A COMPARISON OF THE FORESIGHT AND STROBOSCOPE METHODS OF MODELING CONSTRUCTION PROCESSES

By

VALEH NOWROUZIAN

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN CONSTRUCTION

UNIVERSITY OF FLORIDA

2012

To my parents, Farhad and Mahnaz, my husband, Babak and my brother, Roosbeh,
who supported me throughout my education

ACKNOWLEDGMENTS

My first thanks go out to my parents, who have supported me in my choice to attend the University of Florida for my graduate studies. Without their continued support and encouragement, I would not have been able to complete my education.

Also my thanks go out to Dr. Flood, Dr. Issa, and Dr. Lucas for serving on my committee and being patient with my need to complete the majority of my research from across the country, and their willingness and enthusiasm to helping me achieve this goal.

Lastly, my thanks go out to my classmates, who also served as colleagues and moral support throughout my education. They brought out the best in me, and helped me achieve things I never thought were possible for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CPM	Critical Path Method
DES	Discrete Event Simulation
LSM	Linear Scheduling Method

Abstract of Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science in Building Construction

A COMPARISON OF THE FORESIGHT AND STOBOSCOPE METHODS OF MODELING CONSTRUCTION PROCESSES

By

Valeh Nowrouzian

August 2012

Chair: Ian Flood
Major: Building Construction

Construction planning is one of the fundamental steps in the execution and management of construction projects. Its purpose is to optimize the value of construction projects by managing the selection of the technology, the identification of work tasks, the assignment of duration and cost to individual work tasks, determination of the required resources, and defining the interactions among the work tasks. Developing the construction plan is a challenging task. Basically, a planner determines the sequence of the work tasks for the specific project based on the drawings and specifications to optimize the project value. Selection of the right tool for planning the construction process is critical in assisting the project manager develop an optimal budget and schedule for all resources. Construction planners usually have to make compromises in selecting an appropriate planning tool due to the limitations of each method in terms of modeling versatility, simplicity and flexibility. An important approach to form the construction plan is simulation of the construction process using computer based discrete-event simulation techniques. This is the most versatile method currently available although it lacks the simplicity of planning tools such as the Critical Path Method (CPM) and linear scheduling.

This research evaluates a new modeling method suitable for project planning, Foresight, that has the objective of achieving the versatility of construction simulation whilst maintaining the simplicity in use of tools such as CPM and linear scheduling. Specifically this research will compare Foresight with Stroboscope which is the most sophisticated widely available construction simulation tool currently available.

CHAPTER 1
OPENING REMARKS

Motivation for Research

Construction planning is one of the important components of construction management that will specify the steps of the project process. The optimization of a project depends on how well the plan is developed. Using a well thought-out plan will benefit the entire industry. Therefore it is important which method of planning is being utilized to develop the model of planned work.

Time and cost are important factors of the construction processes. Time delays on construction processes are often accompanied by heavy cost impacts. Construction processes usually do not proceed exactly according to the developed plan as changes are inherent in construction processes. Construction plans should be adapted to these changes frequently. Moreover, Planning also ensures feasibility in terms of available resources, and determining an appropriate combination of resources and construction method.

The efficacy of a planning tool depends on how simple, accurate, versatile, and insightful it is (Flood 2010a). Complexity of the modeling method implies how easy a planner can learn the modeling language, develop a model of a specific construction process, modify that model, and understand and interpret the implications of that model.

Planning as a critical component of the construction process requires more research and development. A powerful method which has simplicity, versatility, and graphical insight can benefit the industry. Foresight (2010a) is a new method that has been developed in an attempt to overcome the limitations with the current planning methods. In a previous study (Zurich 2010) Foresight was compared to the most popular construction planning tool, CPM (the Critical Path Method). The next step is to compare Foresight with the most versatile construction planning tool, simulation, specifically Stroboscope (Martinez 1996) (the most powerful of the widely available construction simulation techniques).

Aim and Research Objectives

The primary goal of this research is to measure the relative performances of the two tools, Foresight and Stroboscope, and their relative value to project planners. The specific objectives of this research are:

- To determine the relative complexities of the two tools in terms of the number of concepts a user has to understand.
- To identify the number of concepts (measuring the level of expertise) and terms (measuring the amount of effort) that are required by the two tools to define individual models.
- To make a broad assessment of the relative versatility of the two tools (the ability to model different types of construction processes) and of the insight provided by their resultant models.

To identify the number of concepts and terms that are required to define individual models, three case studies are selected. These two modeling tools are compared in each case study in terms of:

- Ease of learning
- Ease of understanding
- Ease of model development
- Ease of modifying models

In terms of the scope, this research investigates the *complexity* issue of the two modeling tools in depth whereas the *insight* and *versatility* provided by each of these tools will be broad assessments.

In summary, the overall objective of this research is to determine the extent to which Foresight achieves the versatility of the simulation method, Stroboscope, while maintaining the simplicity of the Critical Path Method.

Literature Review

Modeling Construction Process Methods

The first step to build a construction project is to develop a plan. Construction planning is concerned with optimizing the project objectives and determining the project demands with respect to all resources, time, money, equipment, labor, space, and even information. Scheduling as a part of planning should outline the sequence of events to occur to complete the project by a particular date by assigning duration to each activity based on experience and/or historical information. Modeling a construction project is the basis of the project planning. In order to optimize the cost and time of the construction project, selection of the proper method for modeling construction processes is important. In order to select a proper method for planning a specific project the differences between the available methods of planning should be identified. Comparing the existing modeling tools and investigating whether they meet the requirements for various types of construction and how simple they are to use would help to select the appropriate tool for modeling construction processes.

A wide range of methods have been adopted, adapted and developed for modeling construction processes over the last 100 years. These tools can be grouped into three main categories: the Critical Path Method (CPM); the linear scheduling Method (LSM); and Discrete Event Simulation (DES) (Flood 2010b). Mostly other methods are either an integration or enhancement of these tools. For example, 4D-CAD and nD-CAD methods (Koo and Fischer 2000; Issa et al.2003), that include a dimension for time are strictly CPM models that for visualization purposes hybridized with 3D-CAD (Flood 2010b). A fourth method, Foresight, has been developed recently with the objective of combining the merits of each of these basic approaches (Flood 2010b).

Every project has its unique characteristic and most of the projects include different processes. The best approach for modeling some of these processes may be by using Critical Path Method techniques or for some other processes, linear scheduling or Simulation method of modeling might be the best selection (Flood 2009). In order to model a project, a planner can use more than one modeling method however, using several tools that are not fully compatible reduces the level of optimization of the process and compromises the user ability to plan and control work optimally (Flood 2009). The ideal way to model a construction process is using a single method which is highly versatile, able to model different situations such as repetitive and non-repetitive construction work, easy to use and able to provide an insight to alternative ways of planning (Flood 2009).

Critical Path Method

The Critical Path Method (CPM) is the most popular planning technique in the construction industry due to its simplicity of use and versatility (Hinze 2004). The purpose of the Critical Path Method is to determine when things can happen, and which activities need close management. In this method of scheduling, activities are defined and connected based on their relationship and sequencing in a network and each activity has duration. In this scheduling method, at the starting point of the project and in a logical sequence the particular activities needed to be performed for project accomplishment must be defined (Galloway 2006). Defined activities based on their assigned duration and relationships with other activities in the project provide the user with the critical path. Additional detail for reference or status purpose can be developed later. In order to optimize the project value, the user may change the critical path by modifying the durations, lags, and/or constraints throughout the project's progress (Galloway 2006). So in order to evaluate the project progress in the CPM-based schedule, the user should update the plan frequently (Galloway 2006). The initial set-up to develop a CPM model is challenging. This is why, it is necessary for the user to be knowledgeable enough about the construction process and its sequence in order to be able to develop a CPM-based model. However when it is modeled correctly at the outset, the analysis of CPM process throughout the project's progress would be significantly easier. The Critical Path Method is useful where the user wishes to see how changes would impact the project, but these are only successful if the CPM model is developed correctly with the appropriate relationships at the outset. Based on the initially developed CPM model, the user can determine potential alternatives to reduce the project's duration by allowing for the continuous evaluation of the project's progress (Galloway 2006).

Resource constraints in construction process often have a significant impact on the project schedule. CPM can consider the resource constraints but it has some limitations. CPM in its current form cannot handle many types of resources such as space and money very well. In addition, CPM is time centric and gives secondary concern to all other resources. For example, all dependencies are to do with start and finish times, they are not for example concerned directly with maintaining distances or restricting expenditures.

In CPM scheduling software, the user can assign resources to project activities and control them, but the schedule development can become overly-complicated since it is difficult to ascertain this resource assignment in bar chart format.

CPM scheduling software efficiency factor decreases when the schedule is assigned to multiple calendars (Kim et al. 2005).

Linear Scheduling Method

The Linear Scheduling Method (LSM) is a simple to use and very powerful tool for planning and control of linear construction processes. This method is not as popular as the Critical Path Method. The LSM provides a powerful and simple graphical display of resource allocation that is represented across a time-space interaction while also providing CPM-type calculations. The Linear Scheduling Method was designed to represent repetitive projects. Repetitive projects include repetitive activities which are associated with a consistent unit (high-rise buildings), or are based on a linear geometric layout (pipelines) (Kallantzis 2007).

Linear Scheduling is targeted at projects where there is repetition at a high level. This could be recognized in the steps user has to take to develop a model using this method. Basic steps required to develop a linear schedule include: 1) Activities identification; 2) Activity production rates estimation; and 3) Activity sequence development.

The reason that Linear scheduling is typically applied to repetitive projects is that in non-repetitive projects establishing the production rates are difficult. In linear schedules, the activity is defined by a line in a space that the horizontal axis represents the time and the vertical axis typically represents the distance. The slope of the line represents the production rate (Hinze 2004).

Linear scheduling represents the activities in a way which the horizontal distance or the gap between two activities shows the free float between them, the vertical gap between two activities shows the physical space between them. All activities that are happening at the same time could be identified by drawing a vertical line at the point in time along the horizontal axis. This is also the case in a bar chart schedule. However, in the linear schedules the user could determine the existence of free float at specific time between two activities by visualizing the line slope changes. So it is obvious that converging sloped lines may not allow for the later activity to begin sooner. This is one of the significant advantages of having the graphic representation of a linear schedule. Linear schedules are simple graphic schedules thus provide great visual insight into better ways of conducting a project to improve performance and reduce project duration. Also linear schedules could represent how adding additional crews or modifying the schedule to overtime will impact the schedule easily (Hinze 2004).

A linear Schedule is a static and deterministic method in its current form. The free float between two activities is the time buffer or horizontal distance between them. An interesting characteristic of linear scheduling in regard to free float is that free float between two activities may and usually change as one progresses through the vertical axis which usually represents the distance due to changes in production rates between activities.

A benefit to these schedules is that the user does not need to be concerned with early starts or late starts to the project; rather, LSM typically represents an expected schedule because of the precision associated with using production rates (Hinze 2004).

The important advantage to use linear schedules is that they are simple and quick yet provide highly visual representations of the schedule. The user of this method is able to view the schedule in a time-space continuum. It is significantly easy for user to determine methods of optimizing overall project or activity duration by comparing the effect of adding crews, working overtime, etc.

LSM also is not as thoroughly developed and supported in the industry as CPM is. The disadvantage of the linear schedules method is that it cannot be used to model non-repetitive works at all. It includes some simplistic assumptions which often make it difficult to model real-world repetitive processes (Flood 2010b). Even in projects that are linear, two activities can happen at the same point in time on that linear continuum; this cannot be effectively portrayed in a linear schedule (Yamín 2001). When there are multiple dimensions involved, linear scheduling effectiveness reduces. Another example in which the linear scheduling technique cannot be used easily to model the operations would be when there are different follow paths in operation, such as two underground utility lines that interact at a cross-over point but otherwise follow different routes (Flood 2010b).

Discrete Event Simulation

Construction processes are characterized by a very broad range of complexity. Analyzing complex processes through standard mathematical methods are difficult (Chen & Weng 2008). In simulation modeling, the user develops a model that imitates a real or imaginary dynamic system. The user predicts the behavior of a real or imaginary system in simulation based on the results of experiments. The assumption in DES is that the state of a system changes instantaneously at specific times marked by events. DES is very versatile and is able to model most construction processes (Martinez 1996).

Simulation-based tools have been developed to address the uncertain characteristics of construction. A major application to simulation-based tools is DES. Simulation-based tools could effectively solve operational construction problems (Sang Hyun Lee a, et.al.2006). It gives the planner more flexibility to model but the development process is difficult (Flood 2010b). DES is very versatile in the sense that it can model any type of interaction between tasks and any type of construction process including repetitive and non-repetitive work but it lacks the simplicity of CPM and provides no visual indication of how a system‘s logic determines it’s performance so DES has not been widely adopted in construction industry (Flood 2010b). DES could operate in dynamic modeling mode (Sawhney 1998). It considers stochastic duration, and incorporation of external factors like weather, labor productivity, and equipment breakdown. DES also allows the construction planner to perform a sensitivity analysis involving resource usage.

Stroboscope (STate and ResOurce Based Simulation of Construction ProcEsses) (Martinez 1996) is the most sophisticated simulation method for modeling construction processes which is freely available. Stroboscope is a refinement of CYCLONE (CYCLic Operations NEtwork). CYCLONE was specifically designed for construction by Halpin and Woodhead in (1976). CYCLONE is purely network based which limits the user to model processes at the level of detail required to make decisions. Some of these limitations are that CYCLONE is unable to identify the differences between similar resources, to identify the state of the simulated process and to dynamically use resource properties and the state of the simulation to define model behavior.

Martinez developed Stroboscope (1996) with the objective of enhancing the modeling capabilities of CYCLONE to consider a variety of resources and their specific characteristics, make the state of simulation to control the sequence of tasks and their relative priorities, model resource selection schemes similar to real construction operations, and model material utilization, consumption and production probabilistically.

A stroboscope model is presented by a network. A network consists of resources, links and nodes that are put together (Martinez 1996).

Resources. Resources in Stroboscope are items required to execute tasks. The most important characteristic of resource is the type of resource. The type of resource characterizes the resources based on their traits. Resources and resource types flow from one node to another node through the links. Resources in stroboscope consist of discrete and bulk resources. Resources that represent unique individual entities (e.g., Bulldozer) are discrete or non bulk resources. Those that do not represent individual entities that can be uniquely identified (e.g., Sand) are bulk resources. Bulk resource types in stroboscope are called “Generic Resources” and discrete resource types are called “Characterized Resources” (Martinez 1996).

Links. Links connect network nodes and indicate the direction and type of resources that flow through them. The predecessor is the node located at the tail of the link and the successor is the node located at the head of the link where arrow is located. Only resources with specified characteristic could flow through a link (Martinez 1996).

Nodes. Nodes are the elements of a network that hold the resources that are part of the system. There are two types of nodes available for resources to spend their time in: “Activities” and “Queues” (Martinez 1996).

Activities. Activities are nodes where resources actively spent time in them to perform the task associated with that activity. In Stroboscope there are three types of activities; The Normal Activities, Combi Activities and Consolidator Activities.

- Normal Activities are nodes where the tasks start just after the predecessor activities are finished.
- The Combi Activities are nodes where the activities start after some certain conditions are met. Predecessors to a combi must be Queues since Combis can be associated with the resources that are passive,
- Consolidators are activities that based on the resources they receive start and finish their instances.

Queues. Queues are nodes within which resources are stored. Resources passively spend time in Queue nodes. Each Queue is associated with a specific resource type. The resources have to wait in Queues until the conditions necessary to start the successor Combi are satisfied (Martinez 1996).

Forks and Dynaforks. Fork and Dynaforks are resource-type specific auxiliary network nodes. At the termination of an activity instance, Forks determine which of the successors to be activated and Dynaforks route resources. Auxiliary nodes are drawn smaller than the Normals, Combis and Queues nodes in a network drawing. They are called auxiliary because they do not have duration and resources never spend time in these nodes. Fork and Dynaforks are essentially link accessories. Forks are represented with a small circle that encloses a triangle in a network drawing. Dynaforks are represented with a small circle that encloses five rays in a network drawing.

Structure of stroboscope model files

Stroboscope model files are consisted of statements and comments.

Statement. The format of a statement is as follows:

STATEMENTKEYWORD [Arg1] [Arg2] […] [ArgN];

STATEMENTKEYWORD is the statement keyword, and Arg1, Arg2, …, ArgN are arguments for the statement. each statement must be ended by a semicolon. The statement keyword and its arguments, if any, are separated by white-space. The Stroboscope language is case sensitive. The statement keyword that gives the entire statement a name are always completely capitalized. The available statement keywords are defined by the Stroboscope language (Martinez 1996).

Arguments. The arguments consist of four types:

- User-defined identifier which is name that represent certain modeling elements
- Expressions which are composed of variables, constants, operators and function calls
- Strings which is required by some statements
- Multiple statements on the same line

Comments. Comments begin with a forward slash '/' and continue until end of the line.

Simulation model file processing

Simulation model will read and perform the statements in an input file, ignoring any comments. The execution of the statements will be done in the order in which the statements are inputted (Martinez 1996).

Stroboscope statements are classified in three groups:

- Element definition statements
- Element attribute statements
- Control statements

Element definition statements. The keyword for these statements is usually the generic name for elements of that category. For example, the COMBI statement is used to define Combis (Martinez 1996).

Element attribute statements. Attributes define how Stroboscope elements behave during a simulation. Combis, for example, have attributes such as Priority, Duration and Semaphore (Martinez 1996).

Control statements. These statements initialize the resources of a model or perform a simulation. They have an immediate effect upon execution of the model. For example, INIT is a statement for initializing resources and SIMULATE control statement is used to start running a simulation (Martinez 1996).

Foresight

Foresight is developed at the University of Florida as a new approach to modeling the construction processes. Foresight is an enhancement to the type and paradigm of Simulation modeling with the goal of achieving the simplicity of CPM, visual insight of linear scheduling and the power and versatility of simulation (Flood 2009). Foresight is developed with the objective of model having hierarchical structure and interactive development to improve the user's understanding of the organization and behavior of the system (Flood 2009).

Foresight is a static modeling method which considers stochastic duration. Using Foresight, the planner would be able to visualize the relationship between the structure of a model and the performance of the system. Thus, the planner can see consequences of changing certain parameters within the model immediately on the estimated performance of the system and the user could optimize the solutions (Flood 2010b). Work progress is therefore visible within the model's functional structure. Structure and interaction development as requisite attributes were identified hierarchically in order to facilitate the understanding of the behavior and organization of a system (Flood 2009).

Foresight modeling approach has three main concepts.

Attribute Space. This is the space in which the model represents process. Each axis of this space represents a different attribute involved in the performance of the process, such as time, cost and excavator. The attributes selected to make the space are the resources used to measure the performance. Their impact on the performance of the model is significant (Flood 2009).

Work Units. Individual items of work in the construction process which reside in attribute Space. The work unit could represent different levels of detail. An example for the low level of details is overall structural system. An example for the high level of details is forming and curing. The level of details considered by the user directly impacts on the complexity of the model. The work unit can have any shape and exist across multiple dimensions. Each work unit represents only one item of work and collectively they represent the total area of interest. Work units can be nested within each other. This offers some benefit to the user, for example, the model could be understood at different levels of abstraction, readability of the model will increase, the occurrence of errors in the model development as well as the attempt to define and update a model will decrease (Flood 2009).

Constraints and Objectives. Relationships between the work units and attribute space are defined by constraints. Constraints may be any functional relationship between work units and/or attribute space. The exact location of the work units are determined by the constraints. Some examples for constraints are as follows: to make sure that crews at different work units keep a safe working distance, to make sure the resource demands are satisfied by the available resources. The purpose of the planning is to optimize the value of the process, such as to maximize the profits or reduce the duration of the process. These objectives that are the goals and purpose of planning study are at a higher level of significance compared to constraints and therefore are considered in Foresight modeling significantly (Flood 2009).

One of the objectives in the development of Foresight is interactive development. That implemented in Foresight model by providing the user with visual presentation of the impacts on the model that are caused by any changes and modifications to constraints or work units. Another objective to develop Foresight is enabling the user to visualize directly how the performance of the model is dependent on its elements. This is implemented by allowing the user to visualize the work units plotted in the attribute space which presents the work progress in model's functional structure (Flood 2009).

CHAPTER 2
METHODOLOGY

The overall approach for comparison between Foresight and Stroboscope in this work is qualitative since Foresight is a new method not yet implemented in software. The best approach to compare these two methods and their versatility and complexity is by means of case studies in construction. Providing case study examples demonstrates their use and benefits to the user. The attempt is to model different problems (to compare versatility of methods) for different situations (sensitivity analysis) by Foresight and Stroboscope then compare the process and results. A comprehensive range of cases were selected to allow a broad comparison of the methods.

The first case study refers to variations of an excavation system at a construction site. The theme of the second case study is the concrete production and distribution system and the third study refers to a tunneling operation.

The method to compare the complexity of the two planning tools is through measuring the ease of use by comparing the amount of the information required by each approach to define a model, the amount of effort the user has to input to complete the operation, the visual insight provided by each model, and complexity of the resultant models.

The number of different modeling concepts that had to be employed and the number of terms that had to be defined to complete the model are two metrics selected for measuring the complexity and ease of use of a model. The number of different modeling concepts that had to be employed is the metric to measure the depth of understanding expertise that is required to develop the model. In this analysis each concept, no matter how many times it is employed within a model, is just considered once. The number of terms that had to be defined to complete the model is the metric to measure the amount of effort the developer has to input to complete the project. Ease-of-use of the Foresight and Stroboscope is compared for three different case studies based on modeling complexity.

The sensitivity study in a specific problem is performed in this work to compare the sensitivity of the Foresight and Stroboscope modeling tools to *complexity.* To perform the sensitivity analysis, the excavation system case study is considered for three different situations to compare the functions of these two tools at more complex situations. First model is developed for an excavator versus one truck. In the second situation an excavator is considered with two different types of trucks with variety of capacities and two of each type. In the third situation an excavator is considered with three different types of trucks with variety of capacities and three of each type.

CHAPTER 3
CASE STUDY # 1: EXCAVATION SYSTEM

Introduction

The first case study is to compare the complexity of the Foresight with Stroboscope models for variations of an excavation process which is highly repetitive in nature and would traditionally be best analyzed using simulation methods. This case study considers different levels of detail in this operation to test the complexity sensitivity of the two tools. Before providing an analysis and comparison of both the Stroboscope schedule and the Foresight approach to this project, a brief description of the project is required.

This project includes two cycles. First cycle is the Excavation Cycle which in this cycle an excavator resource performs its tasks digging, sluing, loading and sluing back. Second cycle is the Trucks Cycle which in this cycle truck resources perform their tasks loading, hauling, dumping and returning. Two cycles, *Excavation* and *Trucks*, have an overlap activity named *loading*. Loading the truck should be performed when both resources, excavator and truck, are available since the excavator loads the trucks. This system comprises of an excavator with 1 cu-m bucket capacity and a number of dump trucks of various capacities. There is a constraint that each truck resource must get fully loaded with soil before starting the *hauling* activity.

Comparison of the Foresight and Stroboscope is established for the variants of this excavation model. The first model comprises of one truck with 10 cu-m capacity. The second model comprises of two truck type, 10 cu-m and 15 cu-m capacities, and two of each. The third model comprises of three truck type, 10 cu-m, 15 cu-m and 20 cu-m capacities, and three of each.

Excavation System – Stroboscope

Figure 3-1 shows the Stroboscope modeling of a simple excavation system (see Martinez, (1996)). This figure represents the Stroboscope diagram which logically represents the process of an excavation operation. The details of the text input file written in Stroboscope's own language for this case study is presented in Appendix A,B and C Pages 69-86. This model uses the resources of type *Excavator, Truck, Soil* and *Space*. *Soil* and *Space* are bulk resource types. *Excavator* and *Truck* are discrete resource types. The network contains two Combi named *Load Scoop* and *Set Truck*; six Normals named *Dig, Sleu Back, Haul, Dump, Return* and *Slue*; four Queues named *Exc.Wait, Spot, Truck Wait* and *Moved Soil*; and one Consolidator named *Truck Full.*

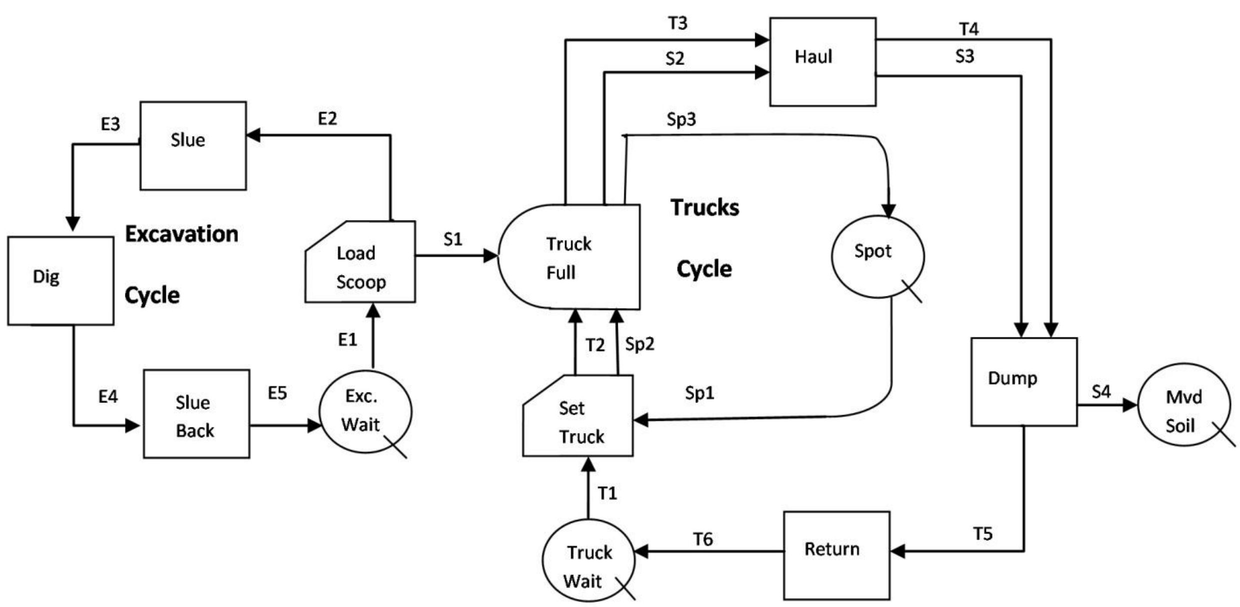


Figure 3-1. Stroboscope model of an excavation system, Stroboscope process diagram (See Martinez (1996))

Moved Soil holds resources of type *Soil*; *Truck Wait* holds resources of type *Truck*; *Exc.Wait* holds resources of type *Excavator;* and *Spot* holds resources of type *Space*. At the beginning of the simulation the *Excavator* initially in the system resides in *Exc.Wait. Truck* initially in the system resides in *Truck Wait. Space* initially in the system resides in *Spot*. The *soil* resources although are generated before the activity *Load Scoop* finishes. Each time the Excavator enters the Load scoop Combi node, before the activity finishes the amount of Soil equal to the Excavator's capacity is generated. Links E1, E2, E3, E4 and E5 represent that excavator is withdrawn from *Exc. Wait* to *Load* activity in order to load the truck with 1 cu-m soil and it is then withdrawn sequentially to the nodes *Slue* to slue where it digs, *Dig* to perform digging, *Slue back* to slue back to the place where it loads the trucks and finally it is withdrawn to *Exc.Wait*. Links T1, T 2, T 3, T 4, T5 and T 6 represent that trucks are initially waiting in *Truck Wait*. The first truck in line is withdrawn first to the *Set Truck* Combi node to get set for the loading. The truck then is withdrawn to the consolidator *Truck Full* where it gets loaded. Note that *Truck Full* is holding one truck at a time for loading. *Truck Full* keeps the truck until it gets fully loaded and amount of soil become equal to truck capacity. The loaded truck is then withdrawn sequentially to the nodes *Haul* to carry soil to dump area*, Dump* where the truck dumps its load*, Return* to move back to the place where it gets loaded, and *Truck Wait* to wait for the next loading when there is a space available. Links S1, S2, S3 and S4 represent that soil that is generated before the activity *Load Scoop* finishes is withdrawn to the consolidator, *Truck Full*. *Truck Full* holds the soil and truck until the amount of the soil become equal to the capacity of the truck in *Truck Full*. The soil then withdrawn sequentially to the nodes *Haul*, *Dump* and *Moved Soil.* Links Sp1, Sp2 and Sp3 represent that a space is initially waiting in the *Spot* Queue node to be drawn to the *Set Truck*. After *Set Truck* activity finishes, the space is withdrawn to the *Truck Full* consolidator and satisfies one of the constraints for consolidation. It then moves back to the *Spot* Queue node and becomes available for the next truck to get set for loading.

A Consolidator is used in the network to model the excavation operation. The excavator is not committed to completely fill a truck available in Consolidator in one step. The excavator with a 1 cu-m bucket is required to leave a truck with 10 cu-m capacity which is partially loaded nine times to slue back and dig and bring new loads of soil to fully load the truck.

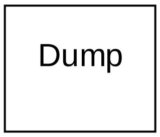
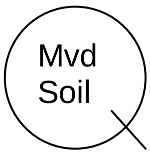
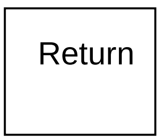
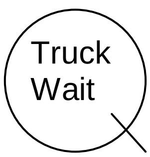
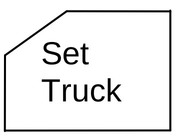
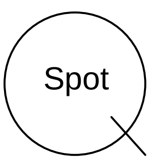
In the Stroboscope excavation network one *Space* Resource is used to constrain loading truck. The assumption is that one truck at a time can get loaded. This is implemented by initializing one unit of space in *Spot*. At the start point of *Setup Truck* activity, the space would be removed from the *Spot* and a truck from the *Trucks Wait.* When the *Set truck* activity finishes, the space and truck would be placed in *Truck Full*. After the truck gets fully loaded in the *Truck Full* and is going to start *hauling* activity, the *Space* resource moves back to *Spot* Queue node and it becomes available for the next truck to get set for loading. Consolidation in *Truck Full* depends on the availability of a truck, a space and an amount of soil equal to the capacity of the truck in the *Truck Full.*

Load Scoop represents the loading of a single scoop of soil into the truck. *Load Scoop* is a Combi node which is followed by *Exc. Wait* Queue. The occurrence of the *Load Scoop* is dependent to the availability of the excavator in the *Exc. Wait*. In addition to this default constraint, by using SEMAPHORE statement another constraint is added to the model which makes the occurrence of *Load Scoop* dependent to the availability of a truck in the *Truck Full*.

Table 3-1. Description of the Stroboscope process diagram components for the model of an excavation system

Component	Description
Slue	In this Normal Activity Node the Excavator slues to the place where it digs.
Load Scoop	In this Combi Activity Node the excavator loads a single scoop of soil into the truck placed in Truck Full Consolidator Node. The start of this activity is conditional to the availability of excavator in Exc. Wait Queue Node and a truck in Truck Full Consolidator Node.
Dig	In this Normal Activity Node the excavator digs.
Slue Back	In this Normal Activity Node the excavator slues back to where it loads the truck.
Exc. Wait	In this Queue Node the excavator waits until a truck become available in Truck Full Consolidator Node. When there is a truck in the Truck Full Consolidator Node, the excavator moves to activity, Load Scoop, to load a scoop of soil into the truck.
Truck Full	This Consolidator Node holds the truck and soil which enter into it until the amount of soil become equal to the truck capacity. When the amount of soil in the Truck Full becomes equal to the truck capacity, it releases the truck and soil.
Haul	In this Normal Node the truck and soil which are released from the Truck Full Consolidator Node perform Haul activity representing that the truck hauls the soil.

Table 3-1. Continued

Component	Description
	In this Normal Node the truck and soil which finished the Haul activity perform Dump activity representing that the truck dumps the soil.
	After the dump activity is performed, the soil moves into this Queue Node, Mvd Soil, and it becomes part of the moved soil.
	After the dump activity is performed, the Truck moves into this Normal Node, Return, and it returns to where it loads.
	In the Truck wait Queue Node, the truck which returns from dumping the soil waits in line to get set for the next load.
	When there is a space available, the first truck in line in the Truck Wait Queue Node enters into the Set Truck Combi Node to get set for loading.
	To satisfy the constraint that one truck at a time can get loaded, one space resource is initialized in Spot Queue Node. When the space is located in Spot Queue Node, it is available. Thus the space and the first truck in line in the Truck Wait Queue Node can enter into the Set Truck Combi Node. After the truck gets loaded in Truck Full Consolidator and is going to start hauling the soil, the space moves back to the Spot Queue Node and becomes available for the next truck in line in the Truck Wait Queue Node.

As *Load Scoop activity* finishes an amount of soil equal to the excavator capacity generates and is withdrawn to *Truck Full.* The *Truck Full* consolidator evaluates its ConsolidateWhen attribute to determine if it should consolidate. Consolidation happens when the amount of Soil in the *Truck Full* node is equal or greater than the capacity of the truck in Consolidator *Truck Full*.

Figure 3-2 represents the time-wise results of the model measured at the dump activity for a situation where there are 2 dump trucks of 10 cu-m capacity each and 2 dump trucks of 15 cu-m capacity each. Table 3-1 describes each node and its function for the excavation case study.

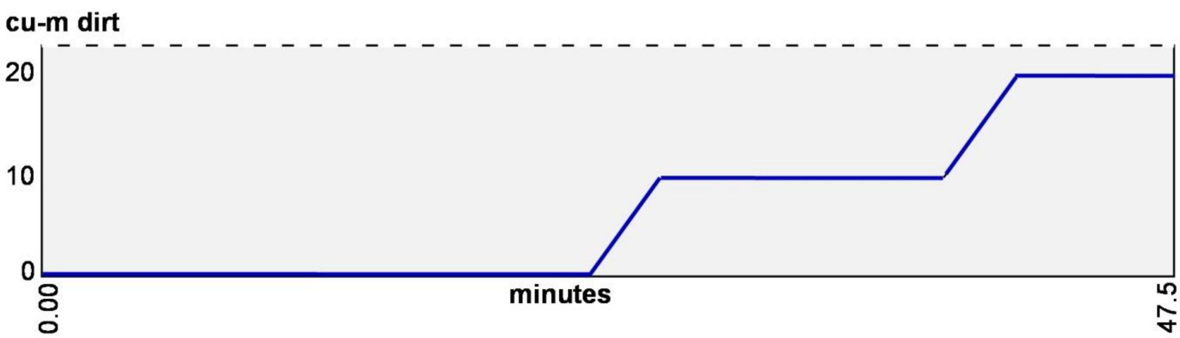


Figure 3-2. Stroboscope model of an excavation system, typical simulation output (moved soil; 47.5 minutes of production)

Excavation System - Foresight

The newly proposed hierarchical constraint-based approach to modeling, Foresight, provides a more visual insight to the excavation process. Figure 3 represents the Foresight model for the equivalent model of the same excavation system.

Figure 3-3, part A represents the breakdown of the project into the work units and hierarchical structure of the model. In this case study, there are two resources defining the dimensions of the attribute space: the first, along the horizontal axis, is time in minutes; the second, along the vertical axis, is cubic meter of soil. Figure 3-3, part (b) represents both the model structure and the performance of the system over time with constraints. At the highest breakdown of the operation, two work units named *Exc.*

Cycle and *Truck Cycle* are defined. Work unit, *Truck Cycle,* breaks down into the second level of work units named *Load Truck, Haul, Dump* and *Return*. *Exc. Cycle* work unit breaks down to the next level of work units named *slue, dig, slue and load*. The capacity of the excavator is 1 cu-m. Each of the two trucks has 10 cu-m capacity. In order to fully load a truck, the *Exc. Cycle* work unit is repeating ten times inside each of the *Load Truck* work unit. In this case study, the second level work units *Truck cycle* and *Excavation cycle* do not follow an exact hierarchy although the model in form is strongly hierarchical. It means, they are not exclusive from each other and neither nested within one or other, but rather they overlap each other. The overlap shows the section of the work where both the excavator and trucks perform together and the excavator loads the truck.

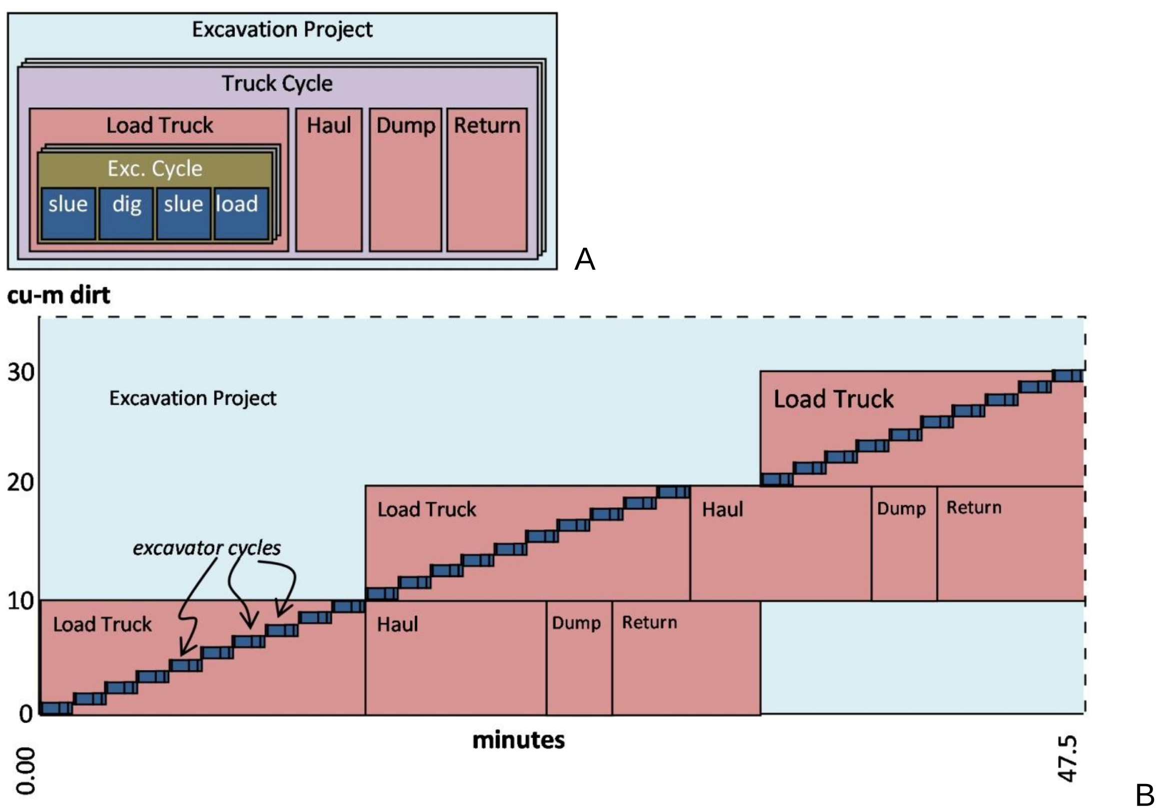


Figure 3-3. Foresight model of an excavation system. A) hierarchical model structure. and B) constrained model (2 dump trucks; first 47.5 minutes of production) (Flood 2012)

Figure 3- 3, part B represents the model as it exists in the time and cu-m of soil excavated dimensions, where all constraints for the first two truck loads are resolved. The process of the model would continue until some overall constraint such as a limit on the amount of soil to be excavated was reached.

In order to compare the Foresight and Stroboscope, variety of resources for this excavation model was used as follows:

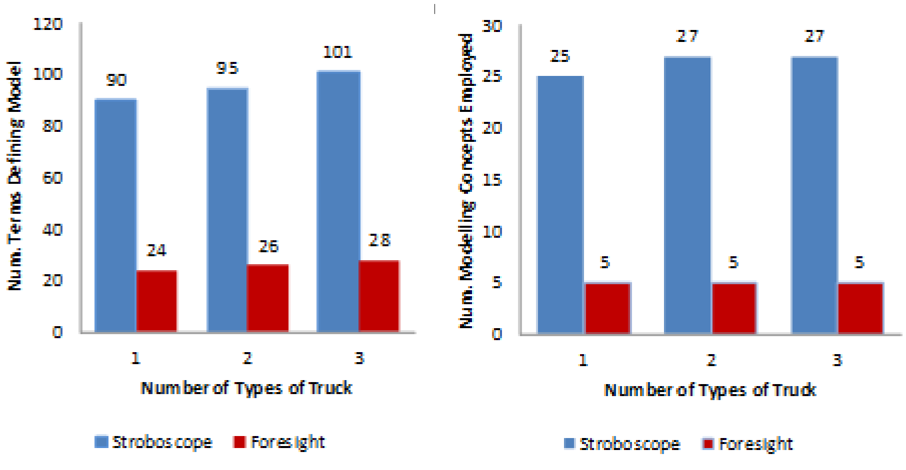
- 1 Truck (10 cu-m capacity)
- 2 Trucks (10 cu-m capacity) + 2 Trucks (15 cu-m capacity)
- 3 Trucks (10 cu-m capacity) + 3 Trucks (15 cu-m capacity) + 3 Trucks (20 cu-m capacity)

Beside the variety of resources mentioned above, all other parameters of the model between the model variants were kept constant (ex. Activity durations for the different truck capacities).

Comparison Results

The complexity of the Foresight and Stroboscope modeling development for each of the variants of the three excavation models are compared in terms of the number of concepts and terms. As mentioned in chapter 2, the number of concepts is a measure of the depth of understanding expertise that is required to develop the model and the number of terms is a measure of the amount of effort required to develop the model by the user and is considered to be any definition or item that is essential to define the structure and performance of the model. Figure 3-4, part A represents the number of terms required for both Stroboscope and Foresight to define the excavation model for each of the three variants. Example terms in Stroboscope are definitions of the activities and queue nodes, links and activities durations, the definition f the resources like excavator and trucks and their characteristics if any for example their capacities and their numbers, and the definitions of the simulation time. Example terms in Foresight are the attributes such as time and soil, the work units and their constraints, and the repetition of work units (note, the amount of work to be modeled is implicitly defined by the constraints on the highest level work unit). Figure 3-4, part A shows that the amount of information required by Foresight to define these models is about 26% of that using Stroboscope. Based on the comparison it can be conclude that Foresight and Stroboscope models are identical in terms of the process logic represented.

Figure 3-4, part B makes a similar comparison but in terms of the number of concepts required to define a model for both Stroboscope and Foresight for each of the three variants of excavation model. In this analysis each concept, no matter how many times it is employed within a model, is just considered once. Example concepts in Stroboscope are definitions of types of nodes including consolidator, definition of link and its attributes and duration for activities, the definition f the resources types like bulk and characteristic, and the definitions of the simulation time. To model the process in Foresight, however, only five concepts required to be defined: (i) the types of attributes; (ii) the work units; (iii) the constraints defining the relative locations of the various boundaries of the work units; (iv) nesting of work units; (v) repetition of work units. It could be seen in figure 3-4, part B that the number of concepts employed by Foresight is about 19% to 20% of that employed by Stroboscope. It could be conclude that a Foresight model-user must learn how to use the 5 base concepts to show each logical construct in a system, such as making sure that the excavator accomplishes the correct number of cycles to fully load a truck. However, a Stroboscope model-user to achieve each logical construct must also learn how to figure out the various Stroboscope modeling components.



A B

Figure 3-4. Complexity of variants of the excavation model for Foresight vs. Stroboscope in terms of number of A) terms and B) concepts

CHAPTER 4
CASE STUDY # 2: CONCRETE PRODUCTION AND DISTRIBUTION SYSTEM

Introduction

A second case study was developed to compare the complexities of Foresight and Stroboscope models for production and distribution of wet concrete via a hopper. The intent was to consider very different types of construction system to see if the relative advantages are maintained. Before providing an analysis and comparison of both the Stroboscope and Foresight models for this project, a brief description of the operation is required.

The system comprises of 1 mixer, 1 hopper and 2 trucks. The mixer produces wet-concrete in a 1 cu-m batches. The hopper has 5 cu-m capacity which is used for temporary storage of the wet-concrete. Each truck has 3 cu-m capacity. Trucks are used to distribute the wet-concrete. Mixer performs tasks loading, mixing and pouring. Hopper receives concrete from the mixer. Mixer and hopper have an overlap activity (pour concrete to hopper) which requires both resources to be accomplished. Trucks receive concrete from the hopper. One of the constraints for this model is that when the hopper gets full, the mixer should stop loading. It can start loading again when the hopper loads a truck and some space become available. Trucks perform tasks *receiving concrete from hopper*, *traveling, pouring* and *returning*. Hopper and trucks have an overlap activity (*Receive concrete from hopper*) which requires both resources to be accomplished. The problem is designed for situation where only one truck could get loaded at a time by the hopper.

Concrete Production and Distribution System Case Study - Stroboscope

Figure 4-1 shows the Stroboscope model for concrete production and distribution system case study. This figure represents the Stroboscope diagram which logically represents the process of concrete production and distribution operation. The details of the text input file written in Stroboscope's own language for this case study is presented in Appendix D, Pages 86-94. This model uses the resources of type *Mixer, Space, Spacey, Concrete, Truck and Permit. Concrete, Space, Spacey* and *Permit* are bulk resource types. *Mixer* and *Truck* are discrete resource types. The network contains Three Combi named *Load, Pour Concrete to Hopper* and *Receive Concrete*; four Normals named Mix*, Travel, Pour Concrete* and *Return*; eleven Queues named *Mixer to Load, Mixer to Pour, Spacey I, Spacey II, Spacey III, Spot I, Spot II, Concrete Wait, Trucks Wait, Allow* and *Mvd Concrete*;

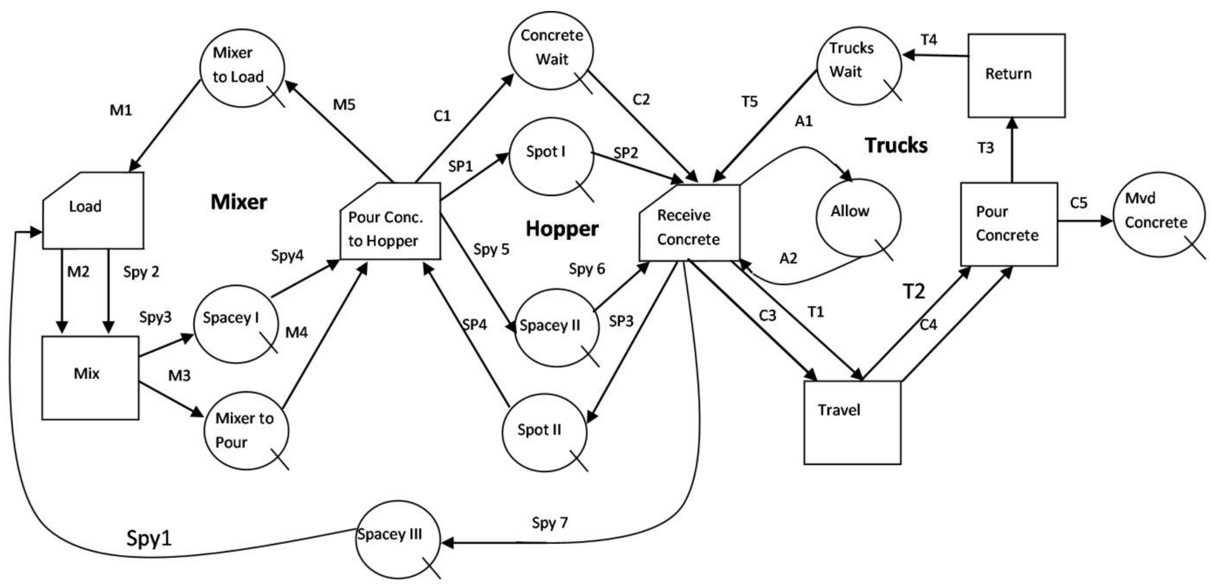


Figure 4-1. Stroboscope model of a concrete production and distribution system

Mvd Concrete and *Concrete Wait* hold resources of type *Concrete*; *Spacey I, Spacey II* and *Spacey III* hold resources of type *Spacey*; *Spot I* and *Spot II* hold resources of type *Space, Trucks Wait* holds resources of type *Truck; Allow* holds resources of type *Permit* and *Mixer to Load* and *Mixer to Pour* hold resources of type *Mixer*. At the beginning of the simulation the *Mixer* resource initially in the system resides in the Queue node, *Mixer to Load; Truck* resources initially in the system reside in the Queue node, *Trucks Wait; Spacey* resources initially in the system reside in the Queue node, *Spacey III; Space* resources initially in the system reside in the Queue node, *Spot II* and *Permit* resource initially in the system resides in the Queue node, *Allow.* However, the *Concrete* resources are generated before the activity, *Pour Concrete to Hopper,* finishes. Each time the Mixer and Spacey enters the *Pour Concrete to Hopper* Combi node, before the activity finishes the amount of concrete equal to the mixer capacity is generated. Links M1, M2, M3, M4 and M5 represent that Mixer are withdrawn from *Mixer to Load* to activity *Load* to get loaded and then withdrawn sequentially to the nodes *Mix* to operate mixing activity, to *Mixer to Pour* to wait for loading the Hopper when there is space*,* to *Pour Concrete to Hopper* to pour its load in to the hopper and to *Mixer to Load* waiting for the next load when there is a space available in the hopper. Links Spy1, Spy2, Spy3, Spy4, Spy5, Spy6 and Spy7 represent that Spacey, representing the availability of the space in the Hopper as a criteria for loading the Mixer, is withdrawn (whenever hopper pours concrete to a truck and a space become available in hopper) from *Spacey III* sequentially to the nodes, *Load the Mixer, Mix*, *Spacey I* waiting to be withdrawn, *Pour Concrete to Hopper*, *Spacey II*, *Receive Concrete* when there is a Truck available and again to *Spacey III* to satisfy one of the constraints of *Load* activity. Links C1, C 2, C 3, C 4 and C5 represent that Concrete is generated before the activity *Pour Concrete to Hopper* finishes. It is then waits in the *Concrete Wait* to be withdrawn to the *Receive Concrete* when a Truck is available in the *Trucks Wait*. It then would be withdrawn sequentially to the nodes *Travel* to be hauled, to *Pour Concrete* to be poured and to *Mvd Concrete* to become part of the placed concrete. Links Sp1, Sp 2, Sp 3 and Sp 4 represent that Space, representing the availability of a space in the Hopper as a criteria for receiving concrete from the mixer, are withdrawn from *Spot II* sequentially to the nodes, *Pour Concrete to Hopper* to show that a space is full, *Spot I* to wait to be withdrawn when there is a truck, *Receive Concrete* showing the space is available and again to *Spot II* to satisfy one of the constraints to the *Pour Concrete to Hopper* activity. Links T1, T 2, T 3 and T 4 represent that the truck is withdrawn from *Trucks Wait* sequentially to the nodes, *Receive Concrete* to get loaded with concrete, *Travel* to carry the concrete, *Pour Concrete* to pour the Concrete, *Return* to go back where it gets loaded, *Trucks Wait* to wait in line for the next loading and *Receive Concrete* to get loaded. Links A1and A2 represent that Permit is withdrawn from *Allow* to *Receive Concrete* constraining the loading of the trucks to one truck at a time (First In First Out).

Since the trucks have 3 cu-m capacities each, the activity, *Receive Concrete,* needs to receive concrete resources 3 times -each time 1 cu-m- to fully load the truck. For this reason, the availability of 3 cu-m concrete resources in the *Concrete Wait* Queue node is defined to be a criteria that should have been met in order for the activity, *Receive Concrete*, to start. This has been defined by ENOUGH statement. In a situation when the *Concrete Wait* has more than 3 cu-m concrete resources stored in it, the links Sp2, Spy6 and C2 are defined to draw only 3 resources of type Space, Spacey and Concrete to *Receive Concrete* activity by using DRAWUNTIL statement. Table 4-1 describes each node and its function for the excavation case study.

Table 4-1. Descriptions of the Stroboscope process diagram components for the model of a concrete production and distribution system

Component	Description
Mixer to Load	At the beginning of the simulation, in the Mixer to Load Queue Node a mixer is initialized. During the simulation, the mixer waits in this Queue Node until there is a space available in hopper so it can load the aggregates and materials.
Load	In this Combi Activity Node the mixer loads materials. The start of this activity is conditional to the availability of mixer in the Mixer to Load Queue Node and a spacey in Spacey III Queue Node.
Mix	In this Normal Activity Node the Mixer mixes the materials.
Mixer to Pour	The Mixer waits in the Mixer to Pour Queue Node until there is a space available in the hopper so it can pour the concrete in the hopper.
Spacey I	The Spacey waits in the Spacey I Queue Node until the mixer can pour the concrete in the hopper.
Pour Conc. to Hopper	In the Pour Conc. to Hopper Combi Node, the Mixer pours the concrete to the hopper. Before this activity finishes, the amount of concrete equal to mixer batch is generated. The start of this activity is conditional to the availability of a mixer and a Space in the hopper.

Table 4-1. Continued

Component	Description
Concrete Wait	The concrete that is generated before the Pour Conc. to Hopper activity finishes enters to the Concrete Wait Queue Node. The concrete stays in the Concrete Wait Queue Node until the amount of concrete in the Concrete Wait Queue Node becomes equal to the truck capacity. When the amount of concrete in the Concrete Wait Queue Node becomes equal to the truck capacity, the Concrete Wait Queue Node satisfies one of the constraints to the Receive Concrete Combi Node.
Receive Concrete	The Receive Concrete Combi Node is where the trucks receive concrete from the hopper. The start of this activity is conditional to the availability of a truck in the Trucks Wait Queue Node and the amount of concrete equal to the truck capacity in the Concrete Wait Queue Node.
Travel	In this Normal Activity Node the truck that is loaded with concrete travels to the place where it places the concrete.
Pour Concrete	In this Normal Activity Node the loaded trucks place the concrete.
Mvd Concrete	After the Pour Concrete Activity Node finishes, the concrete enters the Mvd Concrete Queue Node and stays there.
Return	After the dump activity is performed, the Truck flows into this Normal Node, Return, and it returns to where it receives the concrete load.
Truck Wait	At the beginning of the simulation, in the Truck Wait Queue Node numbers of trucks are initialized. During the simulation, in the Truck wait Queue Node, the truck which returns from placing the concrete waits in line for the next concrete loading.

Table 4-1. Continued

Component	Description
Spacey III	At the beginning of the simulation, five Spacey resources representing the availability of space in the hopper are initialized in the Spacey III Queue Node. During the simulation, the Spacey resources wait in the Spacey III Queue Node and satisfy one of the constraints for loading the mixer.
Spacey II	The Spacey resources wait In the Spacey II Queue Node showing that the space in hopper is full. Each time the Receive Concrete Combi Node finishes three spaceys move to spacey III Queue node and become available and satisfy one of the constraints for loading the mixer.
Spot I	At the beginning of the simulation, five Space resources representing the availability of space in the hopper are initialized in the Spot I Queue Node. During the simulation, the Space resources wait in the Spot I Queue Node and satisfy one of the constraints for pouring the concrete to the hopper.
Spot II	The Spacey resources wait In the Spot II Queue Node showing that the space in hopper is full. Each time the Receive Concrete Combi Node finishes three spaces move to spot I and they become available and satisfy one of the constraints for pouring the concrete to the hopper.
Allow	The assumption is that one truck at a time can receive concrete from the hopper. To satisfy this assumption, at the beginning of the simulation one permit is initialized in Allow Queue Node. When the permit is located in Allow Queue Node, it is available. Thus the permit and the first truck in line in the Trucks Wait Queue Node can enter into the Receive Concrete Combi Node. After the truck gets loaded, the permit moves back to the Allow Queue Node and becomes available for the next truck in line in the Trucks Wait Queue Node.

Concrete Production and Distribution System Case Study - Foresight

Figure 4-2 represents the Foresight equivalent model of the above concrete production and distribution system for performing the total of six concrete placing (three by each truck). In this case study, there are two resources defining the dimensions of the resource space: the first, along the horizontal axis, is time in minutes; the second, along the vertical axis, is cubic meter of concrete.

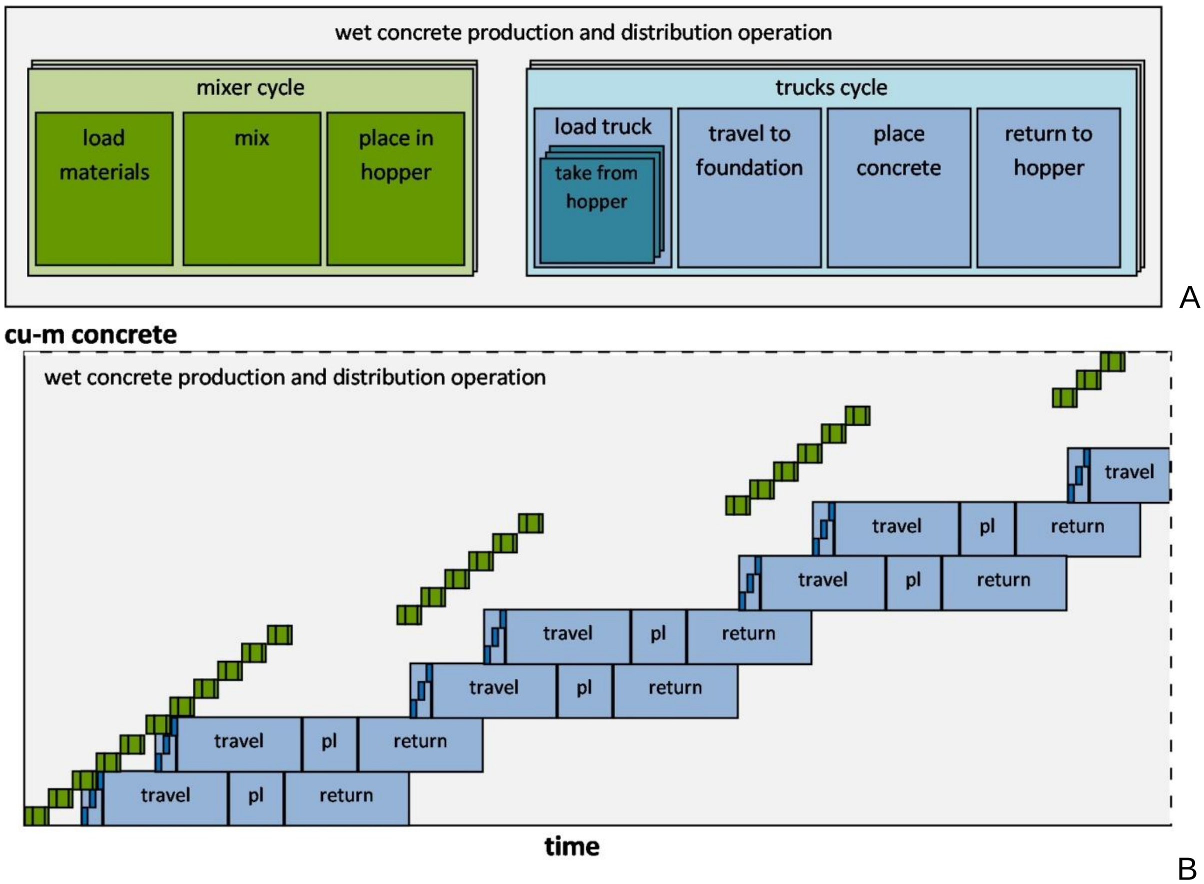


Figure 4-2. Foresight model of a concrete production and distribution system. A) hierarchical model structure and B) constrained model (first 6 distribution truck cycles)

Figure 4-2, part A represents the hierarchical structure of the model while Figure 4-2, part B represents both the model structure and the performance of the system over time with constraints. Figure 4-2, part A represents the breakdown of the project into the work units. At the highest breakdown of the operation, two work units named *Mixer Cycle* and *Truck Cycle* are defined. Work unit, *Truck Cycle,* breaks down into the second level work units named *Load Truck, Travel to Foundation, Place Concrete* and *return to hopper*. *Load truck* work unit breaks down to the next level to the work unit *take from hopper*. Since the capacity of truck is 3 cubic meter *take from hopper* work unit is repeating tree times in each *load truck* work unit. The work unit, *Mixer Cycle,* breaks down into the next level to work units named *load materials, mix* and *place in hopper.* Figure 4-2, part B indicates that when hopper is full (when *mixer cycle* has been performed 5 times back to back without *load truck* being performed due to unavailability of truck) , the mixer stops loading until the hopper loads a truck and space become available for more concrete.

Comparison Results

The complexity of the Foresight and Stroboscope modeling development for concrete production and distribution system is compared in terms of the number of concepts and terms. As mentioned in chapter 2, the number of concepts is a measure of the depth of understanding expertise that is required to develop the model and the number of terms is a measure of amount of effort required to develop the model by the user and is considered to be any definition or item that is essential to define the structure and performance of the model.

Figure 4-3, part A represents the number of terms required for both Stroboscope and Foresight to define the concrete production and distribution system. It shows that the amount of information required by the Foresight was just 20% of that using Stroboscope. Based on the comparison it can be conclude that the Foresight and Stroboscope models are identical in terms of the process logic represented.

Figure 4-3, part B represents the number of concepts required for both Stroboscope and Foresight to define the concrete production and distribution system. It shows that the number of concepts employed by Foresight is about 19% of that employed by Stroboscope. Based on the comparison it can be conclude that Foresight and Stroboscope models are identical in terms of the process logic represented.

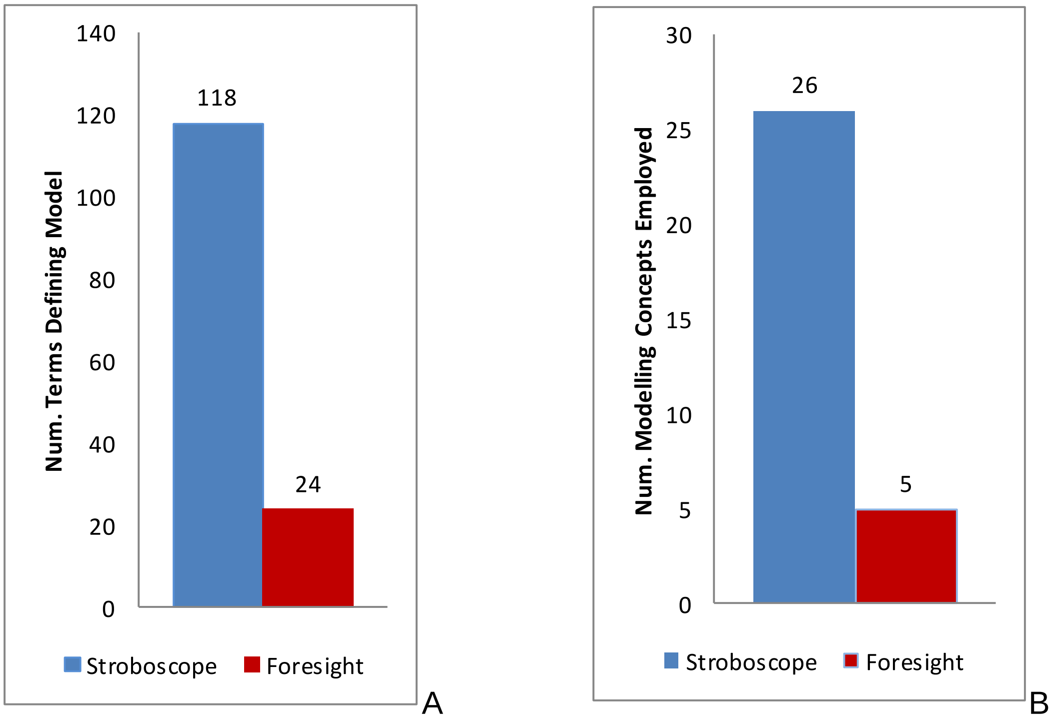


Figure 4-3. Complexity of the concrete production and distribution system for Foresight vs. Stroboscope in terms of number of A) terms and B) concepts

CHAPTER 5
CASE STUDY # 3: SEWER TUNNELING SYSYTEM

Introduction

A third case study to compare the complexities of the Foresight and Stroboscope models was developed for a more elaborate system, that is concerned with constructing a sewer tunnel with 2 meter internal diameter. Before providing an analysis and comparison of both the Stroboscope and Foresight models of this project, a brief description of the operation is provided.

In this case study, the operation of the tunneling is assumed to be through clay and the lining to be formed from the concrete ring segments. The system comprises of three cycle; the excavation, concrete lining and light track. The system consists of two tunneling crews that start in the midpoint of the tunnel and operate their tasks a head in opposite directions. The excavation of clay is performed by crews with a pneumatic spade. For each 1 m length of the tunnel, three skip loads of excavated material are required. After excavation of a 1 m length of the tunnel, the crew starts bringing in a set of concrete ring segments to lining that section of the tunnel. After the excavation and lining of a 3 m of the tunnel, the crew lay a new section of light track. In excavation cycle the resources perform dig, load, push, load, haul and empty. In concrete lining cycle the resource moves through the activities load, push, place rings and return. the Light track cycle include activities load, push, position and return.

Tunneling System Case Study – Stroboscope

Figure 5-1 shows the Stroboscope tunneling model for 1 crew for tunneling system. This figure represents the Stroboscope diagram which logically represents the process of the tunneling operation. The details of the text input file written in

Stroboscope's own language for this case study is presented in Appendix E, Pages 94-105.

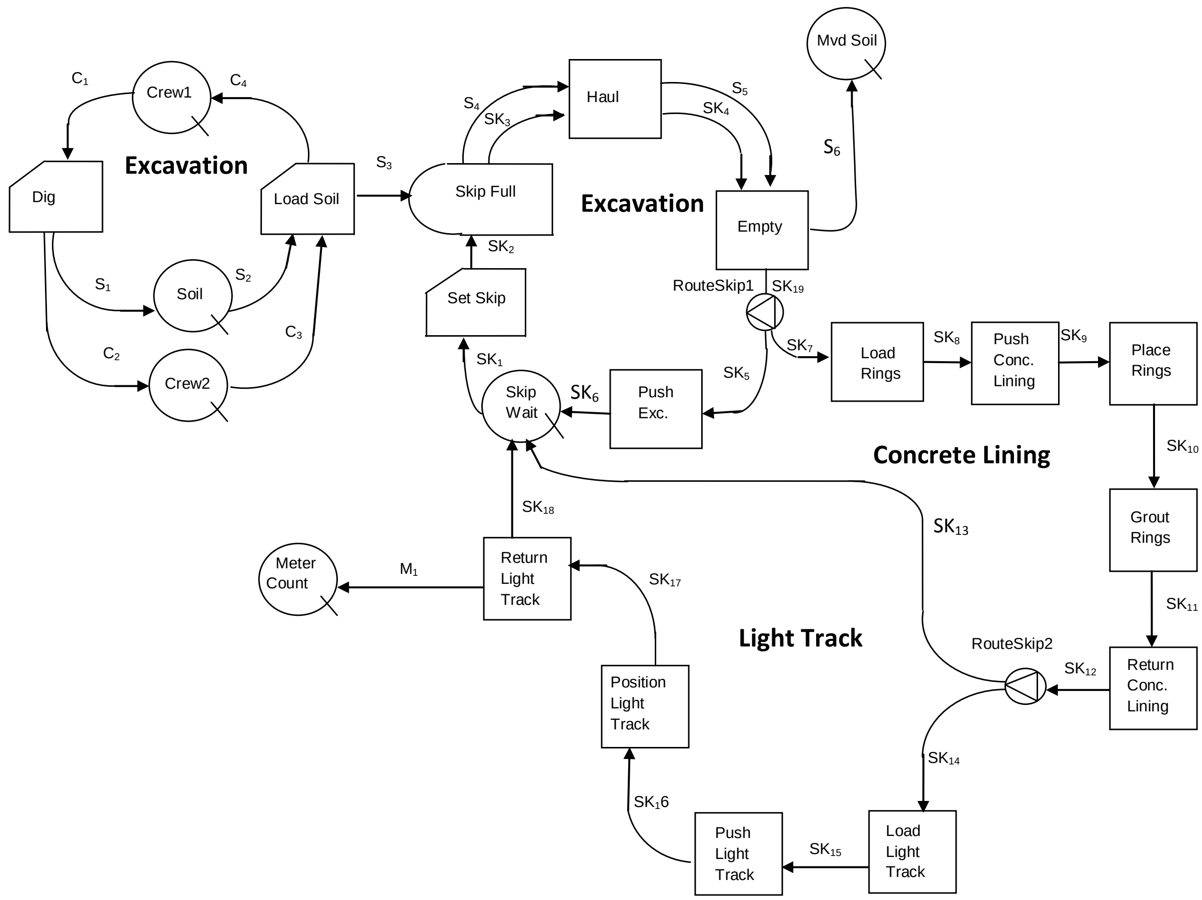


Figure 5-1. Stroboscope model of a tunneling system for 1 crew

This model uses the resources of type *Crew, Soil, Skip,* and *Meter*. *Soil, Crew* and *Meter* are bulk resource types. *Skip* is discrete resource type. The network contains Three Combi named *Dig, Load Soli* and *Set Skip*; twelve Normals named *Haul, Empty, Push Exc., Load Rings, Push Conc. Lining, Place Rings, Grout Rings, Return Conc. Lining, Load Light Track, Push Light Track, Position Light Track and Return Light Track*; six Queues named *Crew1, Crew 2, Soil, Skip Wait, Mvd Soil, and Meter Count*; one Consolidator named *Skip Full;* and two auxiliary Fork nodes named *RouteSkip1* and *RouteSkip2*.

Table 5-1. Descriptions of the Stroboscope process diagram components for the model of a tunneling system

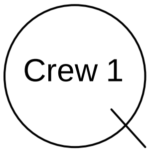
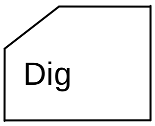
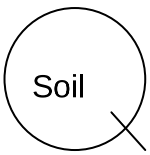
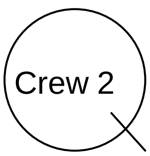
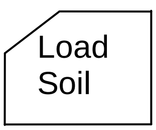
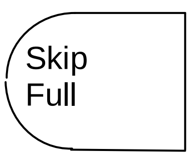
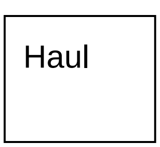
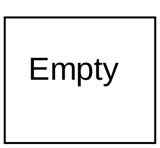
Component	Description
	At the beginning of the simulation, in the Crew 1 Queue Node a crew is initialized.
	In this Combi Activity Node the crew digs. Before this activity finishes, the amount of soil is generated.
	The soil that is generated before the Dig activity finishes enters to the Soil Queue Node.
	The crew waits in the Crew 2 Queue Node.
	In this Combi Activity Node the crew loads the skip with soil. The start of this activity is conditional to the availability of a crew in the crew 2 Queue Node, soil in the Soil Queue Node and a skip in the Skip Full Consolidator Node.
	This Consolidator Node holds the skip and soil which enter into it until the amount of soil becomes equal to the skip capacity. When the amount of soil in skip Full becomes equal to the skip capacity, it releases the skip and soil.
	In this Normal Node the skip and soil which are released from the Skip Full Consolidator Node perform Haul activity representing that the truck hauls the soil.
	In this Normal Node the skip and soil which finished the Haul activity perform Dump activity representing that the skip empties the soil.

Table 5-1. Continued

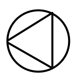
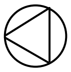
Component	Description
Mvd Soil	After the empty activity is performed, the soil moves into this Queue Node, Mvd Soil, and it becomes part of the moved soil.
Push Exc.	After the empty activity is performed and when the one meter length of the tunnel is not excavated yet, the skip flows back to the excavation cycle. The resource performs the Push Exc. Normal Activity Node and it returns to where it loads the soil.
RouteSkip1 	This accessory node of the link decides whether to send the resource, whether back to the excavation cycle or to the concrete lining cycle. If the one meter length of the tunnel has been excavated the resource would be routed to the concrete lining cycle otherwise, it would be routed to the excavation cycle.
Load Rings	After the empty activity is performed and when the one meter length of the tunnel is excavated, the skip moves into the concrete lining cycle. The resource performs the Load Rings Normal Activity Node . In this node skip gets loaded with the concrete ring segments.
Push Conc. Rings	In this activity, the skip that is loaded with the concrete ring segments is pushed to the place where the ring segments should be placed.
Grout Rings	In this activity, the grout would be applied to the rings.
Return Conc. Lining	In this activity, the resource returns from concrete lining cycle.

Table 5-1. Continued

Component	Description
RouteSkip2 	This node is an accessory of the link which decides whether to send the resource, whether back to the excavation cycle or to the light track cycle. If the three meter length of the tunnel has been excavated and lined, the resource would be routed to the light track cycle, otherwise it would be routed back to the excavation cycle.
Load Light Track	After the return conc. lining activity is performed and when the three meter length of the tunnel is excavated, the skip moves into the Light Track cycle. The resource performs the Load Light Track Normal Activity Node . In this node resource gets loaded.
Push Light Track	In this activity, the resource is pushed.
Position Light Track	In this activity, the resource is positioned.
Return Light Track	In this activity, the resource returns to the excavator cycle. Before this activity finishes three meter resources, representing the accomplishment of the three meter length of the tunnel, is generated.
Meter Count	The meter that is generated before the Return Light Track activity finishes enters to the Meter Count Queue Node. The content of this Queue node shows the length of the tunnel that is completed.

Mvd Soil and *Soil* hold resources of type *Soil*; *Crew 1* and *Crew 2* hold resources of type *Crew*, *Skip Wait* holds resources of type *Skip, and Meter Count* holds resources of type *Meter*. At the beginning of the simulation the *Crew* resources initially in the system reside in Queue, *Crew1, and Skip* resources initially in the system reside in Queue, *SkipWait,* although the *Soil* resource is generated before the activity *Dig* finishes *and Meter* resources are generated before the activity *Return Light Track* finishes. Each time the Crew enters the *Dig* Combi node, before the activity finishes the amount of Soil equal to one third of skip's capacity is generated and each time the skip enters the *Return Light Track* Combi node, before the activity finishes three Meter resource is generated. Links C1, C2, C3 and C4 represent that crew is withdrawn from *Crew 1* to the activity *Dig* to perform digging and then it is withdrawn sequentially to the nodes *Crew 2* to wait until the skip is available, *Load* to load the skip with soil and *Crew* 1Queue node for the next loading. Links S1, S2, S3 and S4 represent that soil is generated before the activity *Dig* finishes. It is then withdrawn sequentially to the nodes *Skip Full* to stays there until the amount of soil in the *Skip Full* become equal to the skip capacity, *Haul* to be carried away, and finally *Empty* to be dumped and become part of *Moved Soil.* Links SK1 through SK19 represent that Skip is initially waiting in the *Skip Wait* to be withdrawn to the *Set Skip* and then sequentially to the nodes *Skip Full* to get loaded, *Haul* to carry away the dirt and *Empty* to dump the dirt. The resource after dumping soil would be routed back to the Excavation cycle to the *Push exc*. to move back to the place where it gets loaded if the one meter length of the tunnel has not been excavated, if so it would be routed to the Concrete Lining cycle. In the Concrete Lining cycle the resource is withdrawn sequentially to the nodes *Push Conc*. *Lining* to move, *Place Rings* to position rings, *Grout Rings* to perform grouting the rings and *Return Conc. Lining* to move back either to the Excavation Cycle or Light Track Cycle. The resource would be routed to the Excavation cycle if the three meter length of the tunnel has not been excavated, if so it would be routed to the Light Track cycle. In the Light track cycle the resource is withdrawn sequentially to the nodes *Push light track* to move, *Position Light Track* to place, *Return Light Track* to move back and finally to *Skip Wait* where it had been started. Link M1 represents that three Meter resources that are generated before the activity *Return Light Track* finishes flow to the Meter Count Queue node and stay there. This process represents an operation for the first three meter along the length of tunneling and it repeats until the entire length of the tunnel has been constructed.

A Consolidator is used in the network to model the tunneling operation. The crew is not committed to completely fill a skip available in Consolidator in one step. The crew needs to leave the skip that is partially loaded two times to go back and dig and come back with new loads of soil to accomplish loading the skip.

The Load Soil is a Combi node which is followed by the *Soil* and *Crew 1* Queue nodes. The occurrence of the Load Soil is dependent to the availability of crew in the crew 1 Queue node and soil in the soil Queue node. In addition to this default constraint, by using SEMAPHORE statement another constraint is added to the model which make the occurrence of Load Soil dependent to the availability of a skip in *skip Full*.

As *Load Soil* activity finishes, amount of soil equal to the Skip capacity is generated. It is withdrawn to the *Skip Full* consolidator. *Skip Full* consolidator evaluates its ConsolidateWhen attribute to determine if it should consolidate. Consolidation happens when the amount of Soil in the *Skip Full* node is equal or greater than the capacity of the Skip.

Two Forks, RouteSkip1 and RouteSkip2, are used in the network to model the tunneling operation. As resource reaches the Forks, Forks choose one of its successors and sends the resource there. It is defined in the model that RouteSkip1 sends the resource back to the Push Exc. if one meter length of the tunnel has not been excavated, if so it sends the resource to the Load Rings activity in the Concrete Lining cycle. RouteSkip 2 sends the resource back to the excavation cycle if three meter length of the tunnel has not been excavated and lined, if so it routs the resource to the Load Light Track activity in the Light Track cycle.

Tunneling System Case Study - Foresight

In order to develop the tunneling operation in Foresight, a component oriented approach should be adapted such that each work unit indicates the construction of a physical component or sub-component of the facility under construction. The effective way to develop these models is through top-down, hierarchical approach. In this case the break down process starts with highest level component (the complete facility) and then it is broken down into the more detailed components which it includes. For the tunnel operation, the first two levels of breakdown are shown in Figure 5-2. The second level work units here are: *excavation* representing the cutting of the tunnel; *concrete lining* which includes grouting each 1 m in length of concrete ring segments in place; and *light track* used to haul a manually propelled train used for spoil removal and concrete ring segments delivery which are laid in 3 m lengths.

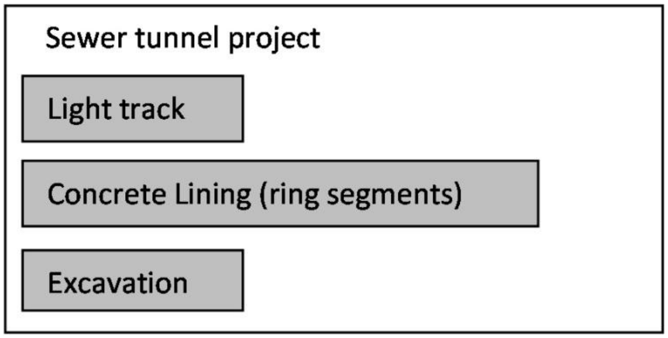


Figure 5-2. Two levels of work units for the Foresight sewer-tunnel model (Flood 2010b)

Since at the highest level of breakdown, the operation will be repeated in every 3 m through the length of the track, the work units, showing the construction of a 3 m through the length of the tunnel, can easily represent the model until the tunnel is completed. Likewise, to complete a 3 m length of the tunnel, the construction of 1m lined sections of the tunnel must be repeated 3 times. This is shown schematically in Figure 5-2 not showing any constraints. Figure 5-3, part A represents that for a 1 m length of tunnel, two work units *excavation* and *concrete lining* are considered. The work unit showing this 1 m lined section is repeated until a 3 m through the length of tunnel has been excavated and lined, after which the work unit, *light track,* for section is repeated until the completion of tunnel. Three work units, *light track*, *concrete lining*, and *excavation* are further broken down to their sub-work units in order to achieve certain level of detail that seems appropriate to analyze the model. Figure 5-3, part A represents further breakdown up to seven levels, with the deepest work units defined the *dig* and *load* nested within the *excavation* work unit. as When work units are added to the model more constraints can happen. Figure 5-3, part B represents the result of this, considering *time* against *tunnel length* for the first 3 meters of the tunneling operation. The main constraints, in this scenario, are as follows:

- The work unit showing 3 m *tunnel sections* are placed serially in both the *time* and *tunnel length* dimensions.

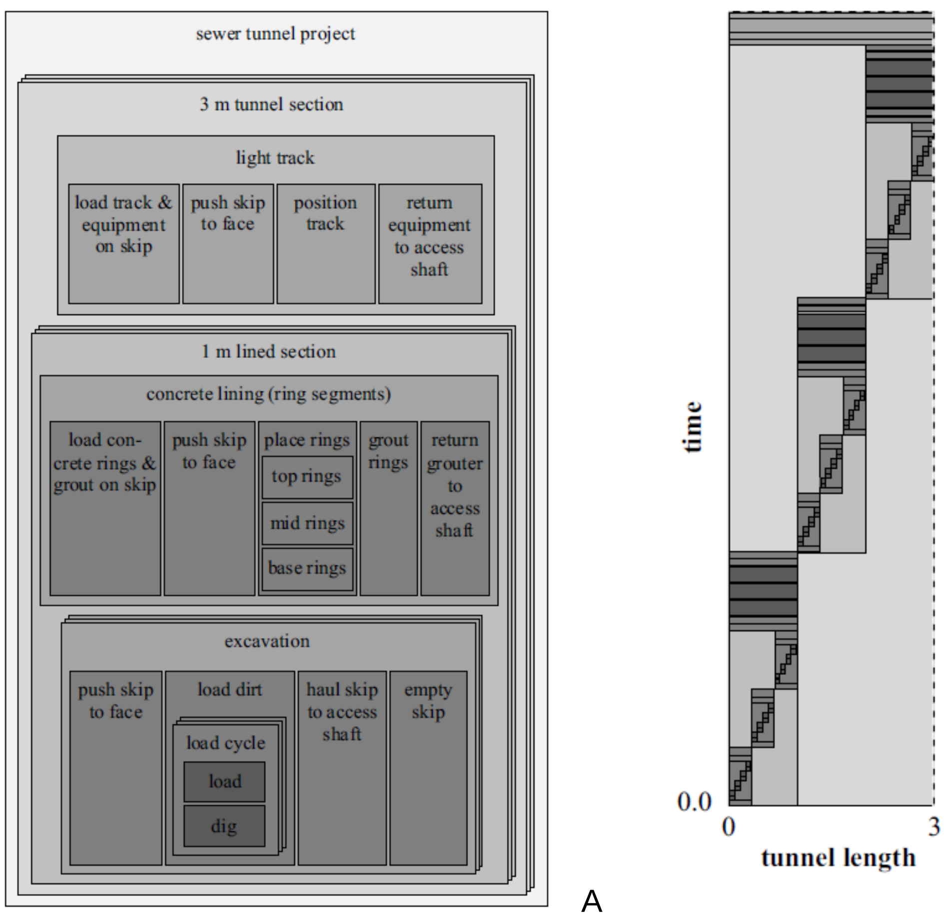
- The span of work unit showing the *sewer tunnel project* in the *tunnel length* direction is limited to the value of tunnel length.

- The work units of 3 *m tunnel section* start at the left side of the work unit, *sewer tunnel project,* and spanned to the right side of the work unit *sewer tunnel project*.

- The work units of 1 m *lined section* are placed serially both in the *time* and *tunnel length* dimensions.

- The work units of 1 m *lined section* extend from the left side to the right side of their work unit 3 *m tunnel section*.

- The two work units, *excavation* and *concrete lining,* are placed subsequently in the *time* dimension.



A B

Figure 5-3. Foresight modeling of a sewer-tunnel operation A) model hierarchy B) progress over first 3 meters (Flood 2010b)

Figure 5-4 represents the tunneling model for the 30 meters. For the purpose of readability, the model is only represented to a level of breakdown that represents the work units *excavation, concrete lining* and *light track* (color coded in orange, green, and blue respectively). As could be seen in the Figure 5-4 the operation of the project falls into a curve, which is caused due to the increase of the duration to remove soil and bring concrete ring segments to the tunnel face with the tunnel length. More refinements could be made to this model, in order to increase the accuracy and/or detail to provide the user with opportunity to decide about selection of equipment types. It is an option to add other attributes such as crew members and provide the opportunity for them to be shared between different work units simultaneously.

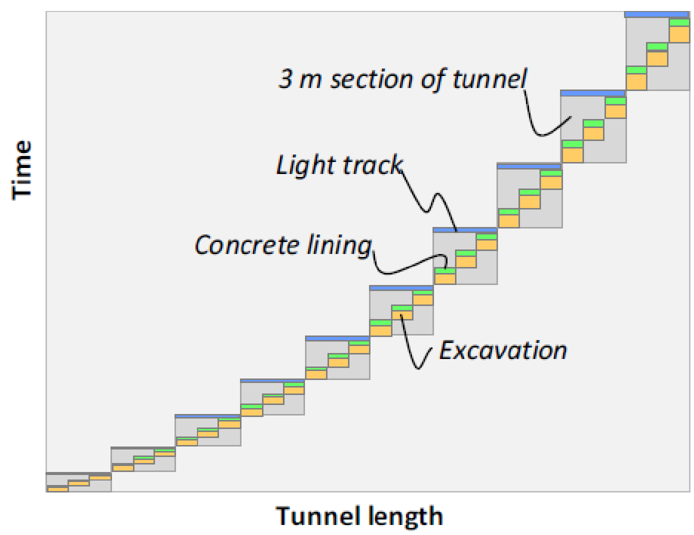


Figure 5-4. Foresight modeling of a sewer-tunnel operation for the first 30 meters (Flood 2010b)

To indicate the visual ability of these models, the condition where two separate crews will be engaged for tunneling, each starting at the same point but heading in opposite directions is considered. In a case that crew performance records indicate that 1 crew tends to operate about 50% faster than the other, the user is able to find a starting point that would optimize the total project duration. Figure 5-5 represents the model for a 60 m tunnel with two crews starting at the midpoint, with faster crew heading to the right and the slower crew heading to the left. The chart shows that the faster crew could start 3 m or 6 m to the left of the midpoint to optimize the project duration- both alternatives could be tested quickly.

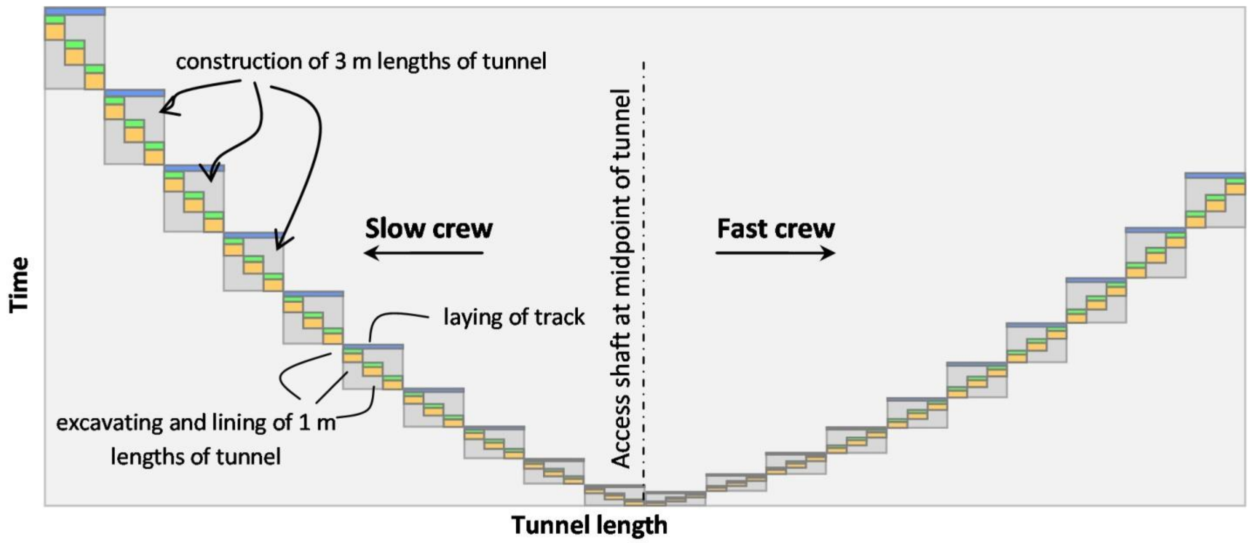


Figure 5-5. Foresight model of a tunneling system with 2 crews starting at center and heading in opposite directions (Flood 2010b)

Comparison Results

The complexity of the Foresight and Stroboscope modeling development for the tunneling system is compared in terms of the number of concepts and terms. As mentioned in chapter 2, the number of concepts is a measure of the depth of understanding expertise that is required to develop the model and the number of terms is a measure of amount of effort required to develop the model by the user and is considered to be any definition or item that is essential to define the structure and performance of the model.

To consider both crews (one heading in each direction), the Stroboscope model as shown in Figure 5-1 would have to be duplicated which it makes the model effectively twice the size.

Figure 5-6, part A represents the number of terms required for both Stroboscope and Foresight to define the tunneling system. It shows that the amount of information required by Foresight was just 23%of that using Stroboscope. Based on the comparison it can be conclude that Foresight and Stroboscope models are identical in terms of the process logic represented.

Figure 5-6, part B represents the number of concepts required for both Stroboscope and Foresight to define the tunneling system. It shows that the number of concepts employed by Foresight is about 14% of that employed by Stroboscope. Based on the comparison it can be conclude that Foresight and Stroboscope models are identical in terms of the process logic represented.

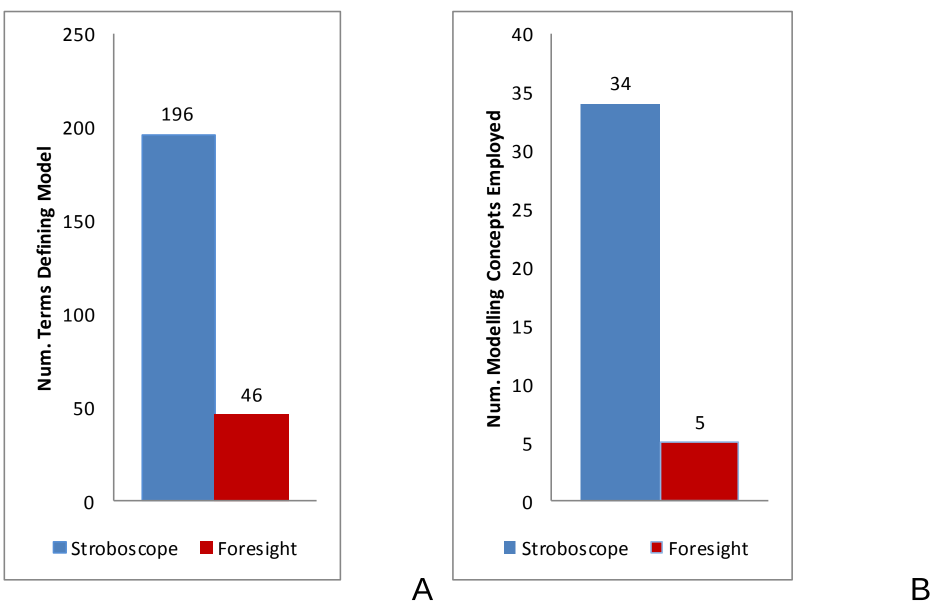


Figure 5-6. Complexity of the tunneling system for Foresight vs. Stroboscope in terms of number of A) terms and B) concepts

CHAPTER 6
QUALITATIVE COMPARISON

Graphical Insight

Planning methods such as linear scheduling that offer the graphical form of insight into the expected progress of a project would benefit the user. Using planning tools that serves the visual insight, the user can see the plan's logic and the relationships between activities. Graphical schedules depict a variety of important production information, such as production rates, and resource progress sequences and direction unlike network based planning tools such as CPM or bar charts. In addition, easy understanding of graphical schedules enhances overall project performance by making production information transparent to people involved with production (Howell 1999).

Simulation techniques separate the logic and performance within a system. Using simulation techniques the user must typically build the entire model (including defining all its parameters) first, before determining any measure of performance. In situations where the target is to perform sensitivity analysis and analyze the model response in variants number of resources, the planner has to modify the input text. The effect of this modification would be presented in the outfile result after running the model. However, the modeling network remains the same and the user cannot see how this modification actually affected the result. This can also reduce the accuracy of the modeling resultant since the user can't visually verify the effect of sensitivity analysis throughout the process.

For example, the Stroboscope process diagram shown in Figure 4-1 does not represent the system performance and it must be fully defined before the simulation can be executed to generate the performance results (shown in Figure 4-1). In contrast, the

Foresight model (Figure 4-3) integrates both logic and performance within one graph, so the impact of work units and constraints on system performance is visually apparent. Foresight models represent the logic and performance of a system in a single framework. In addition, the impact on performance can be seen on-the-fly as these elements are added, amended, and deleted. For example, by examining the Foresight model in Figure 5-5 it can be seen that by positioning the access shaft 3 m to the left would balance the two crews in a way that minimizes the project duration.

The previous three sections clearly show the benefits of the Foresight over Stroboscope in terms of the relative simplicity of the resultant models. Another important benefit of the Foresight over simulation is the visual insight provided by these models.

In conclusion, these characteristics of Foresight greatly extend the utility of the approach. First, they aid model verification (debugging) by allowing the user to see the impact on performance of each model edit. Second, they provide the user with a visual insight that assists the user to identify more optimal designs for a construction system.

Versatility

Foresight is a new form of visual modeling with the objective of combining the merits of other basic approaches that serves the simplicity and versatility. DES is very versatile in the sense that it can model any type of interaction between tasks and any type of construction process including repetitive and non-repetitive work (Flood 2010b). It considers stochastic duration, and incorporation of external factors like weather, labor productivity, and equipment breakdown (Sawhney 1998). Stroboscope, most sophisticated of the freely available simulation methods, consider variety of resources and their specific characteristic, make the state of simulation to control the sequence of tasks and their relative priorities, model resource selection schemes similar to real construction operations, model material utilization, consumption and production probabilistically (Martinez 1996).

Case studies were selected form very different types of construction processes to examine if the relative advantages are maintained and to portray the effectiveness of using the Foresight approach to provide the versatility of the simulation method. Foresight served the versatility to model the selected case studies considering the stochastic duration, productivity rate, and etc.

It could be conclude that Foresight can serve the user with the versatility to model different types of construction processes along with simplicity and visual insight.

CHAPTER 7
CONCLUSION AND FUTURE WORK

Foresight is a new construction modeling method that integrates the benefits of CPM, Linear scheduling, and discrete-event simulation. Foresight is designed with the objective of achieving the visual insight of linear scheduling, the simplicity of CPM, and the versatility of simulation. Although the linear scheduling method provides a modeler with a visual representation of the project, it is limited to repetitive activity projects that generally have a linear construction path. Foresight is designed to achieve that visual insight but provide more versatility in the types of projects that it can model. Although CPM provides a modeler with a basic visual representation of the process logic, it is less useful when trying to represent constraints.

Foresight involves hierarchical and interactive approaches to the development and analysis of process. Foresight is based upon principles which make it versatile enough to model the broad range of construction projects that until now have been modeled by using several different tools.

The method used to compare the complexity of Stroboscope and Foresight considered the amount of information required by each approach to define a model, the amount of effort the user had to input to complete the operation, the visual insight provided by each model, and the complexity of the resultant models. The number of different modeling concepts that had to be employed and the number of terms that had to be defined to complete the model are two metrics for measuring the complexity and ease of use of a model.

The purpose of the case studies was to model three unrelated projects in Stroboscope and apply the hierarchical constraint-based approach and compare. The first one refers to variations of an excavation system at a construction site. The theme of the second study is the concreting production and distribution system and the third study refers to a tunneling operation.

Overall, these three case studies represented very different types of projects on different levels of detail. All were scheduled in Stroboscope format originally, and all benefited from the use of Foresight, in terms of complexity and providing a visual scheduling tool.

Compared to simulation, the resultant models were significantly less complex and required far fewer modeling concepts to be understood. The complexity of modeling languages reflects how easy a planner can learn them. The complexity of the Foresight and Stroboscope modeling languages were compared in terms of number of concepts. The details of concepts for Stroboscope modeling languages are presented in Appendix F, Pages 104-110. The number of different modeling concepts that had to be learned by a user in order to develop a model in Foresight is about 6% of that employed by Stroboscope. Based on the comparison it can be concluded that Foresight and Stroboscope models are significantly identical in terms of the process logic represented. It could be concluded that a Foresight model-user must learn how to use the 5 base concepts to show each logical construct in a system, such as making sure that the excavator accomplishes the correct number of cycles to fully load a truck in excavation system. However, a Stroboscope model-user to achieve each logical construct must also learn how to figure out the various Stroboscope modeling components. The initial set-up to develop a Stroboscope model is challenging. It is necessary for the user to be knowledgeable enough about the construction process, its sequence and Stroboscope language in order to be able to develop a Stroboscope model. However when it is modeled correctly at the outset, the analysis of Stroboscope process throughout the project's progress would be significantly easier. The Stroboscope is useful where the user wishes to see how changes would impact the project, but these are only successful if the Stroboscope model is developed correctly with the appropriate expertise. User based on the initially developed Stroboscope model can determine potential alternatives to reduce the project's duration by allowing for the continuous evaluation of the project's progress. Foresight models have the advantage of representing the progress of work within the model structure. This provides visual insight into how the design of a process will impact its performance, aids model verification on-the-fly, and suggests ways of optimizing project performance.

Some limitations to the research and Foresight included that it is time-consuming- because it has not yet been implemented in software.

Future research should evaluate the ease with which new-users learn to develop and use Foresight models in comparison to the main alternative modeling approaches: CPM, linear scheduling, and simulation. Also, evaluating a model across a multi-dimensional resource space could prove very interesting. Researching software options for providing an electronic, automated use of the tool would also be helpful. Other future work should include performing a more detailed assessments of insight and quality, and developing metrics for measuring versatility and test quantitatively.

APPENDIX A
STROBOSCOPE EXCAVTION SYSTEM ONE TRUCK TYPE

Concepts:

- VARIABLE;
- DISPLAY;
- GENTYPE;
- COMPTYPE;
- QUEUE;
- COMBI;
- NORMAL;
- CONSOLIDATOR;
- SEMAPHORE;
- CONSOLIDATEWHEN;
- LINK LinkName PredecessorNode SuccessorNode;
- LINK LinkName PredecessorNode SuccessorNode ResourceTyoe;
- (When either predecessor or successor node are type-specific.)
- BEFOREEND;
- GENERATE;
- DURATION;
- ONEND;
- PRINT;
- StdOutput;
- Activity.Resource.Count;
- Format meaning of %.2f\n;
- SimTime;
- INIT;
- SIMULATEUNTIL;
- >=;
- QUEUE.CurCount;

Terms:

- Definition of VARIABLE, NumberOfExcavators
- Definition of VARIABLE, NumberOfTrucks
- Definition of VARIABLE, NumberOfSpace
- Definition of VARIABLE, SoilToMove
- Definition of VARIABLE, TruckCapacity
- Definition of VARIABLE, ExcavatorCapacity
- Definition of VARIABLE, ExpectedSetTruckTime
- Definition of VARIABLE, ExpectedLoadScoopTime
- Definition of VARIABLE, ExpectedSleuTime
- Definition of VARIABLE, ExpectedDigTime

- Definition of VARIABLE, ExpectedSleubackTime
- Definition of VARIABLE, ExpectedHaulTime
- Definition of VARIABLE, ExpectedDumpTime
- Definition of VARIABLE, ExpectedReturnTime

- DISPLAY Number of Trucks
- DISPLAY Number of Excavators
- DISPLAY Amount of soil to move
- DISPLAY Capacity of Truck
- DISPLAY Capacity of Excavator
- DISPLAY;
- DISPLAY;
- DISPLAY;

- Definition of Resource CHARTYPE Excavator
- Definition of Characteristic for Resource Excavator Capacity
- Definition of Resource SUBTYPE ExcavatorA
- Definition of Resource CHARTYPE Truck
- Definition of Characteristic for Resource Truck BucketSize
- Definition of Resource SUBTYPE TruckTypeA
- Definition of Resource GENTYPE Space
- Definition of Resource COMPTYPE Soil

- Definition of QUEUE ExcavatorsWait
- Definition of QUEUE TrucksWait
- Definition of QUEUE MvdSoil
- Definition of QUEUE Spot

- Definition of COMBI LoadScoop
- Definition of COMBI SetTruck

- Definition of NORMAL Sleuback
- Definition of NORMAL Haul
- Definition of NORMAL Dump
- Definition of NORMAL Return
- Definition of NORMAL Dig
- Definition of NORMAL Sleu

- Definition of CONSOLIDATOR Truckfull

- Definition of SEMAPHORE LoadScoop
- Definition of TruckFull.Truck.Count
- Definition of CONSOLIDATEWHEN

- Definition of TruckFull.Soil.Count
- TruckFull.TrucktypeA.Count
- TruckFull.TrucktypeA.Count & TruckFull.Soil.Count>=TrucktypeACapacity
- Definition of LINK E1
- Definition of LINK E2
- Definition of LINK E3
- Definition of LINK E4
- Definition of LINK E5
- Definition of BEFOREEND LoadScoop
- Definition of GENERATE Soil

- Definition of LINK S1
- Definition of LINK S2
- Definition of LINK S3
- Definition of LINK S4
- Definition of LINK T1
- Definition of LINK T2
- Definition of LINK T3
- Definition of LINK T4
- Definition of LINK T5
- Definition of LINK T6
- Definition of LINK SP1
- Definition of LINK SP2
- Definition of LINK SP3

- Definition of VARIABLE HoursSimulated
- Term of SimTime
- DURATION LoadScoop
- Definition of DURATION SetTruck
- Definition of DURATION Haul
- Definition of DURATION Dump
- Definition of DURATION Return
- Definition of DURATION Dig
- Definition of DURATION Sleu
- Definition of DURATION Sleuback

- Term of ONEND Dump
- Term of PRINT StdOutput
- Term of StdOutput
- Format %.2f\n
- Definition of INIT ExcavatorsWait
- Definition of INIT TrucksWait
- Definition of INIT Spot
- Definition of SIMULATEUNTIL

- Definition of MvdSoil.CurCount>=SoilToMove
- Definition of MvdSoil.CurCount
- Definition of DISPLAY

Stroboscope Modeling Language Text Input:

```
VARIABLE NumberOfExcavators 1;
VARIABLE NumberOfTrucks 1;
VARIABLE NumberOfSpace 1;
VARIABLE SoilToMove 100;/"Cm"
VARIABLE TrucktypeACapacity 10;/"Cm"
VARIABLE ExcavatorCapacity 1;/"Cm"
VARIABLE ExpectedSetTruckTime 0;/"Mins"
VARIABLE ExpectedLoadScoopTime 0.4;/"Mins"
VARIABLE ExpectedSleuTime 0.1;/"Mins"
VARIABLE ExpectedDigTime 0.8;/"Mins"
VARIABLE ExpectedSleubackTime 0.2;/"Mins"
VARIABLE ExpectedHaulTime 8;/"Mins"
VARIABLE ExpectedDumpTime 3;/"Mins"
VARIABLE ExpectedReturnTime 6.5;/"Mins"
DISPLAY;
DISPLAY "Number of Trucks         : "
   NumberOfTrucks;
DISPLAY "Number of Excavators         : "
   NumberOfExcavators;
DISPLAY;
DISPLAY "Amount of soil to move    : "
   SoilToMove
   "Cm";
DISPLAY "Capacity of Truck        : "
   TrucktypeACapacity
   "Cm";
DISPLAY "Capacity of Excavator        : "
   ExcavatorCapacity
   "Cm";
DISPLAY;
DISPLAY;
DISPLAY;
CHARTYPE Excavator ExcavatorCapacity;
SUBTYPE  Excavator ExcavatorA ExcavatorCapacity;
CHARTYPE Truck BucketSize;
SUBTYPE Truck TrucktypeA TrucktypeACapacity;
GENTYPE Space;
COMPTYPE Soil;
QUEUE ExcavatorsWait Excavator;
QUEUE TrucksWait Truck;
QUEUE MvdSoil Soil;
QUEUE Spot Space;
COMBI LoadScoop;
COMBI SetTruck;
NORMAL Sleuback;
NORMAL Haul;
NORMAL Dump;
NORMAL Return;
NORMAL Dig;
NORMAL Sleu;
CONSOLIDATOR TruckFull;
SEMAPHORE LoadScoop 'TruckFull.Truck.Count';
CONSOLIDATEWHEN TruckFull
    'TruckFull.TrucktypeA.Count & TruckFull.Soil.Count>=TrucktypeACapacity';
LINK E1 ExcavatorsWait LoadScoop;
LINK E2 LoadScoop Sleu Excavator;
LINK E3 Sleu Dig Excavator;
LINK E4 Dig Sleuback Excavator;
```

```
LINK E5 Sleuback ExcavatorsWait;
BEFOREEND LoadScoop GENERATE ExcavatorCapacity Soil;
LINK S1 LoadScoop TruckFull Soil;
LINK S2 TruckFull Haul Soil;
LINK S3 Haul Dump Soil;
LINK S4 Dump MvdSoil;
LINK T1 TrucksWait SetTruck;
LINK T2 SetTruck TruckFull Truck;
LINK T3 TruckFull Haul Truck;
LINK T4 Haul Dump Truck;
LINK T5 Dump Return Truck;
LINK T6 Return TrucksWait;
LINK SP1 Spot SetTruck;
LINK SP2 SetTruck TruckFull Space;
LINK SP3 TruckFull Spot;
VARIABLE HoursSimulated SimTime/60;
DURATION LoadScoop ExpectedLoadScoopTime;
DURATION SetTruck ExpectedSetTruckTime;
DURATION Haul ExpectedHaulTime;
DURATION Dump ExpectedDumpTime;
DURATION Return ExpectedReturnTime;
DURATION Dig ExpectedDigTime;
DURATION Sleu ExpectedSleuTime;
DURATION Sleuback ExpectedSleubackTime;
ONEND Dump PRINT StdOutput
    "The Current Simulation Time ONEND Dump is %.2f\n" SimTime/60;
INIT ExcavatorsWait NumberOfExcavators ExcavatorA;
INIT TrucksWait NumberOfTrucks TrucktypeA;
INIT Spot NumberOfSpace;
SIMULATEUNTIL 'MvdSoil.CurCount>=SoilToMove';
DISPLAY "
**************************************************************************
                Selected result from experiment
";

DISPLAY "Total Hour                  : "
    HoursSimulated
    "Hr";
```

Stroboscope Result from Running the Model:

```
Stroboscope Model Deterministic Basic Excavation (2070382880)


Number of Trucks         : 1
Number of Excavators         : 1

Amount of soil to move    : 100Cm
Capacity of Truck        : 10Cm
Capacity of Excavator        : 1Cm



The Current Simulation Time ONEND Dump is 0.41
The Current Simulation Time ONEND Dump is 0.94
The Current Simulation Time ONEND Dump is 1.46
The Current Simulation Time ONEND Dump is 1.99
The Current Simulation Time ONEND Dump is 2.51
The Current Simulation Time ONEND Dump is 3.03
The Current Simulation Time ONEND Dump is 3.56
The Current Simulation Time ONEND Dump is 4.08
The Current Simulation Time ONEND Dump is 4.60
The Current Simulation Time ONEND Dump is 5.13

************************************************************************
                Selected result from experiment

Total Hour                  : 5.125Hr

-------------------------------------
Execution    Time = 0.047 seconds
```

APPENDIX B
STROBOSCOPE EXCAVTION SYSTEM TWO TRUCK TYPE

Concepts:

- VARIABLE;
- DISPLAY;
- GENTYPE;
- COMPTYPE;
- CHARTYPE
- SUBTYPE
- QUEUE;
- COMBI;
- NORMAL;
- CONSOLIDATOR;
- SEMAPHORE;
- CONSOLIDATEWHEN;
- LINK LinkName PredecessorNode SuccessorNode;
- LINK LinkName PredecessorNode SuccessorNode ResourceTyoe;
- (When either predecessor or successor node are type-specific.)
- BEFOREEND;
- GENERATE;
- DURATION;
- ONEND;
- PRINT;
- StdOutput;
- Activity.Resource.Count;
- Format meaning of %.2f\n;
- SimTime;
- INIT;
- SIMULATEUNTIL;
- >=;
- QUEUE.CurCount;

Terms:

- Definition of VARIABLE, NumberOfExcavators
- Definition of VARIABLE, NumberOfTruckTypeA
- Definition of VARIABLE, NumberOfTruckTypeB
- Definition of VARIABLE, NumberOfSpace
- Definition of VARIABLE, SoilToMove
- Definition of VARIABLE, TruckTypeACapacity
- Definition of VARIABLE, TruckTypeBCapacity

- Definition of VARIABLE, ExcavatorCapacity
- Definition of VARIABLE, ExpectedSetTruckTime
- Definition of VARIABLE, ExpectedLoadScoopTime
- Definition of VARIABLE, ExpectedSleuTime
- Definition of VARIABLE, ExpectedDigTime
- Definition of VARIABLE, ExpectedSleubackTime
- Definition of VARIABLE, ExpectedHaulTime
- Definition of VARIABLE, ExpectedDumpTime
- Definition of VARIABLE, ExpectedReturnTime

- DISPLAY Number of Truck Type A
- DISPLAY Number of Truck Type B
- DISPLAY Number of Excavators
- DISPLAY Amount of soil to move
- DISPLAY Capacity of Truck Type A
- DISPLAY Capacity of Truck Type B
- DISPLAY Capacity of Excavator
- DISPLAY;
- DISPLAY;
- DISPLAY;

- Definition of Resource CHARTYPE Excavator
- Definition of Characteristic for Resource Excavator Capacity
- Definition of Resource SUBTYPE ExcavatorA
- Definition of Resource CHARTYPE Truck
- Definition of Characteristic for Resource Truck BucketSize
- Definition of Resource SUBTYPE TruckTypeA
- Definition of Resource SUBTYPE TruckTypeB
- Definition of Resource GENTYPE Space
- Definition of Resource COMPTYPE Soil

- Definition of QUEUE ExcavatorsWait
- Definition of QUEUE TrucksWait
- Definition of QUEUE MvdSoil
- Definition of QUEUE Spot

- Definition of COMBI LoadScoop
- Definition of COMBI SetTruck

- Definition of NORMAL Sleuback
- Definition of NORMAL Haul
- Definition of NORMAL Dump
- Definition of NORMAL Return
- Definition of NORMAL Dig

- Definition of NORMAL Sleu

- Definition of CONSOLIDATOR Truckfull

- Definition of SEMAPHORE LoadScoop
- Definition of TruckFull.Truck.Count
- Definition of CONSOLIDATEWHEN
- Definition of TruckFull.Soil.Count
- TruckFull.Truck.BucketSize
- TruckFull.Truck.Count &TruckFull.Soil.Count>=TruckFull.Truck.BucketSize
- Definition of LINK E1
- Definition of LINK E2
- Definition of LINK E3
- Definition of LINK E4
- Definition of LINK E5
- Definition of BEFOREEND LoadScoop
- Definition of GENERATE Soil

- Definition of LINK S1
- Definition of LINK S2
- Definition of LINK S3
- Definition of LINK S4
- Definition of LINK T1
- Definition of LINK T2
- Definition of LINK T3
- Definition of LINK T4
- Definition of LINK T5
- Definition of LINK T6
- Definition of LINK SP1
- Definition of LINK SP2
- Definition of LINK SP3

- Definition of VARIABLE HoursSimulated
- Term of SimTime
- DURATION LoadScoop
- Definition of DURATION SetTruck
- Definition of DURATION Haul
- Definition of DURATION Dump
- Definition of DURATION Return
- Definition of DURATION Dig
- Definition of DURATION Sleu
- Definition of DURATION Sleuback

- Term of ONEND Dump
- Term of PRINT StdOutput
- Term of StdOutput
- Format %.2f\n
- Definition of INIT ExcavatorsWait
- Definition of INIT TrucksWait Number of Truck Type A
- Definition of INIT TrucksWait Number of Truck Type B
- Definition of INIT Spot
- Definition of SIMULATEUNTIL
- Definition of MvdSoil.CurCount>=SoilToMove
- Definition of MvdSoil.CurCount

Stroboscope Modeling Language Text Input:

```
VARIABLE NumberOfExcavators 1;
VARIABLE NumberTrucktypeA 2;
VARIABLE NumberTrucktypeB 2;
VARIABLE NumberOfSpace 1;
VARIABLE TrucktypeACapacity 10;/"Cm"
VARIABLE TrucktypeBCapacity 15;/"Cm"
VARIABLE SoilToMove 100;/"Cm"
VARIABLE ExcavatorCapacity 1;/"Cm"
VARIABLE ExpectedSetTruckTime 0;/"Mins"
VARIABLE ExpectedLoadScoopTime 0.4;/"Mins"
VARIABLE ExpectedSleuTime 0.1;/"Mins"
VARIABLE ExpectedDigTime 0.8;/"Mins"
VARIABLE ExpectedSleubackTime 0.2;/"Mins"
VARIABLE ExpectedHaulTime 8;/"Mins"
VARIABLE ExpectedDumpTime 3;/"Mins"
VARIABLE ExpectedReturnTime 6.5;/"Mins"
DISPLAY;
DISPLAY "Number of Truck TypeA         : "
   NumberTrucktypeA;
DISPLAY "Number of Truck TypeB         : "
   NumberTrucktypeB;
DISPLAY "Number of Excavators         : "
   NumberOfExcavators;
DISPLAY;
DISPLAY "Amount of soil to move    : "
   SoilToMove
   "Cm";
DISPLAY "Capacity of Truck TypeA        : "
   TrucktypeACapacity
   "Cm";
DISPLAY "Capacity of Truck TypeB        : "
   TrucktypeBCapacity
   "Cm";
DISPLAY "Capacity of Excavator        : "
   ExcavatorCapacity
   "Cm";
DISPLAY;
DISPLAY;
DISPLAY;
CHARTYPE Excavator ExcavatorCapacity;
SUBTYPE  Excavator ExcavatorA ExcavatorCapacity;
CHARTYPE Truck BucketSize;
SUBTYPE Truck TrucktypeA TrucktypeACapacity;
SUBTYPE Truck TrucktypeB TrucktypeBCapacity;
GENTYPE Space;
COMPTYPE Soil;
QUEUE ExcavatorsWait Excavator;
QUEUE TrucksWait Truck;
QUEUE MvdSoil Soil;
QUEUE Spot Space;
COMBI LoadScoop;
COMBI SetTruck;
NORMAL Sleuback;
NORMAL Haul;
NORMAL Dump;
NORMAL Return;
NORMAL Dig;
NORMAL Sleu;
```

```
CONSOLIDATOR TruckFull;
SEMAPHORE LoadScoop 'TruckFull.Truck.Count';
CONSOLIDATEWHEN TruckFull
    'TruckFull.Truck.Count & TruckFull.Soil.Count>=TruckFull.Truck.BucketSize';
LINK E1 ExcavatorsWait LoadScoop;
LINK E2 LoadScoop Sleu Excavator;
LINK E3 Sleu Dig Excavator;
LINK E4 Dig Sleuback Excavator;
LINK E5 Sleuback ExcavatorsWait;
BEFOREEND LoadScoop GENERATE ExcavatorCapacity Soil;
LINK S1 LoadScoop TruckFull Soil;
LINK S2 TruckFull Haul Soil;
LINK S3 Haul Dump Soil;
LINK S4 Dump MvdSoil;
LINK T1 TrucksWait SetTruck;
LINK T2 SetTruck TruckFull Truck;
LINK T3 TruckFull Haul Truck;
LINK T4 Haul Dump Truck;
LINK T5 Dump Return Truck;
LINK T6 Return TrucksWait;
LINK SP1 Spot SetTruck;
LINK SP2 SetTruck TruckFull Space;
LINK SP3 TruckFull Spot;
VARIABLE HoursSimulated SimTime/60;
DURATION LoadScoop ExpectedLoadScoopTime;
DURATION SetTruck ExpectedSetTruckTime;
DURATION Haul ExpectedHaulTime;
DURATION Dump ExpectedDumpTime;
DURATION Return ExpectedReturnTime;
DURATION Dig ExpectedDigTime;
DURATION Sleu ExpectedSleuTime;
DURATION Sleuback ExpectedSleubackTime;
ONEND Dump PRINT StdOutput
    "The Current Simulation Time ONEND Dump is %.2f\n" SimTime/60;
INIT ExcavatorsWait NumberOfExcavators ExcavatorA;
INIT TrucksWait NumberTrucktypeA TrucktypeA;
INIT TrucksWait NumberTrucktypeB TrucktypeB;
INIT Spot NumberOfSpace;
SIMULATEUNTIL 'MvdSoil.CurCount>=SoilToMove';
DISPLAY "
********************************************************************************
                Selected result from experiment
";
DISPLAY "Time required to move soil : "
   HoursSimulated
   "   Hours";
```

Stroboscope Result from Running the Model:

```
Stroboscope Model Deterministic TruckType (1611489568)


Number of Truck TypeA         : 2
Number of Truck TypeB         : 2
Number of Excavators         : 1

Amount of soil to move    : 100Cm
Capacity of Truck TypeA        : 10Cm
Capacity of Truck TypeB        : 15Cm
Capacity of Excavator        : 1Cm



The Current Simulation Time ONEND Dump is 0.41
The Current Simulation Time ONEND Dump is 0.66
The Current Simulation Time ONEND Dump is 1.04
The Current Simulation Time ONEND Dump is 1.42
The Current Simulation Time ONEND Dump is 1.67
The Current Simulation Time ONEND Dump is 1.92
The Current Simulation Time ONEND Dump is 2.29
The Current Simulation Time ONEND Dump is 2.67

******************************************************************
                Selected result from experiment

Time required to move soil : 2.665   Hours

-------------------------------------
Execution    Time = 0.047 seconds
```

APPENDIX C
STROBOSCOPE EXCAVTION SYSTEM THREE TRUCK TYPE

Concepts:

- VARIABLE;
- DISPLAY;
- GENTYPE;
- COMPTYPE;
- CHARTYPE
- SUBTYPE
- QUEUE;
- COMBI;
- NORMAL;
- CONSOLIDATOR;
- SEMAPHORE;
- CONSOLIDATEWHEN;
- LINK LinkName PredecessorNode SuccessorNode;
- LINK LinkName PredecessorNode SuccessorNode ResourceTyoe;
- (When either predecessor or successor node are type-specific.)
- BEFOREEND;
- GENERATE;
- DURATION;
- ONEND;
- PRINT;
- StdOutput;
- Activity.Resource.Count;
- Format meaning of %.2f\n;
- SimTime;
- INIT;
- SIMULATEUNTIL;
- >=;
- QUEUE.CurCount;

Terms:

- Definition of VARIABLE, NumberOfExcavators
- Definition of VARIABLE, NumberOfTruckTypeA
- Definition of VARIABLE, NumberOfTruckTypeB
- Definition of VARIABLE, NumberOfTruckTypeC
- Definition of VARIABLE, NumberOfSpace
- Definition of VARIABLE, SoilToMove
- Definition of VARIABLE, TruckTypeACapacity

- Definition of VARIABLE, TruckTypeBCapacity
- Definition of VARIABLE, TruckTypeCCapacity
- Definition of VARIABLE, ExcavatorCapacity
- Definition of VARIABLE, ExpectedSetTruckTime
- Definition of VARIABLE, ExpectedLoadScoopTime
- Definition of VARIABLE, ExpectedSleuTime
- Definition of VARIABLE, ExpectedDigTime
- Definition of VARIABLE, ExpectedSleubackTime
- Definition of VARIABLE, ExpectedHaulTime
- Definition of VARIABLE, ExpectedDumpTime
- Definition of VARIABLE, ExpectedReturnTime

- DISPLAY Number of Truck Type A
- DISPLAY Number of Truck Type B
- DISPLAY Number of Truck Type C
- DISPLAY Number of Excavators
- DISPLAY Amount of soil to move
- DISPLAY Capacity of Truck Type A
- DISPLAY Capacity of Truck Type B
- DISPLAY Capacity of Truck Type C
- DISPLAY Capacity of Excavator
- DISPLAY;
- DISPLAY;
- DISPLAY;

- Definition of Resource CHARTYPE Excavator
- Definition of Characteristic for Resource Excavator Capacity
- Definition of Resource SUBTYPE ExcavatorA
- Definition of Resource CHARTYPE Truck
- Definition of Characteristic for Resource Truck BucketSize
- Definition of Resource SUBTYPE TruckTypeA
- Definition of Resource SUBTYPE TruckTypeB
- Definition of Resource SUBTYPE TruckTypeC
- Definition of Resource GENTYPE Space
- Definition of Resource COMPTYPE Soil

- Definition of QUEUE ExcavatorsWait
- Definition of QUEUE TrucksWait
- Definition of QUEUE MvdSoil
- Definition of QUEUE Spot

- Definition of COMBI LoadScoop
- Definition of COMBI SetTruck

- Definition of NORMAL Sleuback
- Definition of NORMAL Haul
- Definition of NORMAL Dump
- Definition of NORMAL Return
- Definition of NORMAL Dig
- Definition of NORMAL Sleu

- Definition of CONSOLIDATOR Truckfull

- Definition of SEMAPHORE LoadScoop
- Definition of TruckFull.Truck.Count
- Definition of CONSOLIDATEWHEN
- Definition of TruckFull.Soil.Count
- TruckFull.Truck.BucketSize
- TruckFull.Truck.Count & TruckFull.Soil.Count>=TruckFull.Truck.BucketSize'
- Definition of LINK E1
- Definition of LINK E2
- Definition of LINK E3
- Definition of LINK E4
- Definition of LINK E5
- Definition of BEFOREEND LoadScoop
- Definition of GENERATE Soil

- Definition of LINK S1
- Definition of LINK S2
- Definition of LINK S3
- Definition of LINK S4
- Definition of LINK T1
- Definition of LINK T2
- Definition of LINK T3
- Definition of LINK T4
- Definition of LINK T5
- Definition of LINK T6
- Definition of LINK SP1
- Definition of LINK SP2
- Definition of LINK SP3

- Definition of VARIABLE HoursSimulated
- Term of SimTime
- DURATION LoadScoop
- Definition of DURATION SetTruck
- Definition of DURATION Haul
- Definition of DURATION Dump

- Definition of DURATION Return
- Definition of DURATION Dig
- Definition of DURATION Sleu
- Definition of DURATION Sleuback

- Term of ONEND Dump
- Term of PRINT StdOutput
- Term of StdOutput
- Format %.2f\n
- Definition of INIT ExcavatorsWait
- Definition of INIT TrucksWait Number of Truck Type A
- Definition of INIT TrucksWait Number of Truck Type B
- Definition of INIT TrucksWait Number of Truck Type C
- Definition of INIT Spot
- Definition of SIMULATEUNTIL 'MvdSoil.CurCount>=SoilToMove';
- Definition of MvdSoil.CurCount
- Display

Stroboscope Modeling Language Text Input:

```
VARIABLE NumberOfExcavators 1;
VARIABLE NumberTrucktypeA 3;
VARIABLE NumberTrucktypeB 3;
VARIABLE NumberTrucktypeC 3;
VARIABLE NumberOfSpace 1;
VARIABLE TrucktypeACapacity 10;/"Cm"
VARIABLE TrucktypeBCapacity 15;/"Cm"
VARIABLE TrucktypeCCapacity 20;/"Cm"
VARIABLE SoilToMove 100;/"Cm"
VARIABLE ExcavatorCapacity 1;/"CY"
VARIABLE ExpectedSetTruckTime 0;/"Mins"
VARIABLE ExpectedLoadScoopTime 0.4;/"Mins"
VARIABLE ExpectedSleuTime 0.1;/"Mins"
VARIABLE ExpectedDigTime 0.8;/"Mins"
VARIABLE ExpectedSleubackTime 0.2;/"Mins"
VARIABLE ExpectedHaulTime 8;/"Mins"
VARIABLE ExpectedDumpTime 3;/"Mins"
VARIABLE ExpectedReturnTime 6.5;/"Mins"
DISPLAY;
DISPLAY "Number of Truck TypeA         : "
   NumberTrucktypeA;
DISPLAY "Number of Truck TypeB         : "
   NumberTrucktypeB;
DISPLAY "Number of Truck TypeC         : "
   NumberTrucktypeC;
DISPLAY "Number of Excavators         : "
   NumberOfExcavators;
DISPLAY;
DISPLAY "Amount of soil to move    : "
   SoilToMove
   "CY";
DISPLAY "Capacity of Truck TypeA        : "
   TrucktypeACapacity
   "CY";
DISPLAY "Capacity of Truck TypeB        : "
   TrucktypeBCapacity
   "CY";
DISPLAY "Capacity of Truck TypeC        : "
   TrucktypeCCapacity
   "CY";
DISPLAY "Capacity of Excavator        : "
   ExcavatorCapacity
   "CY";
DISPLAY;
DISPLAY;
DISPLAY;
CHARTYPE Excavator ExcavatorCapacity;
SUBTYPE  Excavator ExcavatorA ExcavatorCapacity;
CHARTYPE Truck BucketSize;
SUBTYPE Truck TrucktypeA TrucktypeACapacity;
SUBTYPE Truck TrucktypeB TrucktypeBCapacity;
SUBTYPE Truck TrucktypeC TrucktypeCCapacity;
GENTYPE Space;
COMPTYPE Soil;
QUEUE ExcavatorsWait Excavator;
QUEUE TrucksWait Truck;
QUEUE MvdSoil Soil;
QUEUE Spot Space;
```

```
COMBI LoadScoop;
COMBI SetTruck;
NORMAL Sleuback;
NORMAL Haul;
NORMAL Dump;
NORMAL Return;
NORMAL Dig;
NORMAL Sleu;
CONSOLIDATOR TruckFull;
SEMAPHORE LoadScoop 'TruckFull.Truck.Count';
CONSOLIDATEWHEN TruckFull
    'TruckFull.Truck.Count & TruckFull.Soil.Count>=TruckFull.Truck.BucketSize';
LINK E1 ExcavatorsWait LoadScoop;
LINK E2 LoadScoop Sleu Excavator;
LINK E3 Sleu Dig Excavator;
LINK E4 Dig Sleuback Excavator;
LINK E5 Sleuback ExcavatorsWait;
BEFOREEND LoadScoop GENERATE ExcavatorCapacity Soil;
LINK S1 LoadScoop TruckFull Soil;
LINK S2 TruckFull Haul Soil;
LINK S3 Haul Dump Soil;
LINK S4 Dump MvdSoil;
LINK T1 TrucksWait SetTruck;
LINK T2 SetTruck TruckFull Truck;
LINK T3 TruckFull Haul Truck;
LINK T4 Haul Dump Truck;
LINK T5 Dump Return Truck;
LINK T6 Return TrucksWait;
LINK SP1 Spot SetTruck;
LINK SP2 SetTruck TruckFull Space;
LINK SP3 TruckFull Spot;
VARIABLE HoursSimulated SimTime/60;
DURATION LoadScoop ExpectedLoadScoopTime;
DURATION SetTruck ExpectedSetTruckTime;
DURATION Haul ExpectedHaulTime;
DURATION Dump ExpectedDumpTime;
DURATION Return ExpectedReturnTime;
DURATION Dig ExpectedDigTime;
DURATION Sleu ExpectedSleuTime;
DURATION Sleuback ExpectedSleubackTime;
ONEND Dump PRINT StdOutput
    "The Current Simulation Time ONEND Dump is %.2f\n" SimTime/60;
INIT ExcavatorsWait NumberOfExcavators ExcavatorA;
INIT TrucksWait NumberTrucktypeA TrucktypeA;
INIT TrucksWait NumberTrucktypeB TrucktypeB;
INIT TrucksWait NumberTrucktypeC TrucktypeC;
INIT Spot NumberOfSpace;
SIMULATEUNTIL 'MvdSoil.CurCount>=SoilToMove';
DISPLAY "
********************************************************************************
                Selected result from experiment
";
DISPLAY "Time required to move soil : "
   HoursSimulated
   "   Hours";
```

Stroboscope Result from Running the Model:

```
Stroboscope Model Deterministic Three TruckType (27531552)


Number of Truck TypeA         : 3
Number of Truck TypeB         : 3
Number of Truck TypeC         : 3
Number of Excavators         : 1

Amount of soil to move    : 100CY
Capacity of Truck TypeA        : 10CY
Capacity of Truck TypeB        : 15CY
Capacity of Truck TypeC        : 20CY
Capacity of Excavator        : 1CY



The Current Simulation Time ONEND Dump is 0.41
The Current Simulation Time ONEND Dump is 0.66
The Current Simulation Time ONEND Dump is 0.91
The Current Simulation Time ONEND Dump is 1.29
The Current Simulation Time ONEND Dump is 1.67
The Current Simulation Time ONEND Dump is 2.04
The Current Simulation Time ONEND Dump is 2.54
The Current Simulation Time ONEND Dump is 3.04

************************************************************************
                Selected result from experiment

Time required to move soil : 3.04   Hours

-------------------------------------
Execution    Time = 0.047 seconds
```

APPENDIX D
STROBOSCOPE CONCRETE PRODUCTION AND DISTRIBUTION SYSTEM

Concepts:

- VARIABLE;
- DISPLAY;
- GENTYPE;
- CHARTYPE;
- SUBTYPE;
- COMPTYPE;
- QUEUE;
- COMBI;
- NORMAL;
- LINK LinkName PredecessorNode SuccessorNode;
- LINK LinkName PredecessorNode SuccessorNode ResourceTyoe;
- (When either predecessor or successor node are type-specific.)
- BEFOREEND;
- GENERATE;
- ENOUGH;
- DRAWUNTIL;
- DURATION;
- Activity.Resource.Count;
- SimTime;
- ONEND;
- PRINT;
- StdOutput;
- %.2f\n ;
- INIT;
- SIMULATEUNTIL;
- >=;
- QUEUE.CurCount;

Terms:

- Definition of VARIABLE, NumberOfMixers
- Definition of VARIABLE, NumberTrucks
- Definition of VARIABLE, NumberOfSpaces
- Definition of VARIABLE, NumberOfPermits
- Definition of VARIABLE, NumberOfSpaceys
- Definition of VARIABLE, TruckCapacity
- Definition of VARIABLE, ConcreteToMove
- Definition of VARIABLE, MixerCapacity

- Definition of VARIABLE, HooperCapacity
- Definition of VARIABLE, ExpectedLoadMaterialsTime
- Definition of VARIABLE, ExpectedMixTime
- Definition of VARIABLE, ExpectedPourConcreteToHopperTime
- Definition of VARIABLE, ExpectedRecieveConcreteFromHopperTime
- Definition of VARIABLE, ExpectedTravelTime
- Definition of VARIABLE, ExpectedPourTime
- Definition of VARIABLE, ExpectedReturnTime

- DISPLAY Number of Mixers
- DISPLAY Number of Trucks
- DISPLAY Amount of Concrete to move
- DISPLAY Capacity of Mixer
- DISPLAY Capacity of Truck
- DISPLAY Capacity of Hooper
- DISPLAY Amount of soil to move
- DISPLAY
- DISPLAY

- Definition of Resource CHARTYPE Mixer
- Definition of Characteristic for Resource MixerCapacity
- Definition of Resource SUBTYPE MixerA
- Definition of Resource SUBTYPE MixerA Capacity
- Definition of Resource CHARTYPE Truck
- Definition of Characteristic for Resource Truck BucketSize
- Definition of Resource SUBTYPE TruckTypeA
- Definition of Resource SUBTYPE TruckTypeA Capacity
- Definition of Resource GENTYPE Space
- Definition of Resource GENTYPE Spacey
- Definition of Resource GENTYPE Permit
- Definition of Resource COMPTYPE Concrete

- Definition of QUEUE MixerWaitLoading
- Definition of QUEUE MixerWaitPouring
- Definition of QUEUE TrucksWait
- Definition of QUEUE MvdConcrete
- Definition of QUEUE Spot1
- Definition of QUEUE Spot2
- Definition of QUEUE Allow
- Definition of QUEUE ConcreteWait
- Definition of QUEUE Spoty
- Definition of QUEUE SpaceyWait1
- Definition of QUEUE SpaceyWait2

- Definition of COMBI LoadMaterials
- Definition of COMBI PourConcreteToHopper
- Definition of COMBI RecieveConcreteFromHopper

- Definition of NORMAL Mix
- Definition of NORMAL Travel
- Definition of NORMAL Pour
- Definition of NORMAL Return

- Definition of LINK M1
- Definition of LINK M2
- Definition of LINK M3
- Definition of LINK M4
- Definition of LINK M5
- Definition of LINK Sp1
- Definition of LINK Sp2
- Definition of DRAWUNTIL Sp2
- Definition of RecieveConcreteFromHopper.Space.Count
- Definition of RecieveConcreteFromHopper.Space.Count>=TruckCapacity
- Definition of LINK Sp3
- Definition of LINK Sp4
- Definition of BEFOREEND PourConcreteToHopper
- Definition of GENERATE Concrete
- Definition of LINK C1
- Definition of LINK C2
- Definition of LINK C3
- Definition of ENOUGH
- Definition of ConcreteWait.CurCount
- ConcreteWait.CurCount>=TruckCapacity
- Definition of DRAWUNTIL
- Definition of RecieveConcreteFromHopper.Concrete.Count
- Definition of RecieveConcreteFromHopper.Truck.Count
- RecieveConcreteFromHopper.Concrete.Count>=RecieveConcreteFromHopper.
- Definition of LINK C4
- Definition of LINK C5
- Definition of LINK T1
- Definition of LINK T2
- Definition of LINK T3
- Definition of LINK T4
- Definition of LINK T5
- Definition of LINK A1
- Definition of LINK A2
- Definition of LINK Spy1
- Definition of LINK Spy2

- Definition of LINK Spy3
- Definition of LINK Spy4
- Definition of LINK Spy5
- Definition of LINK Spy6
- Definition of DRAWUNTIL
- RecieveConcreteFromHopper.Spacey.Count
- RecieveConcreteFromHopper.Spacey.Count>=TruckCapacity
- Definition of LINK Spy7

- Definition of VARIABLE HoursSimulated
- Term of SimTime
- DURATION LoadMaterials
- Definition of DURATION PourConcreteToHopper
- Definition of DURATION RecieveConcreteFromHopper
- Definition of DURATION Mix
- Definition of DURATION Travel
- Definition of DURATION Pour
- Definition of DURATION Return

- Term of ONEND Pour
- Term of PRINT StdOutput
- Term of StdOutput
- Format %.2f\n

- Definition of INIT MixerWaitLoading
- Definition of INIT TrucksWait Number of Truck Type A
- Definition of INIT Spot2
- Definition of INIT Spoty
- Definition of INIT Allow

- Definition of SIMULATEUNTIL 'SimTime/60>=2';
- Definition of DISPLAY "Time required to move soil :”

Stroboscope Modeling Language Text Input:

```
VARIABLE NumberOfMixers 1;
VARIABLE NumberTrucks 2;
VARIABLE NumberOfSpaces 5;
VARIABLE NumberOfPermits 1;
VARIABLE NumberOfSpaceys 5;
VARIABLE TruckCapacity 3;/"Cm"
VARIABLE ConcreteToMove 50;/"Cm"
VARIABLE MixerCapacity 1;/"Cm"
VARIABLE HooperCapacity 5;/"Cm"
VARIABLE ExpectedLoadMaterialsTime 0.2;/"MINs"
VARIABLE ExpectedMixTime 0.2;/"MINs"
VARIABLE ExpectedPourConcreteToHopperTime 0.5;/"MINs"
VARIABLE ExpectedRecieveConcreteFromHopperTime 0.6;/"MINs"
VARIABLE ExpectedTravelTime 6;/"MINs"
VARIABLE ExpectedPourTime 0.5;/"MINs"
VARIABLE ExpectedReturnTime 8;/"MINs"
DISPLAY;
DISPLAY "Number of Mixers         : "
   NumberOfMixers;
DISPLAY "Number of Trucks         : "
   NumberTrucks;
DISPLAY;
DISPLAY "Amount of Concrete to move    : "
   ConcreteToMove
   "Cm";
DISPLAY "Capacity of Mixer        : "
   MixerCapacity
   "Cm";
DISPLAY "Capacity of Truck        : "
   TruckCapacity
   "Cm";
DISPLAY "Capacity of Hooper        : "
   HooperCapacity
   "Cm";
DISPLAY;
CHARTYPE Mixer Capacity;
SUBTYPE Mixer MixerA MixerCapacity;
CHARTYPE Truck Bucketsize;
SUBTYPE Truck TruckA TruckCapacity;
GENTYPE Space;
GENTYPE Spacey;
GENTYPE Permit;
COMPTYPE Concrete;
QUEUE MixerWaitLoading Mixer;
QUEUE MixerWaitPouring Mixer;
QUEUE TrucksWait Truck;
QUEUE MvdConcrete Concrete;
QUEUE Spot1 Space;
QUEUE Spot2 Space;
QUEUE Allow Permit;
QUEUE ConcreteWait Concrete;
QUEUE Spoty Spacey;
QUEUE SpaceyWait1 Spacey;
QUEUE SpaceyWait2 Spacey;
COMBI LoadMaterials;
COMBI PourConcreteToHopper;
COMBI RecieveConcreteFromHopper;
NORMAL Mix;
```

```
NORMAL Travel;
NORMAL Pour;
NORMAL Return;
LINK M1 MixerWaitLoading LoadMaterials;
LINK M2 LoadMaterials Mix Mixer;
LINK M3 Mix MixerWaitPouring;
LINK M4 MixerWaitPouring PourConcreteToHopper;
LINK M5 PourConcreteToHopper MixerWaitLoading;
LINK Sp1 PourConcreteToHopper Spot1;
LINK Sp2 Spot1 RecieveConcreteFromHopper;
DRAWUNTIL Sp2 RecieveConcreteFromHopper.Space.Count>=TruckCapacity;
LINK Sp3 RecieveConcreteFromHopper Spot2;
LINK Sp4 Spot2 PourConcreteToHopper;
BEFOREEND PourConcreteToHopper GENERATE MixerCapacity Concrete;
LINK C1 PourConcreteToHopper ConcreteWait;
LINK C2 ConcreteWait RecieveConcreteFromHopper;
ENOUGH C2 ConcreteWait.CurCount>=TruckCapacity;
DRAWUNTIL C2 RecieveConcreteFromHopper.Concrete.Count>=RecieveConcreteFromHopper.Truck.Count*TruckCapacity;
LINK C3 RecieveConcreteFromHopper Travel Concrete;
LINK C4 Travel Pour Concrete;
LINK C5 Pour MvdConcrete;
LINK T1 RecieveConcreteFromHopper Travel Truck;
LINK T2 Travel Pour Truck;
LINK T3 Pour Return Truck;
LINK T4 Return TrucksWait;
LINK T5 TrucksWait RecieveConcreteFromHopper;
LINK A1 RecieveConcreteFromHopper Allow;
LINK A2 Allow RecieveConcreteFromHopper;
LINK Spy1 Spoty LoadMaterials;
LINK Spy2 LoadMaterials Mix Spacey;
LINK Spy3 Mix SpaceyWait1;
LINK Spy4 SpaceyWait1 PourConcreteToHopper;
LINK Spy5 PourConcreteToHopper SpaceyWait2;
LINK Spy6 SpaceyWait2 RecieveConcreteFromHopper;
DRAWUNTIL Spy6 RecieveConcreteFromHopper.Spacey.Count>=TruckCapacity;
LINK Spy7 RecieveConcreteFromHopper Spoty;
VARIABLE HoursSimulated SimTime/60;
DURATION LoadMaterials ExpectedLoadMaterialsTime;
DURATION PourConcreteToHopper ExpectedPourConcreteToHopperTime;
DURATION RecieveConcreteFromHopper ExpectedRecieveConcreteFromHopperTime;
DURATION Mix ExpectedMixTime;
DURATION Travel ExpectedTravelTime;
DURATION Pour ExpectedPourTime;
DURATION Return ExpectedReturnTime;
ONEND Pour PRINT StdOutput
    "The Current Simulation Time ONEND Pour is %.2f\n" SimTime/60;
INIT MixerWaitLoading NumberOfMixers MixerA;
INIT TrucksWait NumberTrucks TruckA;
INIT Spot2 NumberOfSpaces;
INIT Spoty NumberOfSpaceys;
INIT Allow NumberOfPermits;
SIMULATEUNTIL 'SimTime/60>=2.175';
DISPLAY "
****************************************************************************************************
```

```
                Selected result from experiment
";
DISPLAY "Time required to move soil : "
   HoursSimulated
   "   Hours";
```

Stroboscope Result from Running the Model:

```
Stroboscope Model Hopper (899083040)


Number of Mixers         : 1
Number of Trucks         : 2

Amount of Concrete to move    : 50Cm
Capacity of Mixer        : 1Cm
Capacity of Truck        : 3Cm
Capacity of Hooper        : 5Cm

The Current Simulation Time ONEND Pour is 0.16
The Current Simulation Time ONEND Pour is 0.18
The Current Simulation Time ONEND Pour is 0.42
The Current Simulation Time ONEND Pour is 0.43
The Current Simulation Time ONEND Pour is 0.67
The Current Simulation Time ONEND Pour is 0.68
The Current Simulation Time ONEND Pour is 0.92
The Current Simulation Time ONEND Pour is 0.93
The Current Simulation Time ONEND Pour is 1.17
The Current Simulation Time ONEND Pour is 1.18
The Current Simulation Time ONEND Pour is 1.42
The Current Simulation Time ONEND Pour is 1.44
The Current Simulation Time ONEND Pour is 1.67
The Current Simulation Time ONEND Pour is 1.69
The Current Simulation Time ONEND Pour is 1.93
The Current Simulation Time ONEND Pour is 1.94
The Current Simulation Time ONEND Pour is 2.18

***********************************************************
                Selected result from experiment

Time required to move soil : 2.1766667   Hours

-------------------------------------
Execution    Time = 0.031 seconds
```

APPENDIX E
STROBOSCOPE SEWER TUNNELING SYSTEM

Concepts:

- VARIABLE;
- SAVEVALUE;
- DISPLAY;
- GENTYPE;
- CHARTYPE;
- SUBTYPE;
- COMPTYPE;
- QUEUE;
- COMBI;
- NORMAL;
- FORK;
- CONSOLIDATOR;
- SEMAPHORE;
- CONSOLIDATEWHEN;
- &
- BEFOREEND;
- GENERATE;
- LINK LinkName PredecessorNode SuccessorNode;
- LINK LinkName PredecessorNode SuccessorNode ResourceTyoe;
- (When either predecessor or successor node are type-specific.)
- ASSIGN
- +
- <
- STRENGTH;
- DURATION;
- Activity.Resource.Count;
- SimTime;
- ONEND;
- PRINT;
- StdOutput;
- %.2f\n ;
- INIT;
- SIMULATEUNTIL;
- >=;
- QUEUE.CurCount;

Terms:

- Definition of VARIABLE, NumberOfCrews
- Definition of VARIABLE, NumberOfSkips
- Definition of VARIABLE, SkipCapacity
- Definition of VARIABLE, SoilGenerate
- Definition of VARIABLE, StrengthOfSK5
- Definition of SAVEVALUE, RoundAWeight
- Definition of SAVEVALUE, RoundBWeight
- Definition of SAVEVALUE, MeterOfTunnel
- Definition of VARIABLE, ZeroRounds
- Definition of VARIABLE, FinTunnelLength
- Definition of VARIABLE, TunnelLengthOfThree
- Definition of SAVEVALUE, ExpectedDigTime
- Definition of SAVEVALUE, ExpectedLoadSoilTime
- Definition of SAVEVALUE, ExpectedHaulTime
- Definition of SAVEVALUE, ExpectedEmptyTime
- Definition of SAVEVALUE, ExpectedPushExcavationTime
- Definition of SAVEVALUE, ExpectedSetSkipTime
- Definition of SAVEVALUE, ExpectedLoadRingsTime
- Definition of SAVEVALUE, ExpectedPushConcreteLiningTime
- Definition of SAVEVALUE, ExpectedPlaceRingsTime
- Definition of SAVEVALUE, ExpectedGroutRingsTime
- Definition of SAVEVALUE, ExpectedReturnConcreteLiningTime
- Definition of SAVEVALUE, ExpectedLoadLightTrackTime
- Definition of SAVEVALUE, ExpectedPushLightTrackTime
- Definition of SAVEVALUE, ExpectedPositionLightTrackTime
- Definition of SAVEVALUE, ExpectedReturnLightTrackTime
- DISPLAY
- DISPLAY
- DISPLAY
- DISPLAY;
- DISPLAY;
- DISPLAY;
- Definition of Resource COMPTYPE Soil
- Definition of Resource CHARTYPE Skip
- Definition of Characteristic for Resource SkipCapacity
- Definition of Resource SUBTYPE SkipA
- Definition of Resource SUBTYPE SkipA Capacity
- Definition of Resource GENTYPE Crew
- Definition of Resource COMPTYPE Meter

- Definition of QUEUE CrewWait1
- Definition of QUEUE CrewWait2

- Definition of QUEUE SoilWait
- Definition of QUEUE MvdSoil
- Definition of QUEUE SkipWait
- Definition of QUEUE MeterCount
- Definition of COMBI LoadSoil
- Definition of COMBI SetSkip
- Definition of COMBI Dig

- Definition of NORMAL Haul
- Definition of NORMAL Empty
- Definition of NORMAL PushExcavation
- Definition of NORMAL LoadRings
- Definition of NORMAL PushConcreteLining
- Definition of NORMAL PlaceRings
- Definition of NORMAL GroutRings
- Definition of NORMAL ReturnConcreteLining
- Definition of NORMAL LoadLightTrack
- Definition of NORMAL PushLightTrack
- Definition of NORMAL PositionLightTrack
- Definition of NORMAL ReturnLightTrack;

- Definition of FORK RouteSkip1
- Definition of FORK RouteSkip2

- Definition of CONSOLIDATOR SkipFull

- Definition of SEMAPHORE

- Definition of SkipFull.Skip.Count
- Definition of SkipFull.Soil.Count
- Definition of CONSOLIDATEWHEN
- Definition of SkipFull.Skip.Capacity
- Definition of of SkipFull.Soil.Count>=SkipFull.Skip.Capacity
- Definition of BEFOREEND
- Definition of GENERATE

- Definition of LINK S1
- Definition of LINK S2
- Definition of LINK S3
- Definition of LINK S4
- Definition of LINK S5
- Definition of LINK S6
- Definition of LINK C1
- Definition of LINK C2

- Definition of LINK C3
- Definition of LINK C4
- Definition of LINK SK1
- Definition of LINK SK2
- Definition of LINK SK3
- Definition of LINK SK4
- Definition of LINK SK19
- Definition of LINK SK5
- Definition of LINK SK6
- Definition of LINK SK7
- Definition of BEFOREEND Haul
- Definition of ASSIGN RoundAWeight
- RoundAWeight+1
- Definition of STRENGTH SK5
- RoundAWeight<StrengthOfSK5
- Definition of STRENGTH SK7
- RoundAWeight>=StrengthOfSK5
- Definition of BEFOREEND LoadRings
- Definition of ASSIGN RoundAWeight
- Definition of BEFOREEND LoadRings
- Definition of ASSIGN RoundBWeight
- RoundBWeight+1
- Definition of LINK SK8
- Definition of LINK SK9
- Definition of LINK SK10
- Definition of LINK SK11
- Definition of LINK SK12
- Definition of LINK SK13
- Definition of BEFOREEND ReturnLightTrack
- Definition of GENERATE Meter
- Definition of LINK SK14
- Definition of LINK SK15
- Definition of LINK SK16
- Definition of LINK SK17
- Definition of LINK SK18
- Definition of STRENGTH SK13
- RoundBWeight<StrengthOfSK5
- Definition of STRENGTH SK14
- RoundBWeight>=StrengthOfSK5
- Definition of BEFOREEND LoadLightTrack
- ASSIGN RoundBWeight
- BEFOREEND ReturnLightTrack
- ASSIGN MeterOfTunnel
- MeterOfTunnel+1

- BEFOREEND
- ASSIGN
- ExpectedDigTime+0.05*ExpectedDigTime
- BEFOREEND
- ASSIGN
- ExpectedLoadSoilTime+0.05*ExpectedLoadSoilTime
- BEFOREEND
- ASSIGN
- ExpectedHaulTime+0.05*ExpectedHaulTime
- BEFOREEND
- ASSIGN
- ExpectedEmptyTime+0.05*ExpectedEmptyTime
- BEFOREEND
- ASSIGN
- ExpectedPushExcavationTime+0.05*ExpectedPushExcavationTime
- BEFOREEND
- ASSIGN
- ExpectedSetSkipTime+0.05*ExpectedSetSkipTime
- BEFOREEND
- ASSIGN
- ExpectedLoadRingsTime+0.05*ExpectedLoadRingsTime
- BEFOREEND
- ASSIGN
- ExpectedPushConcreteLiningTime+0.05*ExpectedPushConcreteLiningTime
- BEFOREEND
- ASSIGN
- ExpectedPlaceRingsTime+0.05*ExpectedPlaceRingsTime
- BEFOREEND
- ASSIGN
- ExpectedGroutRingsTime+0.05*ExpectedGroutRingsTime
- BEFOREEND
- ASSIGN
- ExpectedReturnConcreteLiningTime+0.05*ExpectedReturnConcreteLiningTime
- BEFOREEND
- ASSIGN
- ExpectedLoadLightTrackTime+0.05*ExpectedLoadLightTrackTime
- BEFOREEND
- ASSIGN
- ExpectedPushLightTrackTime+0.05*ExpectedPushLightTrackTime
- BEFOREEND
- ASSIGN
- ExpectedPositionLightTrackTime+0.05*ExpectedPositionLightTrackTime
- BEFOREEND
- ASSIGN

- ExpectedReturnLightTrackTime+0.05*ExpectedReturnLightTrackTime
- LINK M1
- Definition of MeterCount.CurCount

- Definition of VARIABLE HoursSimulated
- Term of SimTime
- DURATION Dig
- Definition of DURATION LoadSoil
- Definition of DURATION Haul
- Definition of DURATION Empty
- Definition of DURATION PushExcavation
- Definition of DURATION SetSkip
- Definition of DURATION LoadRings
- Definition of DURATION PushConcreteLining
- Definition of DURATION PlaceRings
- Definition of DURATION GroutRings
- Definition of DURATION ReturnConcreteLining
- Definition of DURATION LoadLightTrack
- Definition of DURATION PushLightTrack
- Definition of DURATION PositionLightTrack
- Definition of DURATION ReturnLightTrack

- Term of ONEND ReturnLightTrack
- Term of PRINT StdOutput
- Term of StdOutput
- Format %.2f\n

- Definition of INIT CrewWait1
- Definition of INIT SkipWait

- Definition of SIMULATEUNTIL 'SimTime/60>=2';

- MeterCount.CurCount>=FinTunnelLength
- Definition of DISPLAY "Time required to move soil :”

Stroboscope Modeling Language Text Input:

```
VARIABLE NumberOfCrews 1;
VARIABLE NumberOfSkip 1;
VARIABLE SkipCapacity 3;
VARIABLE SoilGenerate 1;
VARIABLE StrengthOfSK5 3;
SAVEVALUE RoundAWeight 0;
SAVEVALUE RoundBWeight 0;
SAVEVALUE MeterOfTunnel 0;
VARIABLE ZeroRounds 0;
VARIABLE FinTunnelLength 9;
VARIABLE TunnelLengthOfThree 3;
SAVEVALUE ExpectedDigTime 2;
SAVEVALUE ExpectedLoadSoilTime 2;
SAVEVALUE ExpectedHaulTime 2;
SAVEVALUE ExpectedEmptyTime 1;
SAVEVALUE ExpectedPushExcavationTime 3;
SAVEVALUE ExpectedSetSkipTime 4;
SAVEVALUE ExpectedLoadRingsTime 5;
SAVEVALUE ExpectedPushConcreteLiningTime 8;
SAVEVALUE ExpectedPlaceRingsTime 7;
SAVEVALUE ExpectedGroutRingsTime 2;
SAVEVALUE ExpectedReturnConcreteLiningTime 3;
SAVEVALUE ExpectedLoadLightTrackTime 4;
SAVEVALUE ExpectedPushLightTrackTime 5;
SAVEVALUE ExpectedPositionLightTrackTime 6;
SAVEVALUE ExpectedReturnLightTrackTime 5;
DISPLAY "Number of skips:"
    NumberOfSkip;
DISPLAY "Number of Crews:"
    NumberOfCrews;
DISPLAY "Length of Tunel:"
    FinTunnelLength;
DISPLAY;
DISPLAY;
DISPLAY;
COMPTYPE Soil;
CHARTYPE Skip Capacity;
SUBTYPE Skip SkipA SkipCapacity;
GENTYPE Crew;
COMPTYPE Meter;
QUEUE CrewWait1 Crew;
QUEUE CrewWait2 Crew;
QUEUE SoilWait Soil;
QUEUE MvdSoil Soil;
QUEUE SkipWait Skip;
QUEUE MeterCount Meter;
COMBI LoadSoil;
COMBI SetSkip;
COMBI Dig;
NORMAL Haul;
NORMAL Empty;
NORMAL PushExcavation;
NORMAL LoadRings;
NORMAL PushConcreteLining;
NORMAL PlaceRings;
NORMAL GroutRings;
NORMAL ReturnConcreteLining;
NORMAL LoadLightTrack;
```

```
NORMAL PushLightTrack;
NORMAL PositionLightTrack;
NORMAL ReturnLightTrack;

FORK RouteSkip1 Skip;
FORK RouteSkip2 Skip;

CONSOLIDATOR SkipFull;
SEMAPHORE LoadSoil 'SkipFull.Skip.Count';
CONSOLIDATEWHEN SkipFull
    'SkipFull.Skip.Count & SkipFull.Soil.Count>=SkipFull.Skip.Capacity';

BEFOREEND Dig GENERATE SoilGenerate Soil;
LINK S1 Dig SoilWait;
LINK S2 SoilWait LoadSoil;
LINK S3 LoadSoil SkipFull Soil;
LINK S4 SkipFull Haul Soil;
LINK S5 Haul Empty Soil;
LINK S6 Empty MvdSoil;
LINK C1 CrewWait1 Dig;
LINK C2 Dig CrewWait2;
LINK C3 CrewWait2 LoadSoil;
LINK C4 LoadSoil CrewWait1;
LINK SK1 SkipWait SetSkip;
LINK SK2 SetSkip SkipFull Skip;
LINK Sk3 SkipFull Haul Skip;
LINK SK4 Haul Empty Skip;
LINK Sk19 Empty RouteSkip1;
LINK SK5 RouteSkip1 PushExcavation;
LINK SK6 PushExcavation SkipWait;
LINK SK7 RouteSkip1 LoadRings;
BEFOREEND Haul ASSIGN RoundAWeight RoundAWeight+1;
STRENGTH SK5 RoundAWeight<StrengthOfSK5;
STRENGTH SK7 RoundAWeight>=StrengthOfSK5;
BEFOREEND LoadRings ASSIGN RoundAWeight ZeroRounds;
BEFOREEND LoadRings ASSIGN RoundBWeight RoundBWeight+1;
LINK SK8 LoadRings PushConcreteLining Skip;
LINK SK9 PushConcreteLining PlaceRings Skip;
LINK SK10 PlaceRings GroutRings Skip;
LINK SK11 GroutRings ReturnConcreteLining Skip;
LINK SK12 ReturnConcreteLining RouteSkip2;
LINK SK13 RouteSkip2 SkipWait;
BEFOREEND ReturnLightTrack GENERATE TunnelLengthOfThree Meter;
LINK SK14 RouteSkip2 LoadLightTrack;
LINK SK15 LoadLightTrack PushLightTrack Skip;
LINK SK16 PushLightTrack PositionLightTrack Skip;
LINK SK17 PositionLightTrack ReturnLightTrack Skip;
LINK SK18 ReturnLightTrack SkipWait;
STRENGTH SK13 RoundBWeight<StrengthOfSK5;
STRENGTH SK14 RoundBWeight>=StrengthOfSK5;
BEFOREEND LoadLightTrack ASSIGN RoundBWeight ZeroRounds;
BEFOREEND ReturnLightTrack ASSIGN MeterOfTunnel MeterOfTunnel+1;
BEFOREEND ReturnLightTrack ASSIGN ExpectedDigTime ExpectedDigTime+0.05*ExpectedDigTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedLoadSoilTime ExpectedLoadSoilTime+0.05*ExpectedLoadSoilTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedHaulTime ExpectedHaulTime+0.05*ExpectedHaulTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedEmptyTime ExpectedEmptyTime+0.05*ExpectedEmptyTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedPushExcavationTime ExpectedPushExcavationTime+0.05*ExpectedPushExcavationTime;
```

```
BEFOREEND ReturnLightTrack ASSIGN ExpectedSetSkipTime ExpectedSetSkipTime+0.05*ExpectedSetSkipTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedLoadRingsTime ExpectedLoadRingsTime+0.05*ExpectedLoadRingsTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedPushConcreteLiningTime ExpectedPushConcreteLiningTime+0.05*ExpectedPushConcreteLiningTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedPlaceRingsTime ExpectedPlaceRingsTime+0.05*ExpectedPlaceRingsTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedGroutRingsTime ExpectedGroutRingsTime+0.05*ExpectedGroutRingsTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedReturnConcreteLiningTime ExpectedReturnConcreteLiningTime+0.05*ExpectedReturnConcreteLiningTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedLoadLightTrackTime ExpectedLoadLightTrackTime+0.05*ExpectedLoadLightTrackTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedPushLightTrackTime ExpectedPushLightTrackTime+0.05*ExpectedPushLightTrackTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedPositionLightTrackTime ExpectedPositionLightTrackTime+0.05*ExpectedPositionLightTrackTime;
BEFOREEND ReturnLightTrack ASSIGN ExpectedReturnLightTrackTime ExpectedReturnLightTrackTime+0.05*ExpectedReturnLightTrackTime;
LINK M1 ReturnLightTrack MeterCount;
VARIABLE HoursSimulated SimTime/60;
DURATION Dig ExpectedDigTime;
DURATION LoadSoil ExpectedLoadSoilTime;
DURATION Haul ExpectedHaulTime;
DURATION Empty ExpectedEmptyTime;
DURATION PushExcavation ExpectedPushExcavationTime;
DURATION SetSkip ExpectedSetSkipTime;
DURATION LoadRings ExpectedLoadRingsTime;
DURATION PushConcreteLining ExpectedPushConcreteLiningTime;
DURATION PlaceRings ExpectedPlaceRingsTime;
DURATION GroutRings ExpectedGroutRingsTime;
DURATION ReturnConcreteLining ExpectedReturnConcreteLiningTime;
DURATION LoadLightTrack ExpectedLoadLightTrackTime;
DURATION PushLightTrack ExpectedPushLightTrackTime;
DURATION PositionLightTrack ExpectedPositionLightTrackTime;
DURATION ReturnLightTrack ExpectedReturnLightTrackTime;
ONEND ReturnLightTrack PRINT StdOutput
    "The Current Simulation Time ONEND ReturnLightTrack is %.2f Hrs\n" SimTime/60;
INIT CrewWait1 NumberOfCrews;
INIT SkipWait NumberOfSkip SkipA;
SIMULATEUNTIL 'MeterCount.CurCount>=FinTunnelLength';
DISPLAY "
*******************************************************************************************************************
                Selected result from experiment
";
DISPLAY "Time required to move soil : "
   HoursSimulated
   "   Hours";
```

Stroboscope Result from Running the Model:

```
Stroboscope Model Tunneling considering Productivity (1723792672)

Number of skips:1
Number of Crews:1
Length of Tunel:9



The Current Simulation Time ONEND ReturnLightTrack is 4.43 Hrs
The Current Simulation Time ONEND ReturnLightTrack is 9.09 Hrs
The Current Simulation Time ONEND ReturnLightTrack is 13.98 Hrs

******************************************************************
                Selected result from experiment

Time required to move soil : 13.976083   Hours

-------------------------------------
Execution    Time = 0.047 seconds
```

APPENDIX F

STROBOSCOPE MODELING LANGUAGE CONCEPTS

Link

	Statement	Arguments
1.	LINK	Link Predecessor Successor [ResourceType];
2.	ONFLOW	Link ActionTarget TargetArguments;
3.	ONRELEASE	ReleaseLink {ActionTarget | SaveProp} [TargetArgument][…];
4.	ASMBASELINK	Link Predecessor Assembler;
5.	DISASMBASELINK	Link DisAssembler Successor;
6.	DRAWAMT	OutOfQueueGenLinkExpression;
7.	DRAWDUR	OutOfQueueLinkExpression;
8.	DRAWUNTIL	OutOfQueueLinkExpression;
9.	DRAWORDER	OutOfQueueCharLink CursoredExpression;
10.	DRAWWHERE	OutOfQueueCharLink CursoredExpression;
11.	ONDRAW	OutOfQueueLink {ActionTarget | SaveProp} [TargetArgument] […];
12.	ONENTRY	Queue {ActionTarget | SaveProp} [TargetArgument] […];
13.	DUALBASELINK	LinkDisAssembler Assembler;
14.	ENOUGH	OutOfQueueLink BooleanExpression;
15.	RELEASEAMT	GenReleaseLink Expression;
16.	RELEASEUNTIL	CharReleaseLink BooleanExpression;
17.	RELEASEORDER	CharReleaseLink CursoredExpression;
18.	RELEASEWHERE	CharReleaseLink CursoredExpression;

19. REVORDER CharLinkOutOfQueue|CharReleaseLink [LogicalExp];
20. STRENGTH OutOfForkLink Expression;
21. ONASSEMBLY Assembler {ActionTarget | SaveProp} [TargetArgument][…];
22. ONDISASSEMBLY Disassembler {ActionTarget | SaveProp} [TargetArgument][…];

Resources

1. GENTYPE GenType;
2. ASSEMBLER Assembler CompCharTypeAssembled;
3. CHARTYPE Char Type [Property] […];
4. SAVEPROPS CharType Property [Property][…];
5. SUBTYPE CharType SubType [Expression] [..];
6. VARPROP CharType Property AnonymouslyCursoredExpression;
7. FILTER Filter CharType CursoredExpression;
8. FILTEREXP Filter CursoredExpression;
9. COMPTYPE CompCharType;
10. DISASSEMBLER Disassembler CompoundCharTypeDisassembled;
11. DYNAFORK ForkResourceType [Stream];
12. FORK ForkResourceType [Stream];

ACTIVITY

1. DURATION Activity Expression;
2. ONEND Activity ActionTarget TargetArguments;
3. ONSTART Activity ActionTarget [TargetArguments][…];
4. BEFOREEND Activity ActionTarget TargetArguments;

Normal

1. NORMAL Normal;

Combi

2. COMBI Combi;
3. PRIORITY Combi Expression;
4. NOENOUGHS Combi;
5. BEFOREDRAWS Combi Action Target Target Arguments
6. SEMAPHORE Combi BooleanExpression;

Queue

1. QUEUE Queue ResourceType;
2. DISCIPLINE CharQueue CursoredExpression;
3. BINQUEUE Binned Queue ResourceType NumberofBins TopOfFirst BotOfLast;
4. INIT SimpleCharQueue PositiveIntExpression SubType;
5. INIT CompoundCharQueue PositiveIntExpression;
6. INIT GenQueue PositiveFloatExpression;

Consolidator

1. CONSOLIDATOR Consolidator;
2. CONSOLIDATEWHEN Consolidator BooleanExpression;

Action Targets:
Value Storage:

1. SAVEVALUE Variable[*] Expression;
2. ARRAY MatrixName Rows Columns [{ InitValue InitValue…}];
3. ARRAY ArrayName Size [{ InitValue InitValue…}];
4. COLLECTOR Collector[*];
5. COLLECT Action Target [Target Arguments][…];
6. MVAVGCOLLECTOR MvAvgCollector[*] MaxSamplesExpression;
7. WGTCOLLECTOR WgtCollector[*];

8. TMWGTCOLLECTOR TmWgtSaveValue[*] Expression;
9. BINCOLLECTOR Binnned collector [*] NumberofBins TopOfFirst BottomOfLast;

OutFiles:

1. APPFILE Alias DiskFileName;
2. OUTFILE Alias DiskFileName;
3. REPORT [Outfile];

Random Number:
The random number stream:
1.the Default Stream

SEED Expression;

2.Additional Srreams:

1. STREAMS NumberOfRandomNumberStreams;
2. SEEDALL Expression [SeperationInHundredThousands];
3. SEEDN Stream Expression;

3.Antithetic Sampling:

1. ANTITHETICS BooleanExpThatTurnsOnOrOffAntitheticSampling];

- **Flow Control Statement:**

2. ENDMODEL;

- **IF Blocks:**

3. IF IfExpression;
4. ENDIF;
5. ELSE;
6. ELSEIF; IfExpression;
7. While-Wend Blocks
8. WHILE;
9. WEND;
10. CONTINUE;
11. BREAK;

Persistent Save Values and Collectors:

1. CLEAR;
2. RESETSTATS;

CONTROLSTATEMENT:

1. SIMULATE;
2. INIT GenQueue PositiveFloatExpression;
3. REPORT [Outfile];
4. DISPLAY [QuotedString|SingleQuotedExpression][…];

ATTRIBUTESTATEMENT:

1. DURATION Activity Expression;

MORE

2. DEBUGOFF;
3. DEBUGON;
4. FUNCTION NameInStrobo DllName NameInDll nArguments [CONSTANT];
5. LOADADDON DllName;
6. SILENTREPLICATE [BooleanExpThatTurnsOnOrOffSilentReplications];
7. SIMULATEUNTIL BooleanExpression;
8. STATEMENT Alias DllName NameInDll;
9. VARIABLE Variable Expression;

LIST OF REFERENCES

Anil Sawhney, A., (1998). Construction Project Simulation Using CYCLONE. NRC Canada.

Chen Po-Han, Weng Haijie. (2008). A Two-Phase GA Model For Resource Constrained Project Scheduling. Automation in Construction.

Flood, I. (2009). "A Hierarchical Construction-based Approach To Modeling Construction and Manufacturing Processes." *2009 Winter Simulation Conference*, M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds., Austin, TX, 1.

Flood, I. (2010a). "Foresight: A Structured Graphical Approach to Constraint-Based Project Planning." *2nd International Conference on Advances in System Simulation, Simul 2010,* IEEE, CSDL, Nice, France, August 2010, 6 pp.

Flood. I. (2010b). "Foresight Versus Simulation: Construction Planning Using Graphical Constraint-Based Modeling." *2010 Winter Simulation Confrence*.

Howell, G. (1999). "What is Lean Construction-1999." *Proc., 7th Ann, Conf. Intl. Group for Lean Constr.,* IGLC-7, Univ. of California, Berkeley, CA, July 26-28.

Galloway, P.D. (2006). "Comparative Study of University Courses on Critical-Path Method Scheduling."*ASCE*, 132(7), 712.

Hinze, J. W. (2004). *Construction Planning and Scheduling*. Pearson Education, Inc., Upper Saddle River, NJ.

Issa, R.A., I. Flood, and W. O'Brien, (Eds.). 2003. *4D CAD and Visualization in Construction: Developments and Applications*. Steenwijk, Netherlands: A. A. Balkema.

Kallantzis, A., Soldatos, J., and Lambropoulos, S. (2007). "Linear versus Network Scheduling: A Critical Path Comparison." *ASCE,* 133(7), 483.

Kim, K., and de la Garza, Jesus M. (2005). "Critical Path Method with Multiple Calendars." *ASCE*, 131(3), 330.

Koo, B., and M.Fischer. 2000. Feasibility Study of 4D CAD in Commercial Construction. *Journal of Construction Engineering and Management* 126(4):251-260.

Yamin, R. A., and Harmelink, D. J. (2001). "Comparison of Linear Scheduling Model (LSM) and Critical Path Method (CPM)." *ASCE*, 127(5), 374.

Julio Cesar Martinez. (1996). STROBOSCOPE. A Dissertation Submitted in Partial Fulfillment of the Requirements For the Degree of Doctor of Philosophy in the University of Michigan.

Sang Hyun Lee, Feniosky Peña-Mora and Moonseo Park, **(**2006). Dynamic Planning and Control Methodology for Strategic and Operational Construction Project Management. Automation in Construction.

Courtney K. Zurich, (2010). A Comparison of CPM-Bar Chart and a Hierarchical Graphical Approach to Planning Construction Processes, Thesis submitted to the University of Florida in partial fulfillment for the degree of MS, University of Florida

BIOGRAPHICAL SKETCH

Valeh Nowrouzian earned a Bachelor of Architectural Engineering from the Azad Tehran Markaz University, Tehran, Iran, in 2007. In July 2007, Valeh came to United State of America. She worked in an architectural office in Las Vegas for couple of months. She decided to continue her education. She applied to the M.E. Rinker, Sr. School of Building Construction to pursue a Master of Science in Building Construction in a continued effort to peruse a career in project management. She published her work on Foresight modeling at the 14th International Conference on Computing in Civil and Building Engineering in Moscow, Russia.