with SD

AB modeling is a powerful simulation technique that has been successfully applied to different fields. It models systems as a collection of autonomous decision-making entities called agents. Each agent is individually responsible for making decisions based on a set of rules. In these simulation models, agents are responsible for depicting the behavior. AB is capable of describing the complex behavior of a system at the tactical level. In this context, SD model integration with AB models would probably be capable of delivering hybrid simulation models that are more sophisticated in describing the real interactions of variables. Further investigation is needed in this domain.

(ii) Causal-Effects Loops Evolution

Another interesting area of research would be the study of the causal-effect loops generation mechanism and the development of a control system to mitigate negative effects and boost positive ones. Causal-effect loops are responsible for system behavior evaluation. In this thesis, the roots of the causal-effects loops were studied and analyzed. The results uncovered the need for further insight into the feedback process. The main attention should be directed at exploring the existing loops and monitoring the new evolving loops due to policy and boundary changes. An algorithm capable of mitigating the negative effects on project success parameters and enforcing the positive effects would be of great value in understating real system behavior and pointing out problematic loops.

(iii) Tri-platform of Building Information Model (BIM), SD, and Lean Construction

This future proposed research work would focus on a tri-platform of SD, BIM, and Lean Construction. BIM is an emerging approach that assists in implementing lean construction processes by eliminating waste, reducing costs, improving productivity, and having a better insight into project interactions. BIM modeling provides a powerful platform for visualizing and simulating workflow in a controlled environment. Aggregating available project data, simulating the proposed sequencing of construction operations, and presenting the project system interdependencies are main challenges to a real 4D model that emulates the real system over time (similar to the SD principle). Having the BIM model represent reality, the Lean Construction elements can be integrated with the BIM model to enhance all aspects of construction operation planning and execution phases. This intersects with hybrid simulation that focused on developing an integrated simulation environment capable of providing reality approximating models. The research theme is: 1) SD model representing the strategic and the context levels of a project; 2) BIM model representing the operational level (activities interactions); and 3) Lean Construction Elements to improve the quality of the model. The integration of the three elements on a single platform is expected to enhance the planning and the execution phases of construction projects under frames that represent the strategic/context and quality.

REFERENCES

Abdel-Hamid, T., & Madnick, S. (1991). *Software project dynamics: An integrated approach*. Englewood Cliffs, NJ.: Prentice-Hall.

AbouRizk, S. (2010). Role of simulation in construction engineering and management. *Journal of Construction Engineering and Management, 126*(10), 1140-1153.

AbouRizk, S., & Hague, S. (2009). An overview of the COSYE environment for construction simulation. *Proceedings of the 41st Conference on Winter Simulation, Dec 13-16,* Austin, TX. 2624-2634.

AbouRizk, S., & Hajjar, D. (1998). A framework for applying simulation in construction. *Canadian Journal of Civil Engineering, 25*(3), 604-617.

AbouRizk, S., Halpin, D., & Lutz, J. (1992). State of the art in construction simulation. *In Proceedings of the 24th Winter Simulation Conference, Dec 13-16,* 1271-1277.

Alvanchi, A., Lee, S. H., & AbouRizk, S. (2011). Modeling framework and architecture of hybrid system dynamics and discrete event simulation for construction. *Computer‐Aided Civil and Infrastructure Engineering, 26*(2), 77-91.

Alvanchi, A., Lee, S., & AbouRizk, S. (2009). Modeling architecture for hybrid system dynamics and discrete event simulation. *Construction Research Congress - Building a Sustainable Future, April 5-7,* Seattle, WA, United states. 1290-1299.

Alzraiee, H., Moselhi, O. and Zayed, T. (2012b) Dynamic planning of earth moving operations. *International Symposium of Automation and Robotics in Construction*, Technische Universiteit Eindhoven (TU/e) June 26–29, 2012

Alzraiee, H., Moselhi, O. and Zayed, T. (2012c) Methodology for synchronizing DES and SD models. *Proceedings of the 2012 Winter Simulation Conferen*ce C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds. December 9-12, 2012 Hotel Intercontinental, Berlin, Germany.

Alzraiee, H., Moselhi, O., and Zayed, T. (2012a) A Hybrid framework for modeling construction operations using discrete event simulation and system dynamics. *Proceedings of Construction Research Congress*- ASCE- West Lafayette, IN, USA"1063-1073.

Amirkhanian, S., & Baker, N. (1992). Expert system for equipment selection for earthmoving operations. Journal of Construction Engineering and Management , 118 (2), 318-331

Baines, T., & Harrison, D. (1999). An opportunity for system dynamics in manufacturing system modeling. *Production Planning & Control, 10*(6), 542-552.

Banks, J., Nelson, B., & Nicol, D. (Eds.). (2000). *Discrete-event system simulation* (3rd ed.). New Jersey, NJ.: Prentice Hall.

Binder, T., Vox, A., Belyazid, S., Haraldsson, H., & Svensson, M. (2004). Developing system dynamics models from causal loop diagrams. *Proceedings of the 22nd International Conference of the System Dynamics Society, July 25-29,* Oxford.

Borshchev, A., Kolesov, Y., & Senichenkov, Y. (2000). Java engine for UML based hybrid state machines. *In Proceedings of the 32 Nd Conference on Winter Simulation, Dec 10-13,* Orlando, FL. 1888-1894.

Brailsford, S., & Hilton, N. (2001). A comparison of discrete event simulation and system dynamics for modelling health care systems. *Proceedings of the 26th Meeting of the ORAHS Working Group,* Glasgow, Scotland. 18-39.

Bryant, R. (1977). *Simulation of Packet Communication Architecture Computer Systems.,*

Borshchev, A. and Filippov. A, (2004). From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. *The 22nd International Conference of the System Dynamics Society,* July 25 – 29, Oxford, England

Buss, A., & Jackson, L. (1998). Distributed simulation modeling: A comparison of HLA, CORBA, and RMI. *Proceedings of the 30th Conference on Winter Simulation, December 13– 16,* Washington, DC. 819-826.

Ceylan, B., & Ford, D. (2002). Using options to manage dynamic uncertainty in acquisition projects. *Acquisition Review Quarterly, 9*(4), 243-258.

Chahal, K., & Eldabi, T. (2008). A generic framework for hybrid simulation in healthcare. *Proceedings of the 26th International Conference of the System Dynamics Societ,y* Miami, FL. 1469-1477.

Chan, A. P. C., Scott, D., & Chan, A. P. L. (2004). Factors affecting the success of a construction project. *Journal of Construction Eng. and Management, 130*(1), 153-155.

Chandy, K., & Misra, J. (1979). Distributed simulation: A case study in design and verification of distributed programs. *IEEE Transactions on Software Engineering,* (5), 440-452.

Chen, D., John, S. Cai, T, and Xiong, M. (2008). A decoupled federate architecture for high level architecture-based distributed simulation, Journal of Parallel and Distributed Computing, 68 (11), 1487-1503

Choi, K., Bae, D., & Kim, T. (2006). An approach to a hybrid software process simulation using the DEVS formalism. *Software Process: Improvement and Practice, 11*(4), 373-383.

Chua, D., & Li, G. (2002). RISim: Resource-interacted simulation modeling in construction. *Journal of Construction Engineering and Management, 128*(3), 195-203.

Cooper, K. (1980). Naval ship production: A claim settled and a framework built. *Interfaces, 10*(6), 20-36.

Cooper, K. (1993 a,b,c). The rework cycle: Why projects are mismanaged. *Project Management Journal* , 5-17.

Cooper, K. (1994). The $2,000 hour: How managers influence project performance through the rework cycle. *Project Management Journal, 25*(1), 11-11.

Coyle, R. (1985). Representing discrete events in system dynamics models: A theoretical application to modeling coal production. *The Journal of the Operational Research Society, 36*(4), 307-318.

Dahmann, J. S., & Morse, K. L. (1998). High level architecture for simulation: An update. *Distributed Interactive Simulation and Real-Time Applications, Proceedings. 2nd International Workshop on, 19-20 Jul,* , 32-40.

Dahmann, J., Fujimoto, R., & Weatherly, R. (1998). The DoD high level architecture: An update. *Proceedings of the 30th Conference on Winter Simulation, December 13 - 16,* Washington, DC. 797-804.

Dissanayake, M., & AbouRizk, S. (2007). Qualitative simulation of construction performance using fuzzy cognitive maps. *Proceedings of the 39th Conference on Winter Simulation, 9-12 Dec,* Washington, D.C. 2134-2140.

Donald, B., & Boyd, P. (1984). *Professional construction management* (2nd edition ed.). New York, USA: McGraw-Hill Science.

El-Rayes, K. and Moselhi, O. (2001). ”Impact of Rainfall on the Productivity of Highway Construction.” J. Constr. Eng. Manage., 127(2), 125–131.

El-Rayes, K., & Moselhi, O. (1998). Resource-driven scheduling of repetitive activities. *Construction Management and Economics, 16*(14), 433-446.

Elwakil, E. (2011). *Knowledge discovery based simulation system in construction.* Ph.D. Dissertation, Building, Civil and Environmental Dept., Concordia University, Montreal, QC, Canada).

Ford, D. (1995). *The dynamics of project management: An investigation of the impacts of project process and coordination on performance.* Ph.D. Dissertation, Massachusetts Institute of Technology, Sloan School of Management, Cambridge, MA.

Ford, D., & Sterman, J. (1998). Dynamic modeling of product development processes. *System Dynamics Review, 14*(1), 31-68.

Ford, D., & Sterman, J. (2003). The liar's club: Concealing rework in concurrent development. *Concurrent Engineering Research and Applications, 11*(3), 211-219.

Ford, D., Anderson, S., Damron, A., Casas, R., Gokmen, N., & and Kuennen, S. (2004). Managing constructability reviews to reduce highway project durations. *Journal of Construction Engineering and Management, 130*(1), 33-42.

Forrester, J. (1961). *Industrial dynamics*. Cambridge, MA: Productivity Press.

Forrester, J. (1965). *Industrial dynamics* (4th Ed. ed.). MA, USA: MIT Press.

Forrester, J. (1975). Industrial dynamics after the first decade. In (Chapter 8 in “Collected papers of Jay W. Forrester ed.,) Wright-Allen Press, Inc.

FPC user’s manual—Version 3.0. (1997). Caterpillar Inc., Peoria, Ill.

Fujii, S., Tsunoda, H., Ogita, A., & Kidani, Y. (1994) Distributed simulation model for computer integrated manufacturing. *Proceedings of the 26th Conference on Winter Simulation, Dec 5–8,* Orlando, FL. 946 - 953.

Fujimoto, R. (2000). *Parallel and distributed simulation systems*. New York, NY: John Wiley & Sons.

Fujimoto, R. (2001). Parallel simulation: Parallel and distributed simulation systems. *Proceedings of the 33nd Conference on Winter Simulation, Dec 9-12,* San Diego, CA. 147-157.

Fujimoto, R. (2003). Parallel simulation: Distributed simulation systems. *Proceedings of the 35th Conference on Winter Simulation: Driving Innovation, Dec 7-10,* New Orleans, LA. 124-134.

Hajjar, D., & AbouRizk, S. (1997). AP2-earth: A simulation based system for the estimating and planning of earth moving operations. *Proceedings of the 29th Conference on Winter Simulation, Dec 7-10,* Washington, DC, USA. 1103-1110.

Hajjar, D., & AbouRizk, S. (1999). Simphony: An environment for building special purpose construction simulation tools. *Proceedings of the 31st Conference on Winter Simulation,* Phoenix, AZ , USA. *, 2* 998-1006.

Hajjar, D., & AbouRizk, S. (2002). Unified modeling methodology for construction simulation. *Journal of Construction Engineering and Management, 128*(2), 174-185.

Halpin, D. (1973). *An investigation of the use of simulation networks for modeling construction operations.* (Ph.D. Dissertation, University of Illinois at Urbana-Champaign).

Halpin, D. (1977). CYCLONE—method for modeling job site processes. *Journal of the Construction Division, 103*(3), 489-499.

Halpin, D., & Riggs, L. (1992). *Planning and analysis of construction operations*. New York, NY.: John Wiley and Sons.

Halpin, D., & Woodhead, R. (1997). *Construction management* (2nd ed.). New York, USA: Wiley.

Halpin, D., Jen, H., & Kim, J. (2003). A construction process simulation web service. *In Proceedings of the 35th Conference on Winter Simulation,* New Orleans, LA. *, 2* 1503-1509.

Han, S. (2008). *A hybrid simulation model for understanding and managing non-value adding activities in large-scale design and construction projects.* Ph.D. Dissertation, University of Illinois at Urbana-Champaign, IL USA.

Hanna, A. S., C. S. Taylor, and K. T. Sullivan. (2005). Impact of Extended Overtime on Construction Labor Productivity. Journal Construction Engineering and Management, 131(6), pp 734-739.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming, 8*(3), 231-274.

Helal, M. (2008). *A hybrid system dynamics-discrete event simulation approach to simulating the manufacturing enterprise.* Ph.D.Thesis, University of Central Florida, Orlando, FL, USA.

Helal, M., Rabelo, L., Sepúlveda, J., & Jones, A. (2007). A methodology for integrating and synchronizing the system dynamics and discrete event simulation paradigms.

International Conference of the System Dynamics Society and 50th Anniversary Celebration, Boston.

Homer, J., Sterman, J., Greenwood, B., & Perkola, M. (1993). Delivery time reduction in pulp and paper mill construction projects: A dynamic analysis of alternatives. Cancun, Mexico. 212-221.

Horner, R. M. W., and Talhouni, B. T. (1995). Effects of accelerated working, delays, and disruptions on labour productivity, Chartered Institute of Building, Ascot, Berkshire, U.K

Huang, G., Lau, J., & Mak, K. (2003). The impacts of sharing production information on supply chain dynamics: A review of the literature. *International Journal of Production Research, 41*(7), 1483-1517.

Huang, R., & Halpin, D. (1994). Visual construction operation simulation: The DISCO approach. *Computer‐Aided Civil and Infrastructure Engineering, 9*(3), 175-184.

Hydro Quebec (2003) “Construction of the Sainte-Marguerite-3 Hydroelectric Development 1994-2002” Bibliothèque nationale du QuébecISBN 2-550-40297-9

Hydro-Quebec Web Site (1999). Sainte-Marguerite-3 Hydroelectric Project: in harmony with the environment.

Hydro-Quebec Web Site (2012). Discover new Sainte-Marguerite-3 Hydroelectric Project. ‹ http://www.hydrosustainability.org/getattachment/871c6bf5-1607-452f-94d3-1e7632330a5c/Sainte-Marguerite,-Canada.aspx›. (July 7th)

Ioannou, P. (1989). UM-CYCLONE user’s manual. *Division of Construction Engineering and Management, Purdue University, West Lafayette, Ind,*

Jefferson, D. (1985). Virtual time. *ACM Transactions on Programming Languages and Systems (TOPLAS), 7*(3), 404-425.

Kamat, V. R., and Martinez, J. C. (2001), “Visualizing simulated construction operations in 3D”, *Journal of Computing in Civil Engineering*, Vol. 14, No. 4, ASCE, Reston, VA, 329-337.

Kannan, G. 1999. A Methodology for the Development of a Production Experience Database for Earthmoving Operations using Automated Data Collection. *PhD Thesis*, Virginia Polytechnic Institute and

Kelton, W., & Law, A. (2000). *Simulation modeling and analysis* (3rd Edition ed.) McGraw Hill.

Kowalewski, S., Stursberg, O., Fritz, M., Graf, H., Hoffmann, I., Preußig, J. Treseler, H. (1999). A case study in tool-aided analysis of discretely controlled continuous systems: The two tanks problem. *Hybrid Systems V,*163-185.

Kuhl, F., Weatherly, R., & Dahmann, J. (1999). *Creating computer simulation systems: An introduction to the high level architecture*. NJ, USA: Prentice Hall PTR.

Lane, D. (2000). You just don't understand me: Models of failure and success in the discourse between system dynamics and discrete event simulation. *Working Paper, London School of Economics and Political Sciences,* London, UK.

Lee, J., Zhou, M., & Hsu, P. (2004). A multi-paradigm modeling approach for hybrid dynamic systems. *IEEE International Symposium on Computer Aided Control Systems Design,* Taipie, Tiwan,. 77-82.

Lee, S. (2006). *Dynamic planning and control methodology: Understanding and managing iterative error and change cycles in large-scale concurrent design and construction projects.* Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, MA, USA.

Lee, S., Han, S., & Pea-Mora, F. (2009). Integrating construction operation and context in large-scale construction using hybrid computer simulation. *Journal of Computing in Civil Engineering, 23*(2), 75-83.

Lee, S., Han, S., & Peña‐Mora, F. (2007). Hybrid system dynamics and discrete event simulation for construction management. *Proceedings of the ASCE International Workshop on Computing in Civil Engineering,* Pittsburgh, Pennsylvania. 232-239.

Lee, S., Peña-Mora, F., & Park, M. (2006). Dynamic planning and control methodology for strategic and operational construction project management. *Automation in Construction, 15*(1), 84-97.

Lee, Y., Cho, M., Kim, S., & Kim, Y. (2002). Supply chain simulation with discrete-continuous combined modeling. *Computers & Industrial Engineering, 43*(1-2),375-392.

Levitt, R., Thomsen, J., Christiansen, T., Kunz, J., Jin, Y., & Nass, C. (1999). Simulating project work processes and organizations: Toward a micro-contingency theory of organizational design. *Management Science, 45*(11), 1479-1495.

Lyneis, J., & Ford, D. (2007). System dynamics applied to project management: A survey, assessment, and directions for future research. *System Dynamics Review, 23*(2‐3), 157-189.

Lyneis, J., Cooper, K., & Els, S. (2001). Strategic management of complex projects: A case study using system dynamics. *System Dynamics Review, 17*(3), 237-260.

Mak, H. (1993). *System dynamics and discrete event simulation modelling.* Ph.D. Dissertation, London School of Economics and Political Science, University of London, London, UK.

Maler, O., Manna, Z., & Pnueli, A. (1992). From timed to hybrid systems. in real-time: Theory in practice, eds. bakker, J.; huizing, C.; roever, W.; rozenberg, G.; springer-verlag,
Germany.

Martin, R., & Raffo, D. (2001). Application of a hybrid process simulation model to a software development project. *Journal of Systems and Software, 59*(3), 237-246.

Martinez, J. (1996). *Stroboscope: State and resource based simulation of construction processes.* Ph.D. Dissertation, University of Michigan, Michigan, MI, USA.

Martinez, J., & Ioannou, P. (1994). General purpose simulation with stroboscope. *Proceedings of the 26th Conference on Winter Simulation, Dec 11-14,* San Diego, CA, USA. 1159-1166.

Martinez, J., & Ioannou, P. (1999). General-purpose systems for effective construction simulation. *Journal of Construction Engineering and Management, 125*(4), 265-276.

Martinez, J., Ioannou, P., & Carr, R. (1994). State and resource based construction process simulation. Washington, D.C.. 177-184.

Marzouk, M. (2002). *Optimizing earthmoving operations using computer simulation*. Ph.D. Dissertation, Concordia University, Montreal, QC, Canada.

Marzouk, M., & Moselhi, O. (2003). Object-oriented simulation model for earthmoving operations. *Journal of Construction Engineering and Management, 129*(3), 173-182.

McKenna, N. (2005). Executing major projects through contractors. Boston, USA. 112-115.

McLean, C., & Riddick, F. (2000). Simulation in the international IMS MISSION project: The IMS MISSION architecture for distributed manufacturing simulation. *Proceedings of the 32nd Conference on Winter Simulation,* Orlando, FL, USA. 1539-1548.

Meadows, D. (1980). "The unavoidable A priori. Elements of the system dynamics methods." Cambridge, MA: MIT Press, 23-57

Miller, D., & Thorpe, J. (1995). SIMNET: The advent of simulator networking. *Proceedings of the IEEE, 83*(8), 1114-1123.

Miller, T. (2002). *Hierarchical operations and supply chain planning* (New York ed.). New York: Springer.

Montgomery, D., & Myers, R. (2002). Response surface methodology: Process and product optimization using designed experiments. *Response Surface Methodology: Process and Product Optimization using Designed Experiments,*

Morecroft, J., & Robinson, S. (2005). Explaining puzzling dynamics: Comparing the use of system dynamics and discrete-event simulation. *Proceedings of the 23rd International Conference of the System Dynamics Society,* Boston, USA.

Morris, P., & Hough, G. (1987). *The anatomy of major projects: A study of the reality of project management*. New York: John Wiley & Sons.

Ogunlana, S., Li, H., & Sukhera, F. (2003). System dynamics approach to exploring performance enhancement in a construction organization. *Journal of Construction Engineering and Management, 129*(5), 528-536.

Oloufa, A. (1993). Modeling operational activities in object-oriented simulation. *Journal of Computing in Civil Engineering, 7*(1), 94-106.

Oloufa, A., Ikeda, M., & Nguyen, T. (1998). Resource-based simulation libraries for construction. *Automation in Construction, 7*(4), 315-326.

Park, M., & Peña‐Mora, F. (2003). Dynamic change management for construction: Introducing the change cycle into model‐based project management. *System Dynamics Review, 19*(3), 213-242.

Paulson, B. (1978). Interactive graphics for simulating construction operations. *Journal of the Construction Division, 104*(1), 69-76.

Paulson, B., Chan, W., & Koo, C. (1987). Construction operations simulation by microcomputer. *Journal of Construction Engineering and Management, 113*(2), 302-314.

Peer, G. A. (1998). "Powerhouse Takes Shape Far from Dam SM-3 Project" Heavy Construction News, March 2001, pp.14-16.

Peña-Mora, F., Han, S., Lee, S. H., & Park, M. (2008). Strategic-operational construction management: Hybrid system dynamics and discrete event approach. *Journal of Construction Engineering and Management, 134*(9), 701-710.

Pepyne, D., & Cassandras, C. (2000). Optimal control of hybrid systems in manufacturing. *Proceedings of the IEEE, 88*(7), 1108-1123.

Pritsker, A.A., O'Reilly, J.J., and Laval, D.K. (1997). Simulation with Visual Slam and Awesim. John Wiley & Sons, Inc., New York.

Rabelo, L., Eskandari, H., Shaalan, T., & Helal, M. (2007). Value chain analysis using hybrid simulation and AHP. *International Journal of Production Economics, 105*(2), 536-547.

Rabelo, L., Helal, M., Jones, A., & Min, H. (2005). Enterprise simulation: A hybrid system approach. *International Journal of Computer Integrated Manufacturing, 18*(6), 498-508.

Raffo, D. (1996). *Modeling software processes quantitatively and assessing the impact of potential process changes on process performance.* Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, USA.

Richardson, G. (1985). Introduction to the system dynamics review. *System Dynamics Review, Albany, NY, Summer 1985, 1*(1), 1-3.

Richardson, G. (1986). Problems with causal - loop diagrams. *System Dynamics Review, 2*(2), 158-170.

Richardson, G. (1997). Problems in causal loop diagrams revisited. *System Dynamics Review, 13*(3), 247-252.

Richardson, G., & Pugh A. (1981). Introduction to system dynamics modeling with dynamo. Cambridge, MA, USA: MIT Press.

Rodrigues, A., & Bowers, J. (1996). System dynamics in project management: A comparative analysis with traditional methods. *System Dynamics Review, 12*(2), 121-139.

Rodrigues, A., & Williams, T. (1998). System dynamics in project management: Assessing the impacts of client behavior on project performance. *The Journal of the Operational Research Society, 49*(1), 2-15.

Rus, I., Collofello, J., & Lakey, P. (1999). Software process simulation for reliability management. *Journal of Systems and Software, 46*(2-3), 173-182.

Sawhney, A. (1996). Computerized tool for hierarchical simulation modeling. *Journal of Computing in Civil Engineering, 10*(2), 115-124.

Sawhney, A., & AbouRizk, S. (1995). HSM-Simulation–Based planning method for construction projects. *Journal of Construction Engineering and Management, 121*(3), 297-334.

Schultz, R., Slevin, D., & Pinto, J. (1987). Strategy and tactics in a process model of project implementation. *Interfaces, 17*(3), 34-46.

Shannon, R. (1998). Introduction to the art and science of simulation. *In Proceedings of the 30th Conference on Winter Simulation,* Los Alamitos, CA, USA. 7-14.

Shewchuk, J., & Chang, T. (1991). An approach to object-oriented discrete-event simulation of manufacturing systems. *Proceedings of the 23rd Conference on Winter Simulation,* Phoenix, AZ , USA. 302-311.

Shi, J., & AbouRizk, S. (1997). Resource-based modeling for construction simulation. *Journal of Construction Engineering and Management, 123*(1), 26-34.

Smith, J. (2003). Survey on the use of simulation for manufacturing system design and operation. *Journal of Manufacturing Systems, 22*(2), 157-171.

Smith, S., Osborne, J., & Forde, M. (1995). Analysis of earth-moving systems using discrete-event simulation. *Journal of Construction Engineering and Management, 121*(4), 388-396.

State University, VA

Steinman Jeff, 1993. "Breathing Time Warp." In proceedings of the 7th Workshop on Parallel and Distributed Simulation (PADS93). Vol. 23, No. 1,109-118.

Steinman, J. (1990). Multi-node test bed: A distributed emulation of space communications for the strategic defense system. Pittsburgh. *, 21*(3) 1111-1115.

Sterman, J. (1992). System dynamics modeling for project management. *Sloan School of Manag., MIT, on-Line Publication.*

Sterman, J. (1993). Breathing time warp. *ACM Transactions on Programming Languages and Systems (TOPLAS), , 23*(1) 109-118.

Sterman, J. (2000). *Business dynamics: Systems thinking and modeling for a complex world*. New York, USA: McGraw-Hill.

Sweetser, A. (1999). A comparison of system dynamics (SD) and discrete event simulation (DES). *17th International Conference of the System Dynamics Society and the 5th Australian & New Zealand Systems Conference,* 20-28.

Tako, A., & Robinson, S. (2009). Comparing model development in discrete event simulation and system dynamics. *Proceedings of the Winter Simulation Conference,* Austin, Texas. 979-991.

Tommelein, I., Carr, R., & Odeh, A. (1994). Assembly of simulation networks using designs, plans, and methods. *Journal of Construction Engineering and Management, 120*(4), 796-815.

Touran, A. (1990). Integration of simulation with expert systems. *Journal of Construction Engineering and Management, 116*(3), 480-493.

Umeda, S., & Zhang, F. (2008). Hybrid modeling approach for supply-chain simulation. *Lean Business Systems and Beyond,* , 453-460.

Venkateswaran, J., & Son, Y. (2005). Hybrid system dynamic—discrete event simulation-based architecture for hierarchical production planning. *International Journal of Production Research, 43*(20), 4397-4429.

Walsh, K., Hershauer, J., Walsh, T., Tommelein, I., & Sawhney, A. (2002). Lead time reduction via pre-positioning of inventory in an industrial construction supply chain. *Proceedings of the Winter Simulation Conference,* San Diego, CA, USA. 1737-1744.

Wolstenholme, R. (1990). *System enquiry: A system dynamics approach*. Chichester: John Wiley and Sons.

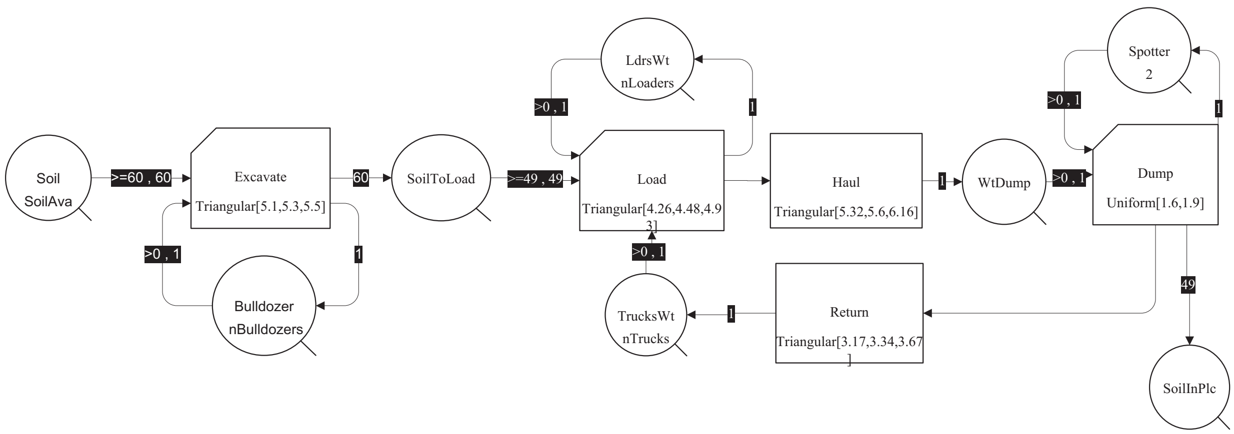
XJ-technologies, (2008) AnyLogic User's manual; www.xjtek.com. Retrieved

Zayed, T., & Halpin, D. (2001). Simulation of Concrete Batch Plant Production. *Journal of Construction Engineering and Management, 127*(2), 132-141.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2000). *Theory of modeling and simulation: Integrating discrete event and continuous complex dynamic systems* (2nd Ed., ed.) Academic press, USA.

Zulch, G., Jonsson, U., & Fischer, J. (2002). Hierarchical simulation of complex production systems by coupling of models. *International Journal of Production Economics, 77*(1), 39-51.

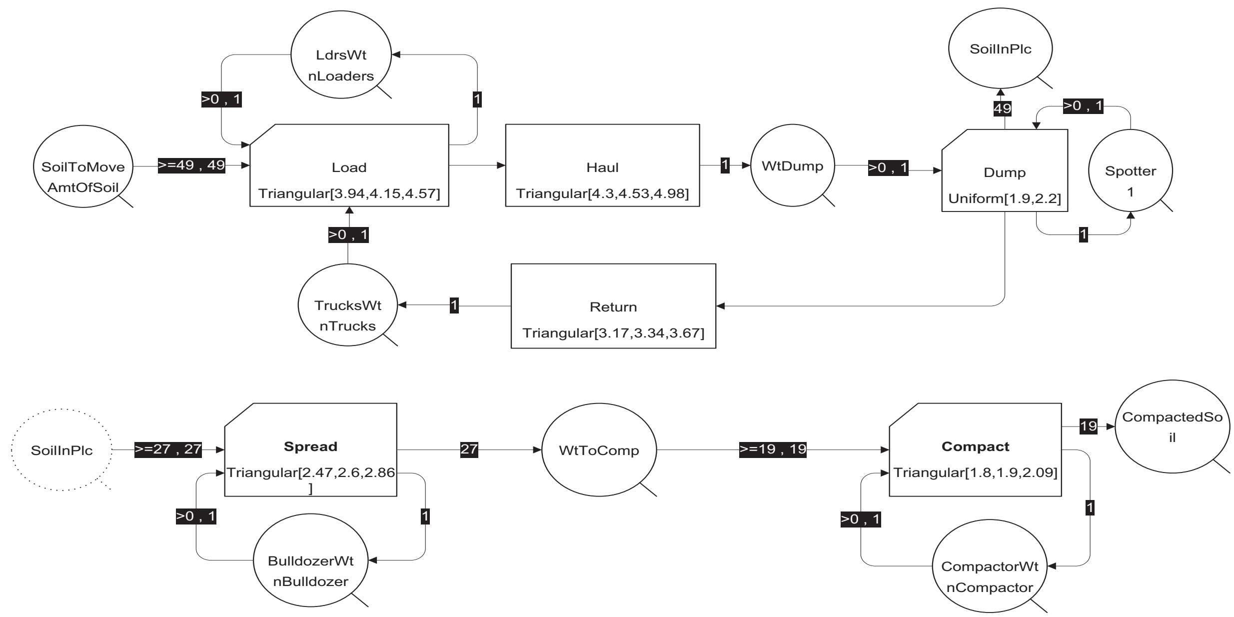
APPENDIX A: DISCRETE EVENT SIMULATION MODELS



nTrucks	Number of trucks	7
nLoaders	Number of loaders	2
nBulldozers	Number of Bulldozers	2
SoilAva	Amount of soil in m3	1038000

EXdrUt	Bulldozer utilization	1-(Bulldozer.AveCount)/nBulldozers
LdrUt	Loader utilization	1-(LdrsWt.AveCount)/nLoaders
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/
ProdRateD	Dumping production rate in m3/hr	49/Dump.AveInter*60
ProdRateE	Excavating production rate in m3/hr	60/Excavate.AveInter*60
Time	Time of operation in hours	SimTime/60

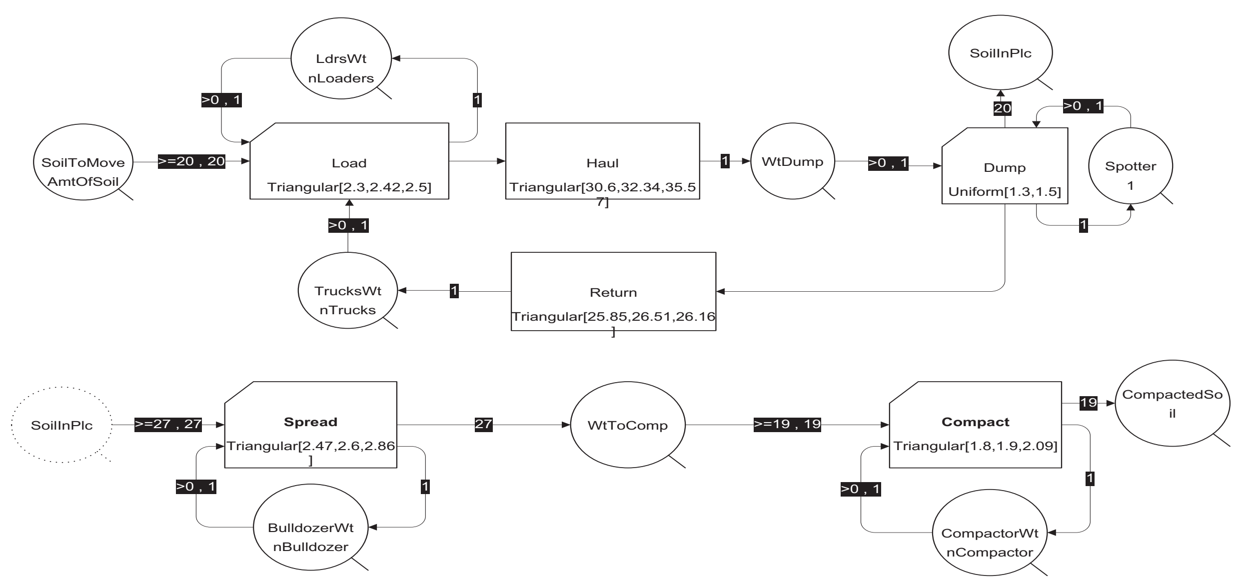
Excavation of Riverbed Model



nTrucks	Number of trucks	8
nLoaders	Number of loaders	2
AmtOfSoil	Amount of soil in m3	192700
nBulldozer	Number of Bulldozer	3
nCompactor	Number of CompactorWt	3

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

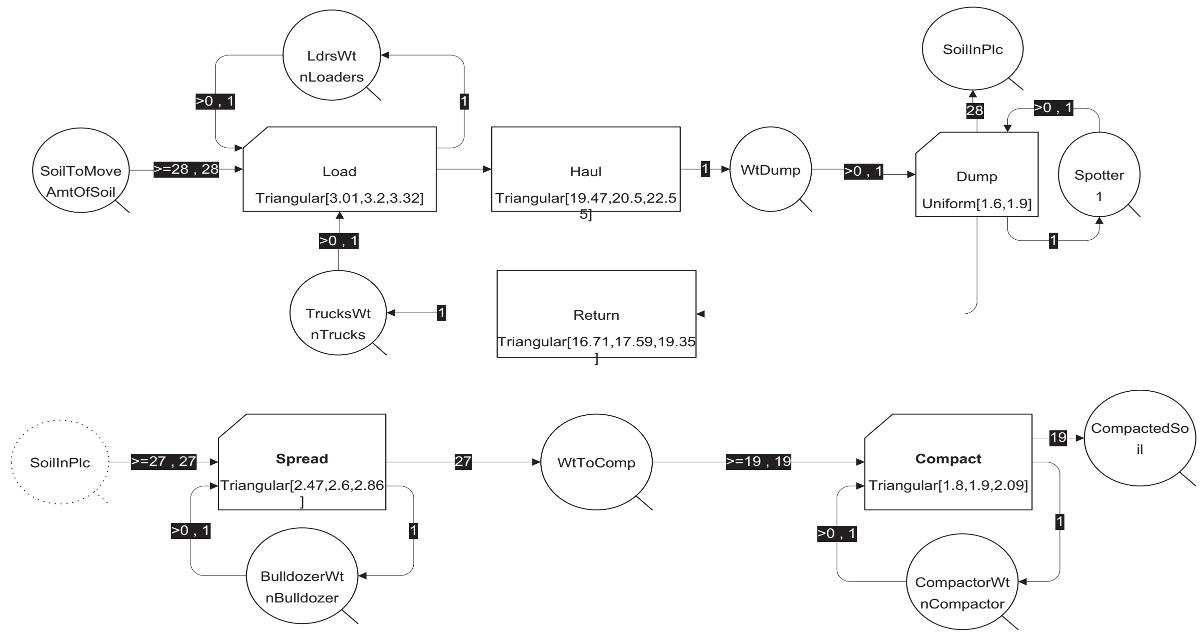
Backfill Rock11



nTrucks	Number of trucks	10
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	14500
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

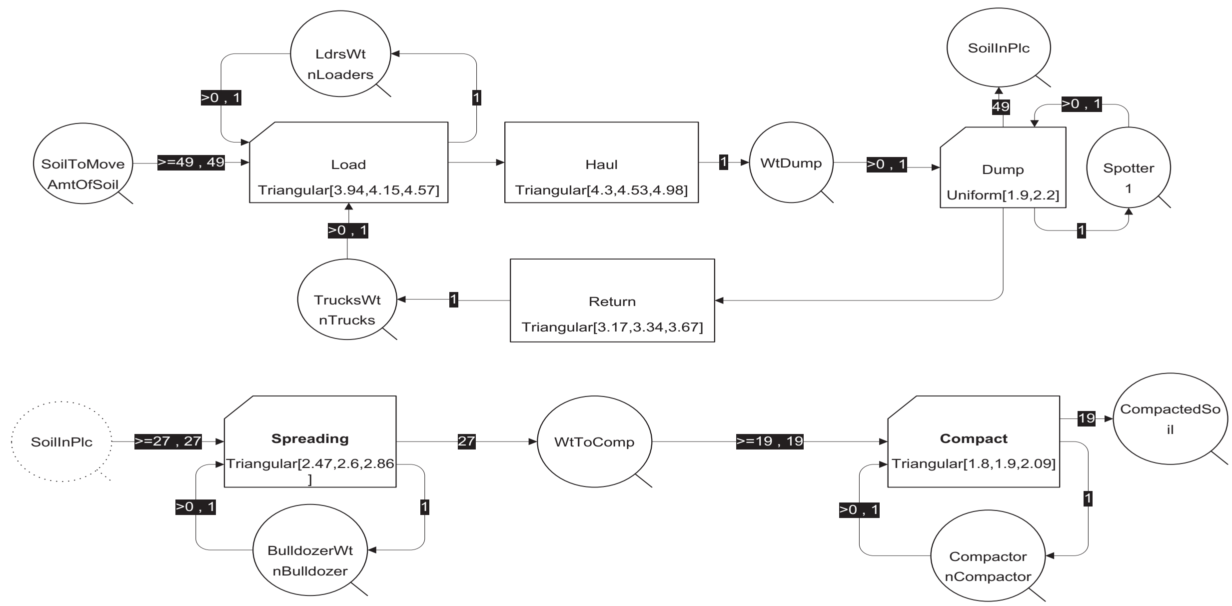
Backfill Granular12



nTrucks	Number of trucks	8
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	29200
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

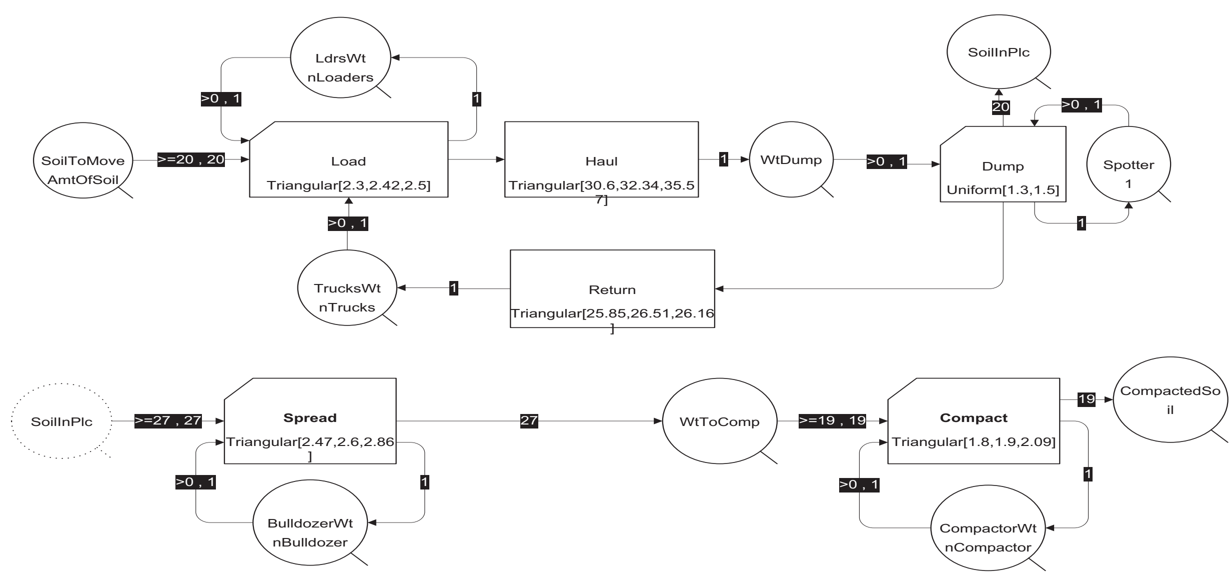
Backfill Moraine13



nTrucks	Number of trucks	8
nLoaders	Number of loaders	2
AmtOfSoil	Amount of soil in m3	3209400
nBulldozer	Number of Bulldozer	3
nCompactor	Number of CompactorWt	3

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

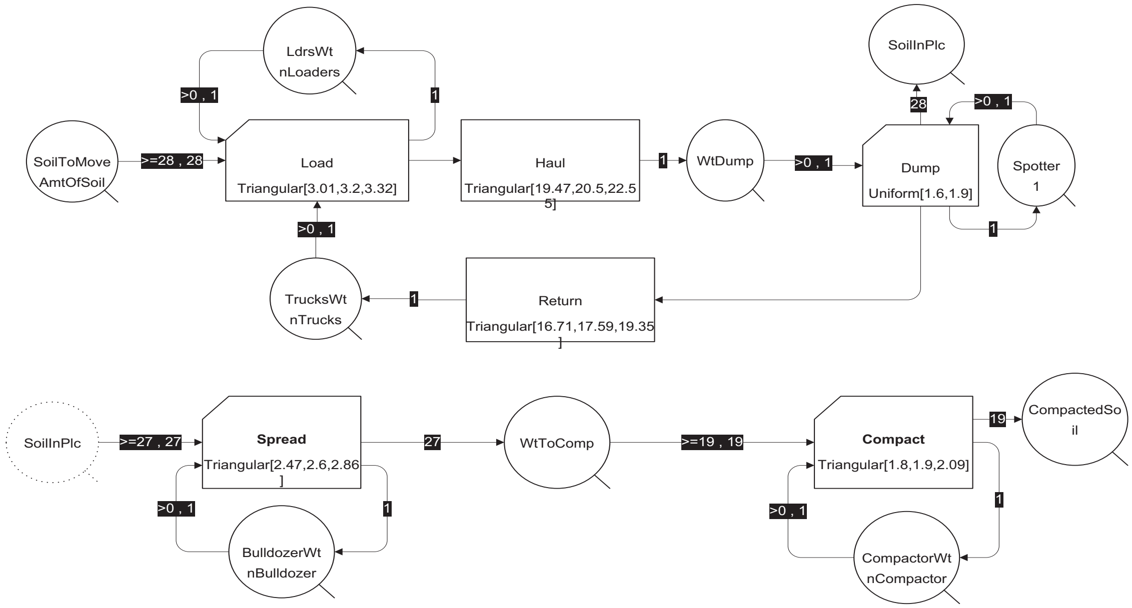
Backfill Rock21



nTrucks	Number of trucks	10
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	286500
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

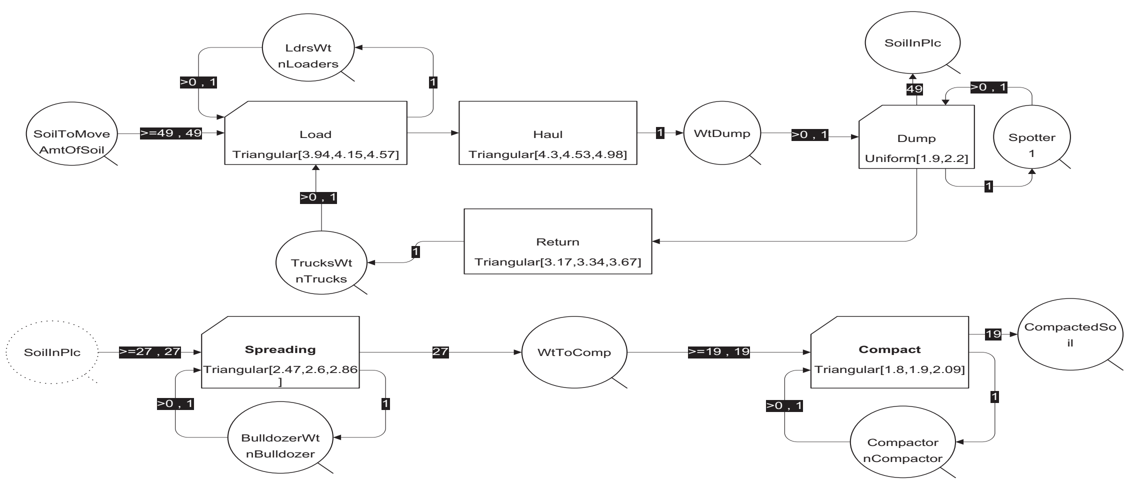
Backfill Granular22



nTrucks	Number of trucks	8
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	555900
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

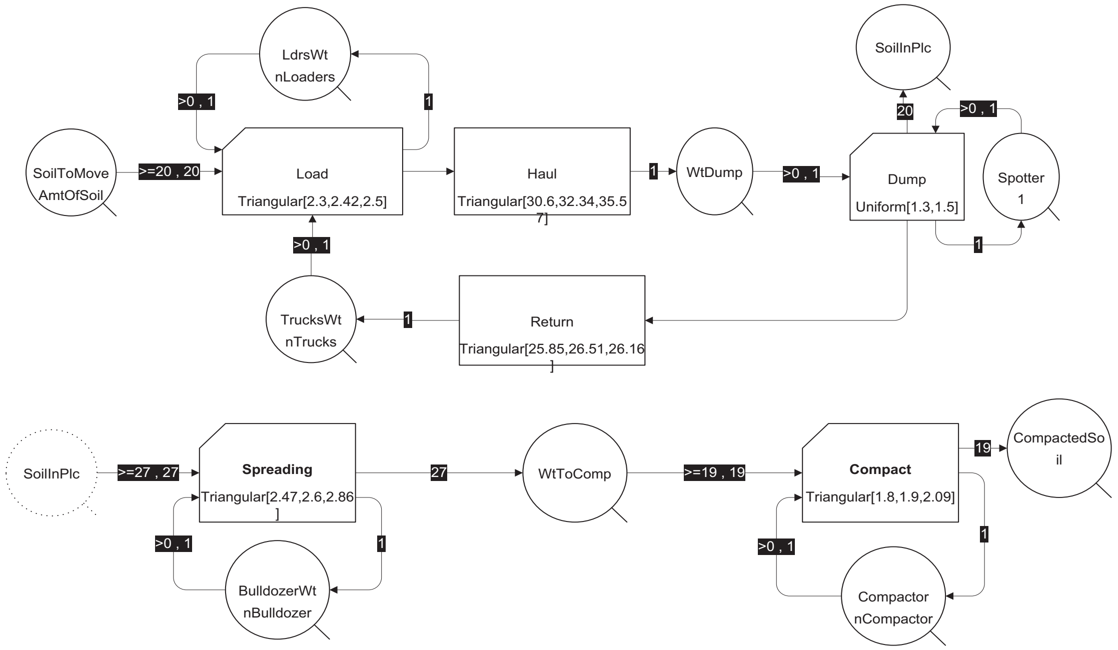
Backfill Moraine23



nTrucks	Number of trucks	8
nLoaders	Number of loaders	2
AmtOfSoil	Amount of soil in m3	1602900
nBulldozer	Number of Bulldozer	3
nCompactor	Number of CompactorWt	3

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

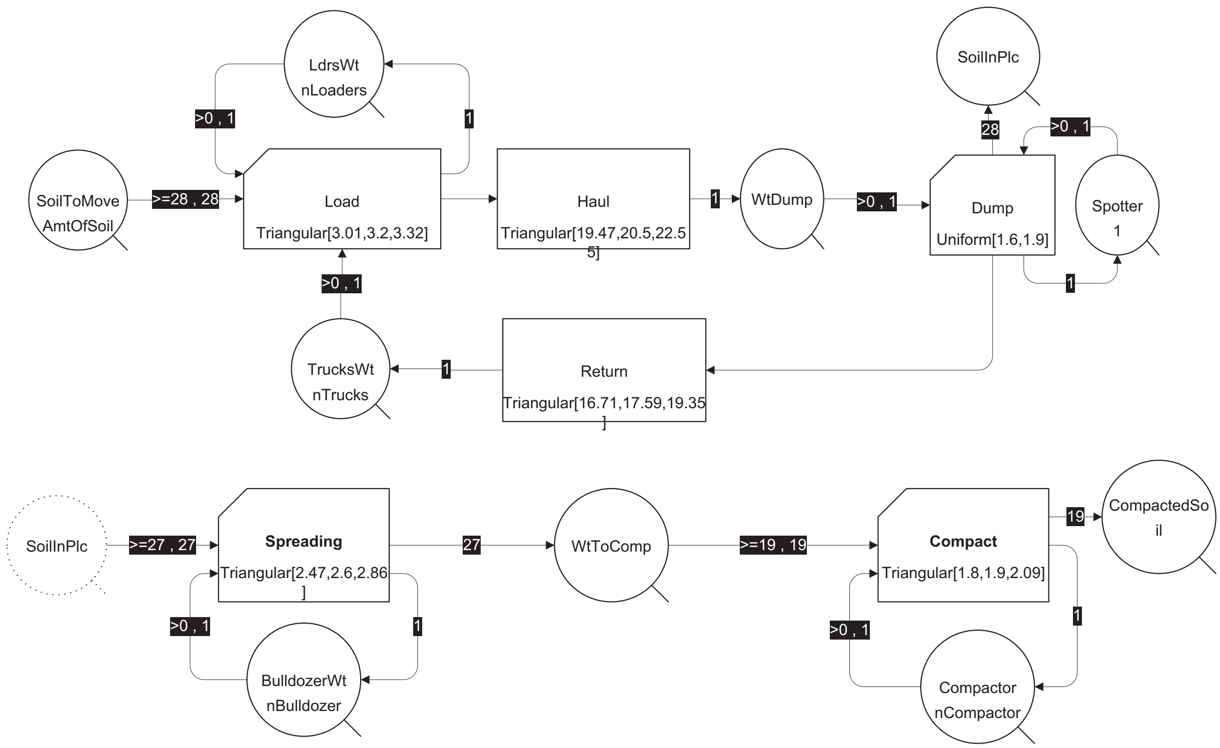
Backfill Rock31



nTrucks	Number of trucks	10
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	139000
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

Backfill Granular32



nTrucks	Number of trucks	10
nLoaders	Number of loaders	1
AmtOfSoil	Amount of soil in m3	269900
nBulldozer	Number of Bulldozer	1
nCompactor	Number of CompactorWt	1

LdrUt	Loader utilization	1-LdrsWt.AveCount
TrkUt	Truck utilization	1-(TrucksWt.AveCount+WtDump.AveCount)/nTrucks
ProdRateD	Dumping production rate in m3/hr	SoilInPlc.TotCount/Time
ProdRateS	Spreading production rate in m3/hr	WtToComp.TotCount/Time
ProdRateC	Compacting production rate in m3/hr	CompactedSoil.TotCount/Time
Time	Time of operation in hours	SimTime/60

Backfill Moraine33

APPENDIX B: SYSTEM DYNAMICS MODEL EQUATIONS

****
System Dynamics Model for Earthmoving Operations
****

Simulation Control Parameters

(003) FINAL TIME = 7000
Units: hr

(004) INITIAL TIME = 0
Units: hr

(005) SAVEPER = 1
Units: hr

****
.Excavation of Soil from Riverbed
****

(007) Average Excavation Time Ex = 1
Units: hr

(008) Dumping Rate considering Rainfall effect[Scope]=
IF THEN ELSE(Production is Interrupted , 0 , Max Dumping Rate[Scope])
Units: m3/hr

(009) Dumping Rate of Excavated Material from Riverbed After Considering Rainfall Effect
=IF THEN ELSE(Production is Interrupted , 0 , Max Dumping Rate Ex)
Units: m3/hr

(010) Excavated Soil= INTEG (Excavation Rate-Dumping Rate of Excavation,0)
Units: m3

(011) Excavation Rate=
MIN (Excavation Rate Considering Rainfall Effect ,
Soil to Excavate from Riverbed
/ Average Excavation Time Ex)*Loss Factor Due Depth
Units: m3/hr

(012) Excavation Rate Considering Rainfall Effect=
IF THEN ELSE(Production is Interrupted , 0 , Max Excavation Rate)
Units: m3/hr

(013) Excavation Scope = 1.038e+006
Units: m3

(014) Max Dumping Rate Ex=1284+ 28 * RANDOM NORMAL (-3, 3, 0, 1, 0)
Units: m3/hr

(015) Max Excavation Rate=Mean Ex+ "Standard Dev. Ex" * RANDOM NORMAL (-3, 3, 0, 1, 0)
Units: m3/hr

(016) Percentage of Dumped Soil Ex=
zidz (Soil Dumped Ex , Excavation Scope)
Units: Dmnl

(017) Productivity of dumped soil Ex=zidz(Soil Dumped Ex, Time)
Units: m3/hr

(018) Soil Dumped Ex= INTEG (Dumping Rate of Excavation,0)
Units: m3

(019) Soil to Excavate from Riverbed =
INTEG(- Excavation Rate , Excavation Scope)
Units: m3

****
.Hauling Operation of the Three Types of Soil
****

(022) {UTF-8}
Units: **undefined**

(023) Backfill Equipment Operator Skills[Scope]=1
Units: Dmnl

(024) Depth Impact on Excavation Productivity Lookup(
[(0,0)-(2e+006,10)],(0,1),(0.35,1),(0.6,0.9),(1,0.85))
Units: Dmnl

(025) Dumping Rate of Excavation=
MIN(Net Dumping Rate of Excavation from Riverbed , Excavated Soil/Average Excavation Time Ex
)
Units: m3/hr

(026) "Eff. Fatigue on productivity Lookup"(
[(0,0)-(1000,2)],(0,1),(1,1),(1.2,0.95),(1.5,0.93),(2,0.9),(3,0.89),(4,0.85),(1000,0.85))
Units: Dmnl

(027) Effect of Fatigue on Quality Lookup(
[(0,0)-(4,2)],(0,1),(1,1),(1.2,0.99),(1.5,0.95),(2,0.93),(3,0.92),(4,0.9))
Units: Dmnl

(028) Effect of Schedule Pressure on Quality Lookup(
[(0,0)-(8000,2)],(0,1),(1,1),(1.05,1),(1.1,0.97),(1.15,0.96),(1.2,0.95),(
1.3,0.9),(1.5,0.85),(2,0.83),(7000,0.8))
Units: Dmnl

(029) Equipment: Haulers, Loaders, Bulldozers, Compactors

(030) Equipment Age Factor[Scope]=0.9
Units: Dmnl

(031) Event:
e1, e2, e3, e4, e5, e6, e7, e8, e9, e10, e11, e12,e13, e14, e15, e16, e17
, e18, e19, e20, e21, e22, e 23, e24

(032) Excavation Equipment Age Factor=0.9
Units: Dmnl

(033) Excavation Equipment Operator Skills=1
Units: Dmnl

(034) Excavation Finished=
IF THEN ELSE(Excavation Progress>=0.9998, 1 , 0)
Units: Dmnl

(035) Excavation Progress=
IF THEN ELSE(Excavation Scope=0 , 0 , Soil Dumped Ex/Excavation
Scope)
Units: Dmnl

(036) Final Project Completion Duration=Project is Done
Units: Dmnl

(037) Fraction Completed=
"Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted"/Total
Scope
Units: Dmnl

(038) Fraction Reduction in Dumping productivity due deteriorating Road Condition Effect of Road Condition
= (Reduction in Dumping productivity due deterioration of Road Condition Effect of Road Condition Lookup1
(MODULO(Time, Repeat Time))+ (Shift - 1))
Units: Dmnl

(039) Impact of Fatigue on Productivity=
"Eff. Fatigue on productivity Lookup"(Overtime Fraction)
Units: Dmnl

(040) Impact of Fatigue on Quality=
Effect of Fatigue on Quality Lookup(Overtime Fraction)
Units: Dmnl

(041) Impact of Schedule Pressure on Quality=
Effect of Schedule Pressure on Quality Lookup(Schedule Pressure)
Units: Dmnl

(042) Loss Factor Due Depth=
Depth Impact on Excavation Productivity Lookup(Percentage Excavated)
Units: Dmnl

(043) Loss of Dumping Rate of Excavated Soil Due to road condition=
Dumping Rate of Excavated Material from Riverbed After Considering Rainfall Effect
*Net Loss Fraction ex
Units: m3/hr

(044) Max Dumping Rate[Rock11]=
1393 + 35*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Granular12]=
187+ 15*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Moraine13]=
304+ 22*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Rock21]=
1393+ 35*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Granular22]=
190+ 16*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Moraine23]=
306+ 22*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Rock31]=
1393+ 37*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Granular32]=
190+ 17*RANDOM NORMAL(-2,2,0,1,0)
Max Dumping Rate[Moraine33]=
381+ 22*RANDOM NORMAL(-2,2,0,1,0)
Units: m3/hr

(045) Mean Ex=
GET XLS CONSTANTS('DES_Model_Sender_Variables.xls' , 'sheet1' , 'B7')
Units: m3/hr

(046) Multiplier=
INTEG(Repeat Pulse,0)
Units: Dmnl

(047) Net Compaction Rate[Scope]=

Compaction Rate Considering Rainfall Effect[Scope]*Backfill Equipment Operator Skills[Scope]*Equipment Age Factor[Scope]
Units: m3/hr

(048) Net Dumping Rate[Scope]=
(Dumping Rate considering Rainfall effect[Scope]-Loss of Dumping productivity Due to Road Condition
[Scope])*Backfill Equipment Operator Skills [Scope]*Equipment Age Factor[Scope]
Units: m3/hr

(049) Net Dumping Rate of Excavation from Riverbed=
(Dumping Rate of Excavated Material from Riverbed After Considering Rainfall Effect
-Loss of Dumping Rate of Excavated Soil Due to road condition
)*Excavation Equipment Operator Skills*Excavation Equipment Age Factor
Units: m3/hr

(050) Net Loss Fraction[Scope]= 1+Fraction Reduction in Dumping productivity due deteriorating Road Condition Effect of Road Condition
Units: Dmnl

(051) Net Loss Fraction ex=1+Fraction Reduction in Dumping productivity due deteriorating Road Condition Effect of Road Condition
Units: Dmnl

(052) Net Spreading Rate[Scope]=
Backfill Equipment Operator Skills[Scope]*Spreading Rate Considering Rainfall Effect
[Scope]*Equipment Age Factor[Scope]
Units: m3/hr

(053) Number of Months=
MODULO(Time, Working Hours in a Month)+1
Units: hr

(054) Overtime Fraction=
Overtimne Lookup(Schedule Pressure)
Units: Dmnl

(055) Overtimne Lookup(
[(0,0)-
(100,20)],(0,0),(1,0),(1.2,0.2),(1.3,0.3),(1.5,0.5),(2,2),(10,5),(20,5),(60,5),(100,5))
Units: Dmnl

(056) Percentage Excavated=
zidz(Total Excavated , Excavation Scope)
Units: Dmnl

(057) Predicted Overtime=

Overtime Fraction*Time Available
Units: hr

(058) prereqtask <-> Scope

(059) Project Finished=
IF THEN ELSE(Project Progress>0.9999, 0 , 1)
Units: Dmnl

(060) Project is Done=IF THEN ELSE(Project was Done :AND:
Fraction Completed > Restart Fraction,1,
IF THEN ELSE(Fraction Completed >= 1,1,0))
Units: Dmnl

(061) Project Progress=IF THEN ELSE(Total Scope=0, 0 , "Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted"
/Total Scope)
Units: Dmnl

(062) Project was Done= DELAY FIXED (Project is Done, 0,0)
Units: Dmnl

(063) Reduction in Dumping productivity due deterioration of Road Condition Effect of Road Condition Lookup1([(0,0)-
(6000,100)],(0,0),(0,0),(1,0),(5,0),(10,0),(20,0.02),(30,0.03),(40
,0.05),(50,0.055),(60,0.06),(70,0.06),(80,0.062))
Units: Dmnl

(064) Repeat Pulse=
PULSE TRAIN(INITIAL TIME,TIME STEP,Repeat Time,FINAL TIME)
Units: Dmnl

(065) Repeat Time=80
Units: hr

(066) Restart Fraction=0.9
Units: Dmnl

(067) Scope:Rock11, Granular12, Moraine13, Rock21, Granular22, Moraine23, Rock31, Granular32
, Moraine33

(068) Shift=MAX (0 , (Multiplier -1)* Reduction in Dumping productivity due deterioration of Road Condition Effect of Road Condition Lookup1
(INITIAL TIME))
Units: Dmnl

(069) "Standard Dev. Ex"=

GET XLS CONSTANTS('DES_Model_Sender_Variables.xls' , 'sheet1' , 'C7')
Units: m3/hr

(070) Total Excavated= INTEG (Excavation Rate,0)
Units: m3

(071) Workforce:
Managers, Engineers, Equipment Operators, Labor

(072) Working Hours in a Month=240
Units: hr

****
.Productivity Loss Due to Road Condition
****

(074) Loss of Dumping productivity Due to Road Condition[Scope]=
Dumping Rate considering Rainfall effect[Scope]* Net Loss Fraction[Scope]
Units: m3/hr

(075) Percentage of reduction in Dumping Rate Due to Road=
Fraction Reduction in Dumping productivity due deteriorating Road Condition Effect of Road Condition
*100
Units: Dmnl

****
.**Project Costs and Resources**
****

(077) All Equipment[Equipment]=80, 12, 15, 15
Units: Equipment

(078) Available Fund During Project Progress=
IF THEN ELSE(Available or Short of Funds>=0, Available or Short of Funds,0)
Units: $

(079) Available or Short of Funds=
Project Authorized Budget-Total Accumulated Project Cost
Units: $

(080) Compaction Rate[Scope]=
IF THEN ELSE(Scope Task is Active C[Scope],
Net Compaction Rate[Scope],0)
Units: m3/hr

(081) Cost Per Unit Soil[Scope]=

4, 3, 2,4, 3, 2,4, 3, 2
Units: $/m3

(082) Equipment Cost[Equipment]= INTEG (Equipment Cost Rate[Equipment],0)
Units: $

(083) Equipment Cost Rate[Equipment]=
All Equipment[Equipment]*Equipment Unit Cost[Equipment]
Units: $/hr

(084) Equipment Unit Cost[Equipment]=25, 60, 40, 40
Units: $/hr/Equipment

(085) Final Work Completed[Scope]= INTEG (Productivity Rate[Scope]+Rework rate[Scope],0)
Units: m3

(086) Initial Workforce[Workforce]=120
Units: workforce

(087) Payment in Unit Time[Workforce]=60, 50, 30, 20
Units: $/hr/Person

(088) Productivity Rate[Scope]= (Quality Check Process Rate[Scope]-Error Rate[Scope])*Impact of Schedule Pressure on Productivity
*Impact of Fatigue on Productivity
Units: m3/hr

(089) Project Authorized Budget=5.5e+007
Units: $

(090) Rework rate[Scope]=MIN(Perceived Rework Rate[Scope], Rework process rate[Scope])
Units: m3/hr

(091) Short of Fund During Project Progress=
IF THEN ELSE(Available or Short of Funds<0, Available or Short of Funds,
0)
Units: $

(092) Soil Cost[Scope]= INTEG (Soil Cost Rate[Scope],0)
Units: $

(093) Soil Cost Rate[Scope]=
Cost Per Unit Soil[Scope]*(Productivity Rate[Scope]+Rework rate[Scope])
Units: $/hr

(094) Total Accumulated Project Cost=
Total Equipment Cost+Total Soil Cost+Total Workforce Cost

Units: $

(095) Total Equipment Cost=SUM(Equipment Cost[Equipment!])
Units: $

(096) Total Soil Cost=SUM(Soil Cost[Scope!])
Units: $

(097) Total Workforce Cost=SUM(Workforce Cost[Workforce!])
Units: $

(098) Workforce Cost[Workforce]= INTEG (Workforce Cost Rate[Workforce],0)
Units: $

(099) Workforce Cost Rate[Workforce]=Payment in Unit Time[Workforce]*Initial Workforce[Workforce]
Units: $/hr

****
.Schedule Pressure and Productivity
****

(102) Actual Final Released Productivity=zidz("Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted", Time)
Units: m3/hr

(103) Forecasted Productivity to Complete=
(((MAX(0, Time needed to reached planned productivity-Time))/Time needed to reached planned productivity)*Perceived Productivity)+(MIN(1 , Time/Time needed to reached planned productivity))*Actual Final Released Productivity
Units: m3/hr

(104) Perceived Productivity=Total Scope/Project Deadline
Units: m3/hr

(105) Project Deadline=4620
Units: hr

(106) Schedule Index used to adjust released productivity to be same as the planned=0.1
Units: Dmnl

(107) Schedule Pressure=
MAX(XIDZ(Time Required , Time Available , Time Required/TIME STEP) , 0)
Units: Dmnl

(108) Scope Size[Scope]=

192700, 14500, 29200, 3.2094e+006, 286500, 555900, 1.6029e+006, 139000, 269900
Units: m3

(109) Soil to Compact[Scope]= INTEG (Compaction Rate[Scope],Spread Soil[Scope])
Units: m3

(110) Time Available=MAX(1, Project Deadline-Time)
Units: hr

(111) Time needed to reached planned productivity=
Project Deadline*Schedule Index used to adjust released productivity to be same as the planned
Units: hr

(112) Time Required=IF THEN ELSE(zidz(Total Work Not Done, Forecasted Productivity to Complete)>=0 , zidz(Total Work Not Done, Forecasted Productivity to Complete) , 0)
Units: hr

(113) TIME STEP = 1
Units: hr

(114) "Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted"=
SUM(Final Work Completed[Scope!])
Units: m3

(115) Total Rework=SUM(Rework[Scope!])
Units: m3

(116) Total Scope=SUM(Scope Size[Scope!])
Units: m3

(117) Total Soil Compacted and Ready for Quality Check=
SUM(Soil Compacted and ready for Quality Check[Scope!])
Units: m3

(118) Total Soil Dumped=SUM(Soil Dumped[Scope!])
Units: m3

(119) Total Soil Spread=SUM(Soil Spread[Scope!])
Units: m3

(120) Total Work Not Done=Total Scope-(Total Rework+Total Soil Compacted and Ready for Quality Check+"Total Project Work of Soil, Hauled, Dumped, Spread, and Compacted")
Units: m3

****
.**Weather Impact**
****

(123) Compaction Rate Considering Rainfall Effect[Scope]=
IF THEN ELSE(Production is Interrupted , 0 , Max Compaction Rate[Scope]
)
Units: m3/hr

(124) Event ID[Event]=Event
Units: Dmnl

(125) Interruption Start Time in schedule[Event]=
80, 400, 900, 1200, 1600, 1900, 2200, 2500, 2900, 3200, 3600, 4000, 4200,
4700, 5000, 5350, 5700, 5900, 6400, 6700, 7000, 7350, 7800, 8100
Units: hr

(126) Is Interrupted[Event]=PULSE(Interruption Start Time in schedule[Event] , Total Interrupt Duration of Work in Hours for Single Month Reference Model [Event])
Units: Dmnl

(127) Max Compaction Rate[Rock11]=
1393 + 36*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Granular12]=
187+ 16*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Moraine13]=
304+ 19*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Rock21]=
1393+ 33*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Granular22]=
190+ 16*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Moraine23]=
306+ 23*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Rock31]=
1393+ 37*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Granular32]=
190+16*RANDOM NORMAL(-2,2,0,1,0)
Max Compaction Rate[Moraine33]=
381+ 25*RANDOM NORMAL(-2,2,0,1,0)
Units: m3/hr

(128) Max Spreading Rate[Rock11]=
1393 + 35*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Granular12]=
187+ 15*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Moraine13]=
304+ 20*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Rock21]=

1393+ 32*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Granular22]=
190+ 16*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Moraine23]=
306+ 22*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Rock31]=
1393+ 35*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Granular32]=
190+ 16*RANDOM NORMAL(-2,2,0,1,0)
Max Spreading Rate[Moraine33]=
381+ 27*RANDOM NORMAL(-2,2,0,1,0)
Units: m3/hr

(129) Production is Interrupted=
IF THEN ELSE(VMAX(Is Interrupted[Event!]) >0 , 1 , 0)
Units: Dmnl

(130) Spreading Rate Considering Rainfall Effect[Scope]=
IF THEN ELSE(Production is Interrupted , 0 , Max Spreading Rate[Scope])
Units: m3/hr

(131) Total Interrupt Duration of Work in Hours for Single Month Reference Model [Event]=66, 74.88, 68, 72, 68, 66, 80, 76.8, 66, 74.88, 68, 72, 68, 66, 80, 76.8, 66, 74.88, 68, 72, 68, 66, 80, 76.8
Units: hr

****
.Workflow Structure
****

(133) Average time[Scope]=1
Units: hr

(134) "Avg, QP Duration"[Scope]=1
Units: hr

(135) Dumped Soil[Scope]=Soil Dumped[Scope]
Units: m3

(136) Dumping Rate[Scope]=MIN(IF THEN ELSE(Scope Task is Active D[Scope], Net Dumping Rate[Scope],0), Soil to Haul[Scope]/ Average time[Scope])
Units: m3/hr

(137) Effect of Schedule Pressure on Productivity Lookup([(0,0)-(400,2)],(0,1),(1,1),(1.05,1),(1.1,0.97),(1.15,0.96),(1.2,0.95),(1.3,0.9),(1.5,0.85),(2,0.83),(300,0.8)
Units: Dmnl

(138) Error Generation Rate[Scope]=0.05

Units: Dmnl

(139) Error Rate[Scope]=
Quality Check Process Rate[Scope]*Error Generation Rate[Scope]
Units: m3/hr

(140) Impact of Schedule Pressure on Productivity=
Effect of Schedule Pressure on Productivity Lookup(Schedule Pressure)
Units: Dmnl

(141) Perceived Quality Rate[Scope]=
1350, 1350,1350, 1350,1350, 1350,1350, 1350,1350
Units: m3/hr

(142) Perceived Rework Rate[Scope]=100
Units: m3/hr

(143) Percentage of Compacted Soil[Scope]=
zidz ((Soil Compacted and ready for Quality Check[Scope]+Final Work Completed
[Scope]+Rework[Scope])
, Spread Soil[Scope])
Units: Dmnl

(144) percentage of Dumped Soil[Scope]=zidz (Soil Dumped[Scope], Scope Size[Scope])
Units: Dmnl

(145) Percentage of Spread Soil[Scope]=zidz (Soil Spread[Scope], Dumped Soil[Scope])
Units: Dmnl

(146) Quality Check Process Rate[Scope]=
MIN(Perceived Quality Rate[Scope], Quality Process Rate[Scope])*Impact of Fatigue on Quality*Impact of Schedule Pressure on Quality
Units: m3/hr

(147) Quality Process Rate[Scope]=
zidz(Soil Compacted and ready for Quality Check[Scope], "Avg, QP Duration"[Scope])
Units: m3/hr

(148) Rework[Scope]= INTEG (
Error Rate[Scope]-Rework rate[Scope],0)
Units: m3

(149) Rework process rate[Scope]=
Rework[Scope]/Rework Process Rate Duration[Scope]
Units: m3/hr

(150) Rework Process Rate Duration[Scope]=1

Units: hr

(151) Scope ID[Scope]=INITIAL(Scope)
Units: m3

(152) Scope Start Percentage of Compaction[Scope]=0.5
Units: Dmnl

(153) Scope Task Flag S[Rock11]=1
Scope Task Flag S[Granular12]=SAMPLE IF TRUE(
Percentage of Spread Soil[Rock11] >= Start Scope Task Percentage Spreading [Granular12], 1 , 0)

Scope Task Flag S[Moraine13]=SAMPLE IF TRUE(
Percentage of Spread Soil[Granular12] >= Start Scope Task Percentage Spreading [Moraine13], 1 , 0)

Scope Task Flag S[Rock21]=SAMPLE IF TRUE(
Percentage of Spread Soil[Moraine13] >= Start Scope Task Percentage Spreading[Rock21], 1 , 0)

Scope Task Flag S[Granular22]=SAMPLE IF TRUE(
Percentage of Spread Soil[Rock21] >= Start Scope Task Percentage Spreading[Granular22], 1 , 0)

Scope Task Flag S[Moraine23]=SAMPLE IF TRUE(
Percentage of Spread Soil[Granular22] >= Start Scope Task Percentage Spreading[Moraine23], 1 , 0)

Scope Task Flag S[Rock31]=SAMPLE IF TRUE(
Percentage of Spread Soil[Moraine23] >= Start Scope Task Percentage Spreading[Rock31], 1 , 0)

Scope Task Flag S[Granular32]=SAMPLE IF TRUE(
Percentage of Spread Soil[Rock31] >= Start Scope Task Percentage Spreading[Granular32], 1 , 0)

Scope Task Flag S[Moraine33]=SAMPLE IF TRUE(
Percentage of Spread Soil[Granular32] >= Start Scope Task Percentage Spreading[Moraine33], 1 , 0)

Units: Dmnl

(154) Scope Task is Active C[Scope]=
IF THEN ELSE (
:NOT: Scope Task is Done C[Scope]
:AND: Scope Task Start Flag C[Scope] > 0.5, 1, 0)
Units: Dmnl

(155) Scope Task is Active D[Scope]=
IF THEN ELSE (
:NOT: Scope Task is Done D[Scope]
:AND: Start Task Flag[Scope] > 0.5, 1, 0)
Units: Dmnl

(156) Scope Task is Active S[Scope]=
IF THEN ELSE (
:NOT: Scope Task is Done S[Scope]
:AND: Scope Task Flag S[Scope] > 0.5, 1, 0)
Units: Dmnl

(157) Scope Task is Done C[Scope]=
IF THEN ELSE((Soil Compacted and ready for Quality Check[Scope]+Rework[Scope
]+Final Work Completed[Scope])>= Spread Soil[Scope], 1, 0)
Units: Dmnl

(158) Scope Task is Done D[Scope]=
IF THEN ELSE(Soil Dumped[Scope] >=0.9999*Scope Size[Scope], 1, 0)
Units: Dmnl

(159) Scope Task is Done S[Scope]=
IF THEN ELSE(Soil Spread[Scope] >= Dumped Soil[Scope], 1, 0)
Units: Dmnl

(160) Scope Task Start Flag C[Rock11]=1
Scope Task Start Flag C[Granular12]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Rock11] >= Scope Start Percentage of Compaction[Granular12], 1 , 0)

Scope Task Start Flag C[Moraine13]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Granular12] >= Scope Start Percentage of Compaction[Moraine13], 1 , 0)

Scope Task Start Flag C[Rock21]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Moraine13] >= Scope Start Percentage of Compaction[Rock21], 1 , 0)

Scope Task Start Flag C[Granular22]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Rock21] >= Scope Start Percentage of Compaction[Granular22], 1 , 0)

Scope Task Start Flag C[Moraine23]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Granular22] >= Scope Start Percentage of Compaction[Moraine23], 1 , 0)

Scope Task Start Flag C[Rock31]=SAMPLE IF TRUE(

Percentage of Compacted Soil[Moraine23] >= Scope Start Percentage of Compaction[Rock31], 1 , 0)

Scope Task Start Flag C[Granular32]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Rock31] >= Scope Start Percentage of Compaction[Granular32], 1 , 0)

Scope Task Start Flag C[Moraine33]=SAMPLE IF TRUE(
Percentage of Compacted Soil[Granular32] >= Scope Start Percentage of Compaction[Moraine33], 1 , 0)

Units: Dmnl

(161) Soil Compacted and ready for Quality Check[Scope]= INTEG (
Compaction Rate[Scope]-Productivity Rate[Scope]-Error Rate[Scope],0)
Units: m3

(162) Soil Dumped[Scope]= INTEG (
Dumping Rate[Scope],0)
Units: m3

(163) Soil Spread[Scope]= INTEG (Spreading Rate[Scope],0)
Units: m3

(164) Soil to Haul[Scope] = INTEG(-Dumping Rate[Scope],Scope Size[Scope])
Units: m3

(165) Soil to Spread[Scope]= INTEG (Spreading Rate[Scope],Dumped Soil[Scope])
Units: m3

(166) Spread Soil[Scope]=Soil Spread[Scope]
Units: m3

(167) Spreading Rate[Scope]=
IF THEN ELSE(Scope Task is Active S[Scope],
Net Spreading Rate[Scope],0)
Units: m3/hr

(168) Start Scope Task Percentage Dumping[Scope]=0.5
Units: Dmnl

(169) Start Scope Task Percentage Spreading[Scope]=0.5
Units: Dmnl

(170) Start Task Flag[Rock11]=1
Start Task Flag[Granular12]=SAMPLE IF TRUE(
percentage of Dumped Soil[Rock11] >= Start Scope Task Percentage Dumping[Granular12], 1 , 0)

Start Task Flag[Moraine13]=SAMPLE IF TRUE(
percentage of Dumped Soil[Granular12] >= Start Scope Task Percentage Dumping[Moraine13], 1 , 0)

Start Task Flag[Rock21]=SAMPLE IF TRUE(
percentage of Dumped Soil[Moraine13] >= Start Scope Task Percentage Dumping[Rock21], 1 , 0)

Start Task Flag[Granular22]=SAMPLE IF TRUE(
percentage of Dumped Soil[Rock21] >= Start Scope Task Percentage Dumping[Granular22], 1 , 0)

Start Task Flag[Moraine23]=SAMPLE IF TRUE(
percentage of Dumped Soil[Granular22] >= Start Scope Task Percentage Dumping[Moraine23], 1 , 0)

Start Task Flag[Rock31]=SAMPLE IF TRUE(
percentage of Dumped Soil[Moraine23] >= Start Scope Task Percentage Dumping [Rock31], 1 , 0)

Start Task Flag[Granular32]=SAMPLE IF TRUE(
percentage of Dumped Soil[Rock31] >= Start Scope Task Percentage Dumping[Granular32], 1 , 0)

Start Task Flag[Moraine33]=SAMPLE IF TRUE(
percentage of Dumped Soil[Granular32] >= Start Scope Task Percentage Dumping [Moraine33], 1 , 0)

Units: Dmnl

APPENDIX C: VB.NET CODE OF HISIM

```
Imports System.IO

Public Class Form1
    Public StroboApp As Object
    Public File_content As String
    Public variable_names(20) As String
    Public var_count As Integer


    Public Function GetStrobo() As Object
        On Error Resume Next
        If (StroboApp Is Nothing) Then
            StroboApp = _
              CreateObject("Stroboscope.Document")
        End If
        GetStrobo = StroboApp
    End Function
    'releases Stroboscope
    Public Sub ReleaseStrobo()
        StroboApp = Nothing
    End Sub

    Private Sub Select .STT File (DES)_Click(sender As System.Object, e As
System.EventArgs) Handles Select .STT File (DES).Click

        OpenFileDialog1.ShowDialog()


    End Sub

    Private Sub OpenFileDialog1_FileOk(sender As System.Object, e As
System.ComponentModel.CancelEventArgs) Handles OpenFileDialog1.FileOk

        Dim FILE_NAME As String = OpenFileDialog1.FileName.ToString()
        TextBox1.Text = FILE_NAME

        Dim objReader As New System.IO.StreamReader(FILE_NAME)
        Dim lineStr As String
        Dim i As Integer = 0

        lineStr = objReader.ReadLine
        Do Until lineStr Is Nothing
            If lineStr.Contains("COLLECTOR") Then
                Dim lineParts As String() = lineStr.Split(" ")
                variable_names(i) = lineParts(1).Substring(0,
lineParts(1).Length - 6)
                i += 1
            End If
            lineStr = objReader.ReadLine
        Loop
```

```
        var_count = i
        objReader.Close()

        Dim objReader2 As New System.IO.StreamReader(FILE_NAME)
        File_content = objReader2.ReadToEnd
        objReader2.Close()

        TextBox2.Text = File_content


    End Sub

    Private Sub Run Discrete Simulation_Click(sender As System.Object, e As
System.EventArgs) Handles Run Discrete Simulation.Click

        Dim nResult As Integer
        Dim oExcel As Object
        Dim oBook As Object
        Dim oSheet As Object
        'On Error Resume Next
        'set name of client running Stroboscope

        oExcel = CreateObject("Excel.Application")
        oBook = oExcel.Workbooks.Add
        'Add data to cells of the first worksheet in the new workbook
        oSheet = oBook.Worksheets(1)
        oSheet.Range("A" & 1).Value = "Variable"
        oSheet.Range("B" & 1).Value = "Average"
        oSheet.Range("C" & 1).Value = "Standard Dev"
        oSheet.Range("D" & 1).Value = "Minimum"
        oSheet.Range("E" & 1).Value = "Maximum"

        GetStrobo()
        StroboApp.ClientVersion("test Strobo")

        nResult = StroboApp.RunStatements(File_content)

        For i = 0 To var_count - 1
            oSheet.Range("A" & i + 2).Value = variable_names(i)
            oSheet.Range("B" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.AveVal")
            oSheet.Range("C" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.SDVal")
            oSheet.Range("D" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.MinVal")
            oSheet.Range("E" & i + 2).Value =
StroboApp.EvaluateExpression(variable_names(i) & "_clt.MaxVal")
        Next

        StroboApp.CloseAllOutputs()
        StroboApp.EndModel()

        StroboApp.SetAskRelease(False)
```

```
        Dim filename As String = Microsoft.VisualBasic.DateAndTime.Day(Now) &
"_" & Month(Now) & "_" & Year(Now) & "$" & Hour(Now) & "_" & Minute(Now) & "_"
& Second(Now)

        'Save the Workbook and Quit Excel

        oBook.SaveAs("C:\Hybrid Simulation Models\" & filename & "result.xls")
        oExcel.Quit()

        ReleaseStrobo()
    End Sub

    Private Sub TextBox2_TextChanged(sender As Object, e As EventArgs) Handles
TextBox2.TextChanged

    End Sub

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

    End Sub

    Private Sub Open EZStrobe_Click(sender As Object, e As EventArgs) Handles
Open EZStrobe.Click
        Dim MyProcess As New Process()
        MyProcess.StartInfo.FileName = "explorer.exe"
        MyProcess.StartInfo.Arguments = "C:\Program Files (x86)\Stroboscope
Simulation Systems\Stroboscope\Program\GUI\EZStrobe.vst"
        MyProcess.Start()
        MyProcess.WaitForExit()
        MyProcess.Close()
        MyProcess.Dispose()
    End Sub

    Private Sub Create Project Folder_Click(sender As Object, e As EventArgs)
Handles Create Project Folder.Click
        Dim di As DirectoryInfo = New DirectoryInfo("c:\Hybrid Simulation
Models")
        ' Determine whether the directory exists.
        If di.Exists Then
            ' Indicate that it already exists.
            MsgBox("That path exists already.")
        Else
            ' Try to create the directory.
            di.Create()
        End If
    End Sub

    Private Sub Open Project Folder_Click(sender As Object, e As EventArgs)

    End Sub


    Private Sub Open Vensim_Click(sender As Object, e As EventArgs) Handles
Open Vensim.Click
```

```
        Shell("C:\Program Files (x86)\Vensim\Vensim.exe")
    End Sub

    Private Sub Close EZStrobe_Click(sender As Object, e As EventArgs)

    End Sub

    Private Sub Exit System_Click(sender As Object, e As EventArgs) Handles
Exit System.Click
        Dim response As MsgBoxResult
        response = MsgBox("Do you want to close form?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
        If response = MsgBoxResult.Yes Then
            Me.Dispose()
        ElseIf response = MsgBoxResult.No Then
            Exit Sub
        End If
    End Sub

    Private Sub Open Project Folder_Click_1(sender As Object, e As EventArgs)
Handles Open Project Folder.Click
        Dim MyProcess As New Process()
        MyProcess.StartInfo.FileName = "explorer.exe"
        MyProcess.StartInfo.Arguments = "C:\Hybrid Simulation Models\"
        MyProcess.Start()
        MyProcess.WaitForExit()
        MyProcess.Close()
        MyProcess.Dispose()
    End Sub


    Private Sub LineShape1_Click(sender As Object, e As EventArgs)

    End Sub

    Private Sub Close EZStrobe_Click_1(sender As Object, e As EventArgs)
Handles Close EZStrobe.Click
        Dim _proceses As Process()
        _proceses = Process.GetProcessesByName("visio")
        For Each proces As Process In _proceses
            proces.Kill()
        Next


    End Sub

    Private Sub Close Vensim_Click(sender As Object, e As EventArgs) Handles
Close Vensim.Click
        Dim _proceses As Process()
        _proceses = Process.GetProcessesByName("vensim")
        For Each proces As Process In _proceses
            proces.Kill()
        Next
    End Sub
End Class
```