University of Alberta

Linear Scheduling: Special Purpose Simulation Template Developed for Simphony.Net

by

Veronica Haring

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Construction Engineering & Management

Department of Civil & Environmental Engineering

Spring 2014
Edmonton, Alberta

ABSTRACT

The need for specialized software, particularly for repetitive construction is growing. Although simulation has been accepted in the research world, the industry has been slow to follow. The goal of this research is the development of a special purpose template for linear scheduling that is simple to use and requires little to know knowledge of simulation. In order to develop this template, a thorough review of the development of the linear scheduling method and existing linear scheduling software is conducted. The major disadvantage of almost all of the existing programs is their inability to incorporate uncertainty. The special purpose template developed not only incorporates uncertainty but also models time and space dependency, material deliveries, milestones, continuous and discrete activities and varying production rates. The template is verified using a fictional case study and the applications of the template are shown through both a fictional and real world case study.

ACKNOWLEDGMENT

I would like to thank my supervisor Dr. Simaan AbouRizk for his support and guidance throughout both the course based and thesis based portions of my program. I would also like to thank Brenda for all the behind the scenes support and Amy for answering the constant stream of citation and formatting questions.

I am eternally grateful for Steve and Dylan’s patience with my programming skills, for adapting every time I came up with a new idea for the template and allowing me to push the limits of Simphony. Without them, this research would not have happened.

I would like to express my gratitude to Maria and Ledcor for providing a real world case study for this research. I would like to further thank Ledcor for the opportunity to spend time on site.

To my family; Mom for always encouraging me and helping me study, even after they added letters to math; Dad for teasing me by telling me girls don’t do math, knowing that it would make me want to work harder; Tina for constantly reminding me that I made my choices; and Sara for relentlessly asking me how many pages of my thesis I had written.

Finally I would like to thank my fiancé Jarret for his unconditional love, even during the stress of midterms and finals. His humour and music kept me grounded and without his support I would not be where I am now.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ALISS – Advanced Linear Scheduling System

CAD – Computer Aided Design

CAP – Controlling Activity Path

CPM – Critical Path Method

CYCLONE – Cyclic Operations Network

ENR – Engineering News Record

GPSS – General Purpose Simulation System

LCPM – Linear Construction Project Manager

LOB – Line of Balance

LOS – Lean Oil Sand

LSM – Linear Scheduling Method

LSM_{VPR} – Linear Scheduling Model with Varying Production Rates

MLS – Multistage Linear Scheduling

MSE – Mechanically Stabilized Earth

NVR – Network Video Recorder

PoE – Power over Ethernet

PULSS - Purdue University Linear Scheduling Software

REPCON – Representing Construction

RPM – Repetitive Project Model

RSM – Repetitive Scheduling Method

SCaRC - Space-Constrained and Resource-Constrained

SIREN – Simulation of Repetitive Networks

SPS – Special Purpose Simulation

SYRUS – System for Repetitive Unit Scheduling

VPM – Vertical Production Method

1.0 INTRODUCTION

1.1 Background

A survey conducted in 1983 of the top 400 construction contractors by Engineering News Record, found that 87% of respondents indicated that scheduling has the potential to improve productivity (Arditi 1985). Due to the competitive nature of the construction industry, companies are narrowing their focus and becoming specialists in certain types of construction. This specialization has resulted in the need for focused scheduling tools (Yamín and Harmelink 2001). This need is seen through the results of a study done by Tavakoli & Riachi (1990). As a part of the study, questionnaires were sent out to the directors of the ENR's top 400 contractors. The survey was intended to determine the present status of the use of the Critical Path Method (CPM). It was found that 93% of the companies were using CPM, but only 26% used the method on all contracts.

In order to see the productivity improvements scheduling can produce, the correct scheduling tool must be chosen (Naaman 1974; Birrell 1980; Harmelink and Yamin 2001). The project manager should select a scheduling tool after taking into consideration which personnel will be using it, their familiarity with the method and the level of detail required in the schedule (Chrzanowski and Johnston 1986). Network techniques are excellent tools for certain types of construction. However, indiscriminate use of network analysis, on projects of a repetitive nature, by a wide variety of organizations has often resulted in spectacular failure (Arditi and Albulak 1979). Why then, with evidence indicating the contrary, is network analysis still being used to schedule projects of a repetitive nature?

One reason contractors aren't exploring scheduling options is that many government and private clients require the construction company to submit a network schedule (Peer 1974). Contractors don't want to spend the extra time and resources developing a separate schedule for their own use. Another reason could be that contractors demonstrate more willingness to adopt innovations in construction methods than in management, taking the attitude that "We have always done it this way; everyone else does it this way, so why change and who needs more stress?" (Russell 1985). A third reason could be the lack of commercially available software that addresses the needs of the industry (Duffy et al. 2011).

Arditi (1985) explains that the lack of software is an issue of time and that the parties involved in construction have limited resources. Construction companies do not wish to consume resources by actively take part in conducting research into productivity related problems. On the other hand, AbouRizk & Hajjar (1998) argue that the lack of software is an issue of complexity; partially the complexity of the construction process itself and partially the effort required to learn the language required to create a model. This is echoed by Russell & Wong (1993) who state that the barrier to the use of a computerized system for projects with significant repetition is the tedium of describing the projects and the difficulties in testing alternative strategies and maintaining a current schedule despite changes and delays. One solution to this issue of complexity is simulation, specifically special purpose simulation. This option is explored below.

1.2 Construction Simulation: Simphony

According to AbouRizk et al. (2011) construction simulation is the science of developing and experimenting with computer-based representations of construction systems to understand their underlying behaviour. Simulation modeling of construction activities allows the user to test important management decisions while yielding output that can be easily interpreted in terms of management decisions (Ashley 1980). Simulation allows the user to accurately experiment with various approaches for completing a project without having to set foot on site (AbouRizk 2010). AbouRizk (2010) goes on to say that simulation is most effective when:

- Problems are characterized by uncertainty
- Problems are technically or methodically complex
- Repetition is evident
- Flexibility in modeling logic and knowledge is required
- An integrated solution is required
- Detail and accuracy matter

All of the above are true of repetitive construction projects.

Simulation has been widely accepted in the research community, but not yet by industry. A reason for this could be that when modelling, users tend to use a general purpose simulation approach without much thought, which is not efficient in the construction domain (Ekyalimpa et al. 2012). Ekyalimpa et al. (2012) explains that constructing an effective and meaningful model demands creativity and an in depth knowledge of simulation. As a result, building models of operations using general purpose simulation requires sufficient simulation expertise and a lot of time and effort.

Special purpose simulation on the other hand, is a computer based environment built to enable someone who is knowledgeable in a given domain, but not necessarily simulation, to model a project (AbouRizk and Hajjar 1998). The project is modelled within that domain in a manner where symbolic representations, navigation schemes within the framework, creation of model specifications, and reporting are completed in a format native to the domain itself. In other words knowledge of simulation is not required, only knowledge of the construction process itself. Furthermore, special purpose simulation tools find a balance between the need for accurate modeling and the desire for reduced complexity (AbouRizk and Hajjar 1998).

Simphony (AbouRizk and Mohamed 2000) is a Microsoft Windows based computer system that allows for the creation of special purpose simulation tools. It will be utilized as a part of this research to create a special purpose template for linear scheduling.

1.3 Research Objectives

The primary objective of this research is to identify a framework for scheduling repetitive projects. This framework will be used to develop a simulation tool that addresses the needs of the industry while remaining simple to use. These objectives will be achieved by the following methods:

1. Reviewing current repetitive scheduling literature to obtain a firm grasp on the various methods available as well as the advantages and disadvantages associated with these methods.
2. Reviewing current repetitive scheduling programs and software to identify their strengths and weaknesses.
3. Using the information obtained, a suitable framework for scheduling repetitive projects will be developed. This framework will serve as the algorithm for the development of a repetitive scheduling tool.
4. The developed tool will be verified using a text book case study
5. The applications of the tool will be shown through a real world case study

1.4 Thesis Organization

Chapter 1 introduces the problem, provides background information and outlines the objectives and proposed methods for the completion of this research.

Chapter 2 provides a literature review of the linear scheduling method. Its origins as the line of balance method are explained and its progression to the method it is today is shown. The chapter then presents a comparison of the linear scheduling method to other scheduling methods and explores the incorporation of uncertainty in linear scheduling.

Chapter 3 reviews current linear scheduling simulation programs. The various programs are compared and their strengths and weaknesses are highlighted.

Chapter 4 summarizes the requirements of a repetitive scheduling framework, presents the algorithm developed from the requirements and introduces Simphony as the tool for implementing the algorithm. The special purpose template developed for Simphony is explained, the modeling elements that make up the template are presented and the error detection methods are described.

Chapter 5 uses an example problem from literature to verify the results of the template and to show its applications. The problem is explained, the models are shown, the results are discussed and further applications are presented.

Chapter 6 applies the template to a real life case study. The case study is introduced, the data collection methods are discussed, the models are presented and the results are analyzed.

Chapter 7 provides a summary of this research and recommends future areas of research and improvement for the template.

2.0 LITERATURE REVIEW

The origins of linear scheduling are unclear and because of this, it is believed that the method may have multiple origins (Johnston 1981). Wherever linear scheduling began, many techniques, with the common goal of scheduling repetitive work, have been developed under various names. These techniques can be divided into two groups: line of balance (LOB) and linear scheduling methods (LSM) (Vorster and Bafna 1992). Although the focus of this research is on linear scheduling methods, a brief introduction to line of balance techniques will be provided to illustrate how linear scheduling has developed and to identify areas of further research. Section 2.1 provides an introduction to the line of balance method. Section 2.2 introduces the various methods referred to as linear scheduling methods. Section 2.3 compares linear scheduling methods to other scheduling methods. Section 2.4 outlines the research done into incorporating uncertainty into linear scheduling methods.

2.1 Line of Balance (LOB)

The line of balance technique originated around the time of World War II (Carr and Meyer 1974; Al Sarraj 1990), and was initially used in manufacturing to maintain and evaluate the flow of products through a production line. It was not fully exploited by the construction industry, despite its advantages for repetitive units (Carr and Meyer 1974), and faded into obscurity upon the advent of network diagrams (Arditi and Albulak 1986).

The line of balance technique requires three inputs (Carr and Meyer 1974):

1. A unit network showing activity interdependencies and time required between activity and unit completion;
2. An objective chart showing cumulative calendar schedule of unit completion; and
3. A progress chart showing the completion of the activities for each unit

The unit network is also referred to as the program chart and is the basic unit of the line of balance technique (Halpin and Riggs 1992). Its purpose is to define the operations to be performed, the sequence of the operations and the processing and assembly lead times. The program chart is similar to the critical path method (CPM) network for one unit of the repetitive construction process; i.e. a typical floor in a multistory building. An example of a program chart can be seen in Figure 2-1(a). In this example, the activities are indicated as squares and the lead time is indicated by the time scale “Months prior to shipment”.

The objective chart shows the time it will take to complete the units in the project with time on the x-axis and the number of units on the y-axis. An example of an objective chart can be seen in Figure 2-1(b).

The progress chart shows the actual progress of the project. The progress of each activity is measured in the field and then plotted on the progress chart. The number of units in the project is shown on the y-axis, with the same scale as the objective chart, and the activities are shown on the x-axis. An example of a progress chart can be seen in Figure 2-1(c).

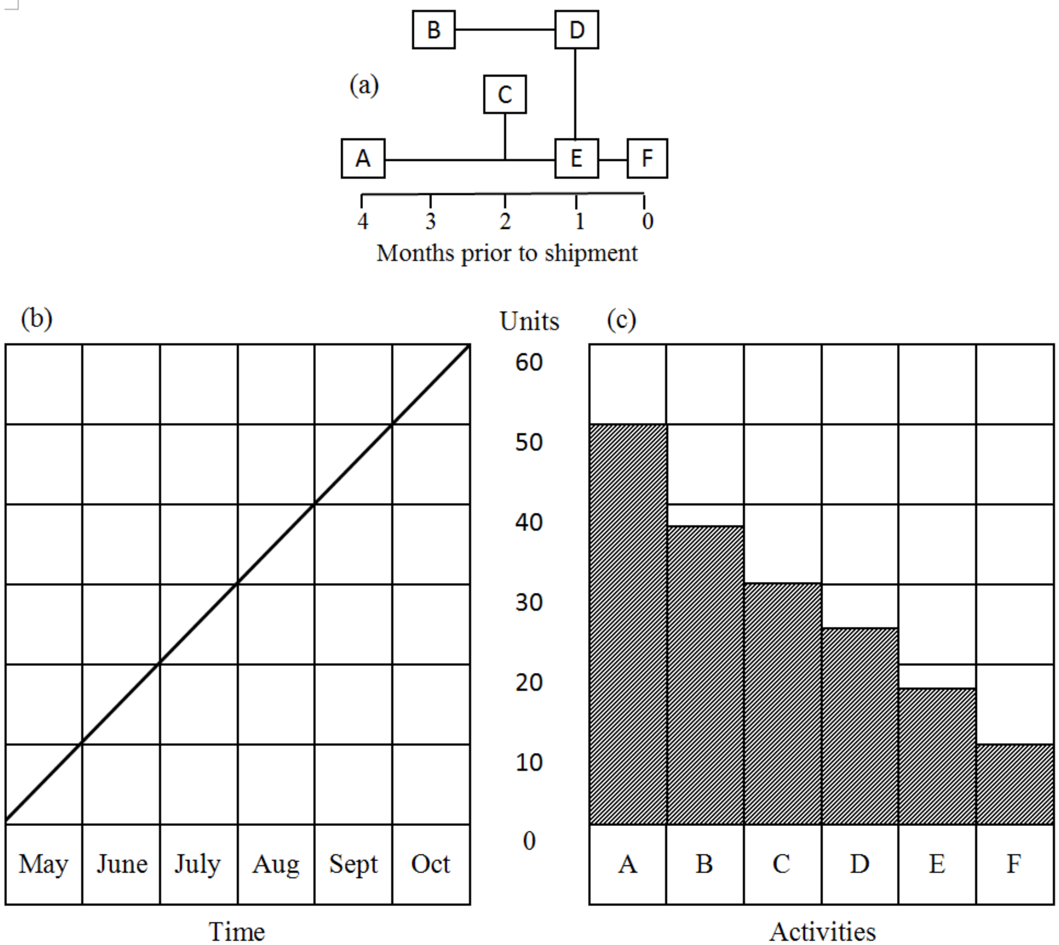


**Figure 2-1 Line of Balance Input Charts (adapted from (Mattila and Abraham 1998))
(a) Program Chart; (b) Objective Chart; (c) Progress Chart**

From these three inputs, the LOB can be drawn. Consider activity B, which must be completed 3 months prior to shipment. If the date under consideration is June 1st as indicated in Figure 2-2, three months from then would be September 1st. A line is drawn up from September 1st to the line on the objective chart and then horizontally to activity B on the progress chart. To stay on schedule, 40 units should have started activity B by June 1st. The progress chart in Figure 2-2 indicates that less than 40 units have been started and therefore activity B is behind schedule. Corrective action needs to be taken to ensure that activity B does not impede the progress of the remaining units. In Figure 2-2, activities A, C and

F are on schedule, activities B and E are behind schedule and activity D is ahead of schedule.

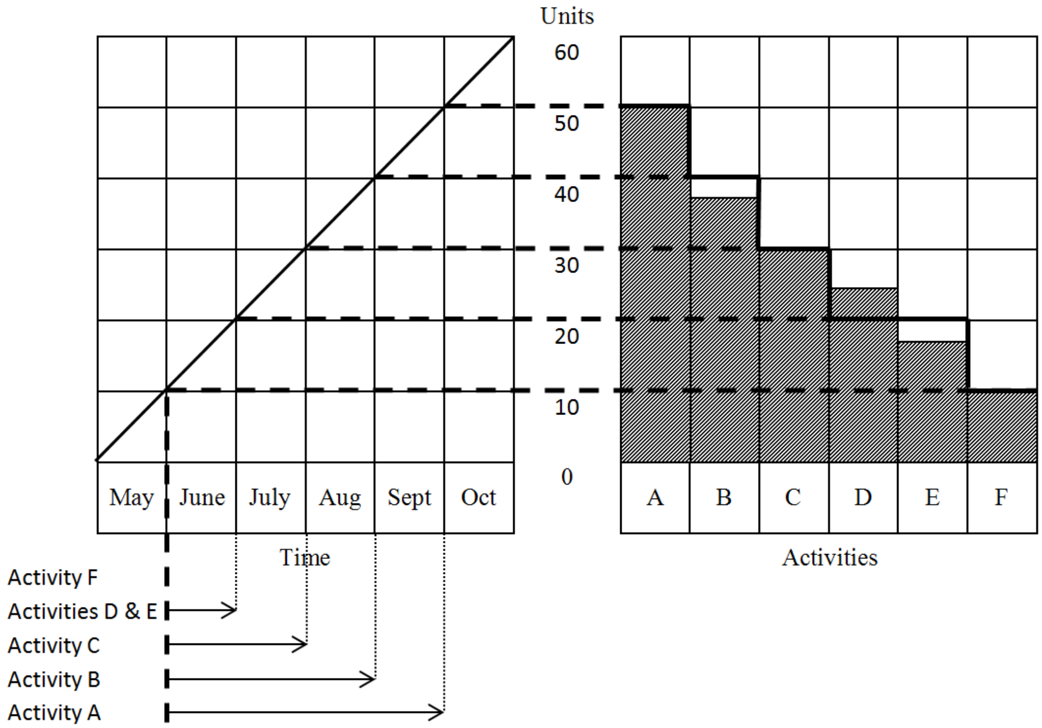


Figure 2-2 Line of Balance (adapted from (Mattila and Abraham 1998))

2.2 Linear Scheduling Methods (LSM)

The linear scheduling method was developed partially from the LOB technique and as such, many authors refer to linear scheduling methods as line of balance. When comparing the LOB technique to the LSM, the difference is simply in emphasis (Johnston 1981; Harmelink 1995). The LOB technique emphasizes the progress chart and balance line, where the LSM emphasizes the time space graph, which is similar to the LOB's objective chart. The LSM essentially expands on the objective chart by including all activities as their own individual lines (continuous activities). The LSM can also account for activities that do not occur on every unit, or along the entire project (discrete activities).

Since the early 1970's, several variations of linear scheduling methods have been developed under a variety of different names. A chronological history of these methods is presented below.

The construction planning method or technique described by Peer (1974) is an early example of linear scheduling. Peer states that this method can be divided into the following steps or objectives:

1. To break down the project into constituent component processes
2. To divide realization of these processes between adequate production crews
3. To define technological connections between them and activity categories
4. To decide which production line should dictate the progress pace of the project, due to economic considerations or resource limits
5. To make an approximate estimate of the resulting construction time and decide how many production units should be employed in parallel
6. To balance the progress of non-critical production lines with that of the chosen critical one, aiming at achieving work continuity
7. To check the possibility, within practical limits, of shortening construction time by the introduction of planned breaks in continuity or changes in crew size

8. To analyze the whole process in terms of time and activity situations and produce a plan

In the article, Peer speaks to the need for a better scheduling method for repetitive projects and that the development of construction planning procedures other than networks is essential.

The vertical production method (VPM) introduced by O'Brien (1975) is a method specific to high rise construction. O'Brien explains that high rise construction is a hybrid project, requiring a hybrid schedule. All of the mobilization, prep work and construction of the foundation should utilize a network diagram. The repetitive floors of the building should utilize the VPM due to the repetitive nature of the work. O'Brien mentions that it would be possible to computerize the VPM, but does not provide details on how that would be achieved.

Selinger (1980) introduced a method whose two main requirements were to: (1) permit work continuity requirements and (2) determine resource quantities as to minimize project duration. Unlike previous methods, Selinger does not focus on activity durations, but on the importance of continuity in preventing idle time.

Birrell (1980) introduced the matrix construction plan. Work squads made up of resources that do the same activities repeatedly are scheduled in an effort to maintain work continuity. A matrix is produced with time phases on one axis and work locations one the other. The work locations are in the order that work squads will pass through them. Although this method allows for easy cost and schedule control (Birrell 1980), it makes many assumptions that impede its usefulness in the field (Kavanagh 1985).

Johnston (1981) makes the first attempt to pull together the various methods under one name; the linear scheduling method. The article explains the basic elements of the linear scheduling method, particularly, axis parameters, activity production rates, activity interruption, buffers and discrete activities. These elements are shown in Figure 2-3.

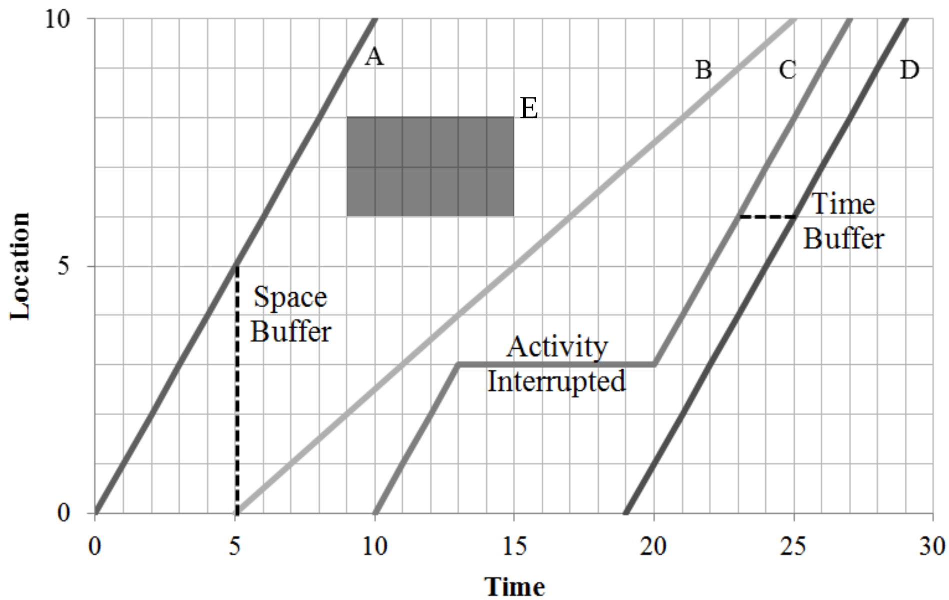


Figure 2-3 Basic Elements of the Linear Scheduling Method

The axis parameters are always time vs. location, with time typically in the units of days. The units of location depend on the type of project and can be anything from floors in a high rise building to meters in a pipeline project. Activities are represented as individual lines and the slope of each line represents the production rate of that activity. For example, in Figure 2-3 the production rate of activity A is 1 unit of location/day. Activity interruption is represented by a horizontally straight segment of line as shown in activity C in Figure 2-3. Buffers are either a required distance or time interval between activities (Johnston 1981). Figure 2-3 shows both a time buffer of 2 time units between activities C and D and a space buffer of 5 location units between activities A and B. Discrete tasks are those that have little to no repetition during a project; i.e. a culvert on a highway project. These tasks are represented as a block on the linear schedule. In Figure 2-3, activity E is a discrete task that occupies locations 7 and 8 for 6 time units. Johnston (1981) explains that these discrete portions of the project are best scheduled by other methods. Once the duration is determined through network analysis it can be added to the linear schedule.

The time spaced scheduling method introduced by Stradal & Cacha (1982) produces a schedule with time on the horizontal axis and units on the vertical axis. The article presents a multitude of projects (pipeline, apartment, multistory building, road section and railway bridge) all scheduled by this method.

Chrzanowski & Johnston (1986) apply the linear scheduling method to an actual road construction project that had previously been scheduled using CPM. In the schedule, time was plotted on the vertical axis and stations along the horizontal axis. This is opposite of what Johnston (1981) explains as axis parameters, but Chrzanowski & Johnston (1986) argue that axis assignment should be up to the scheduler. The schedule created also includes discrete activities that are linked to separate CPM networks.

Arditi & Albulak (1986) refer to their method as line of balance; however, their method emphasizes the time space graph and is therefore a linear scheduling method. The article provides a sample schedule for a road pavement operation. The first attempt at a schedule had too few activities making it far too simplistic. The second attempt shows more activities, but is difficult to read. The article concludes that the degree of detail required for a liner schedule must be evaluated carefully. Too many activities and the schedule becomes a jungle of lines. Too few activities and the schedule will be of little practical use. Choosing an appropriate scale to effectively communicate the information contained in the schedule is critical.

One of the first mentions of making linear scheduling a computer-based tool is presented by Russell & Caselton (1988). Russell & Caselton developed a two-state variable, N-stage dynamic programming formulation with a methodology to extract sensitivity information which permits near optimal alternative solutions. The method developed accounts for the reality of repetitive construction and includes precedence relationships and the ability to address work continuity constraints. The authors intend their research to be directed towards the development of a practical computer based tool for repetitive projects.

Reda (1990) states that all graphical approaches to scheduling (i.e. linear scheduling methods) attempt only to finish the project as early as possible with no care for the impact on the direct cost. Reda developed the repetitive project model (RPM), which includes a network technique, a graphical technique and an analytical technique to address this issue. The objective of the RPM is to minimize the project direct cost while satisfying the following (Reda 1990):

1. Maintain a constant production rate
2. Maintain crew work continuity
3. Allow for time buffers
4. Allow for stage buffers
5. Specify a feasible project duration

The first objective is met by setting a constant duration for the same activity at all stages. This does not reflect real world situations however, as production rates can and often vary from location to location in a project (i.e. clearing and grubbing the length of a highway project).

Lutz (1990) integrates linear scheduling and simulation to determine the linearity of production lines. The MicroCYCLONE program (Halpin and Riggs 1992) was utilized as the simulation tool and the data obtained was used to plot the linear schedules. Lutz states that the process should be further computerized to automatically produce the schedules.

Moselhi & El-Rayes (1993) developed a dynamic programming model for generating optimized schedules that incorporates cost as an important variable in the decision making process. For each activity in the project, the model assists the planner in choosing the optimum crew formation from a set of alternatives.

Senouci & Eldin (1996) introduce a dynamic programming formulation for scheduling linear projects with non-sequential activities. It is noted in the paper that programs developed to computerize linear scheduling to date are unable to address variable production rates. The formulation performs cost time trade off analysis, addresses multiple production rates and handles both continuous and discrete activities. The authors conclude by stating that the approach presented provides a building block for the formulation of a more comprehensive computerized system.

Wang & Huang (1998) introduce the multistage linear scheduling (MLS) method which is applicable when scheduling repetitive projects with interval time limitations. An example of this type of project would be a diaphragm wall, where excessive idle time between activities could lead to collapse. The article schedules a project using both the LOB technique and the MLS method and compares the outcomes.

Harris and Ioannou (1998) argue that previous presentations of repetitive scheduling methods have been more complicated than necessary and that the various names for similar methods could be a reason for their limited acceptance. Their purpose is to integrate the methods into one that ensures work continuity and is applicable to both vertical and horizontal construction. This method is called the repetitive scheduling method (RSM) and includes the concept of controlling points, which allow for the rotation of activity lines to achieve continuous resource utilization. The controlling sequence, similar to the critical path of the critical path method (CPM), passes through all of the control points. The control points are the locations at which the controlling sequence switches from production line to production line.

Harmelink and Rowings (1998) aim to provide a level of analytical capability on par with the critical path method, to the linear scheduling method. In this method, named the linear scheduling model, seven activity types are defined. These activity types are as follows and can be seen in Figure 2-4.

1. Continuous full-span linear
2. Intermittent full-span linear
3. Continuous partial-span linear
4. Intermittent partial-span linear
5. Full-span block
6. Partial-span block
7. Bar

The purpose of the method is to define the controlling activity path (CAP), which is the continuous path of longest duration through the project. Activities on the CAP must be completed as planned in order to finish the project within the overall planned duration. The CAP is found by identifying all possible logical sequences through the activities on the linear schedule, performing the upwards pass and then performing the downwards pass. The article goes on to explain how the upward and downward passes are performed for the seven types of activities through various examples of smaller projects. For larger projects, completing this process manually would be time consuming. The article concludes by stating that this method provides the foundation for a computer-based linear scheduling algorithm. For a more information on the determination of the controlling activity path, refer to Harmelink (1995)

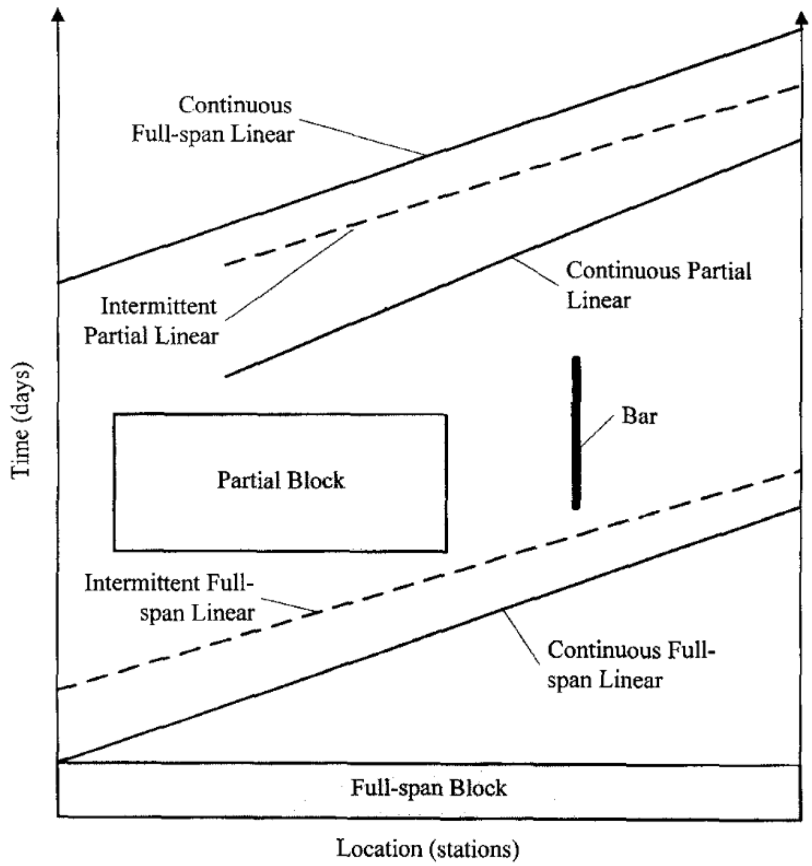


Figure 2-4 Activity Types (Source: (Harmelink and Rowings 1998))

Mattila and Park (2003) compare Harris and Ioannou (1998) repetitive scheduling method and Harmelink and Rowings (1998) linear scheduling model for finding the controlling path of a linear schedule. It was found that both methods identify the same controlling path, however additional work on more complicated schedules is recommended. In addition, it is recommended that the techniques must be computerized in order to gain acceptance by the industry.

Liu et al. (2005) explains that linear construction projects can be either discrete or continuous. A discrete project is defined as one made up of identical units such as houses or high rise floors. A continuous project has no clearly defined units and is characterized by its geometric layout, such as a highway, pipeline or tunnel. The paper explains that currently, almost all linear simulation is based on discrete projects and cannot support continuous linear projects. An integrated simulation-GA (genetic algorithm) approach is suggested as a solution. For this integrated system, activities are separated into three categories; non-repetitive discrete activities that occur only once (i.e. mobilization, surveying and construction of a temporary road), repetitive discrete activities that occupy a considerable work area (i.e. culverts, grading and base laying) and continuous activities (i.e. excavation, clearing and grubbing). The system outlined in the paper is under development and has not yet been implemented.

Ipsilandis (2007) explains that construction planning is often more complicated than just optimizing cost or schedule and introduces a multiobjective linear programming model for scheduling linear repetitive projects (MOLPS-LRP). This model uses an algorithm that allows for various optimizations of the linear schedule and allows the user to consider schedules besides those that minimize duration or maximize work continuity. Ipsilandis concludes by stating that a fully integrated software implementation of this approach will enhance the applicability of the method to real world projects.

2.3 Comparison of Linear Scheduling Methods and Network Methods

As the linear scheduling method developed, much research was done into the usefulness of various scheduling methods for repetitive projects. There is an overwhelming amount of literature outlining the advantages of the linear scheduling method and disadvantages of other scheduling methods for repetitive construction projects. A summary of the key points is provided below.

Bar charts are the most common tool for scheduling construction projects due to their simplicity. However, bar charts do not show the interdependencies of activities and because of this, there is no way of knowing how changes will affect the project as a whole (Johnston 1981; Stradal and Cacha 1982; Chrzanowski and Johnston 1986; Arditi et al. 2002). Another disadvantage is that both bar charts and network methods (i.e. CPM) cannot show productivity changes throughout an activity (Johnston 1981; Stradal and Cacha 1982). This can be frustrating for managers as they can see the block of time that an activity requires, but have no way of knowing what is going on within that block of time (Michael Freeman, personal communication, June 27, 2013). This production rate imbalance can cause work stoppages, inefficient resource utilization and excessive cost (Arditi et al. 2002). The linear scheduling method provides more information on the planned method of construction than a bar chart (Johnston 1981).

The critical path method is another common tool for scheduling construction projects; however, it requires knowledge of the method to be understood. This causes problems on site as many subcontractors and workers are unfamiliar with the method (Kavanagh 1985; Arditi and Albulak 1986; Tavakoli and Riachi 1990). Arditi & Albulak (1979) discovered that the network diagram they produced had to be transformed into weekly bar charts in order to be used by site personnel. In contrast, one of the most significant advantages of LSM is its ability to simply display a detailed work schedule that is easily understood by field staff (O'Brien 1975; Johnston 1981; Chrzanowski and Johnston 1986).

Another advantage of LSM over the critical path method is the ability to monitor progress and work continuity. Progress can be shown on a linear schedule, whereas it cannot be shown on a network that is not time scaled (Arditi and Albulak 1986). The entire project is easily visualized with a linear schedule and work continuity is seen on the schedule itself. There is no way of guaranteeing work continuity, or displaying it on a network diagram (Kavanagh 1985; Reda 1990). For a schedule to be useful it must have a positive visual impact and easily associate work activities with project times (Chrzanowski and Johnston 1986).

In terms of preparation and updating, the linear schedule is simpler and faster to prepare/update than a network diagram (O'Brien 1975; Arditi and Albulak 1986). This is the main advantage of the LSM over the CPM. As the repetition on a project increases, the CPM schedule becomes clumsy, full of repetition and much too detailed (Birrell 1980; Johnston 1981; Stradal and Cacha 1982; Chrzanowski and Johnston 1986; Reda 1990). For example, Arditi & Albulak (1986) explain that the road surface treatment project they were scheduling had 58 activities. For 100km of road, there would be 5800 activities in the CPM schedule which would have been costly to produce and extremely time and space consuming.

Yamin & Harmelink (2001) compare the linear scheduling method to the critical path method by scheduling a small bridge and small highway rehabilitation project with both methods. Their results are summarized in Table 2-1 and are in line with the key points found in the research presented.

Table 2-1 Comparison of Scheduling Methods (adapted from (Yamín and Harmelink 2001))

Attribute	CPM	LSM
Aid in reduction of uncertainty/risk	Although CPM schedules use fixed duration for activities, it can be easily complemented by PERT with statistical capabilities. This feature helps planners to get a better idea of time and schedule risks	There is no formal method developed to date that could allow LSM to determine uncertainties in time completion.
Aid in improving production and economical operation	With the incorporation of resource leveling/allocation techniques, CPM schedules can improve the overall completion time and costs by affecting production (add or remove resources). Some limitations have been identified when scheduling continuous projects – difficult to maintain continuity in crew utilization.	Limited capabilities in improving production by changing resources. Easy to schedule continuity on linear projects, improving coordination and productivity.
Aid in achieving better understanding of objectives	In complex projects, CPM network can be very convoluted. This complexity makes them difficult to understand and communicate	LSM is very easy to understand, and it can be used at every level of the construction project.
Accurate calculations	CPM allows the PM to calculate the time it would take to complete a project, and together with the PERT could provide statistical insights to this process. It is difficult to accurately determine and represent space restrictions (if any).	Location/time calculation is easily done. This is the greatest advantage of LSM over CPM when scheduling linear projects. This capability allows PM to accurately plan activities both in time and location
Critical path	It is the main feature of the	The LSM algorithm

	CPM, which can be done very easily.	calculates the controlling activity path (CAP) which is equivalent to the critical path, with the additional features of location criticality.
Ease of use	Extensive computerization has made the CPM easier to use. However, the user needs a considerable amount of training before actually being able to produce valuable information for controlling purposes.	Very intuitive and easy to understand. It can be used at all levels of the company (managers, superintendents and crew). Lack of computerization makes it difficult to use in large and complex projects.
Easy to update	The method could be difficult to update. Once several updates have been done, it becomes difficult to read. Updated schedules are usually out of date when they are finished.	Updating LSM is simple. Linear schedules can be used as as-built documents for claim purposes or for historical productivity databases.

2.4 Uncertainty in Linear Scheduling

The research done into the advantages of the linear scheduling method over other methods for repetitive projects highlighted one major disadvantage – to date, only deterministic approaches to linear scheduling have been taken (Yamín and Harmelink 2001; Trofin 2004). As almost construction projects have a level of uncertainty associated with them, this is a major issue. The following is a chronological look at the limited research done into incorporating uncertainty into linear schedules.

Dressler (1974) makes an attempt to solve this issue by splitting the project into production zones. These production zones are segments of the project that have constant construction conditions such as soil data or surface conditions. The scheduled total construction time is also divided into segments based on significant weather periods. Probability distributions are then assigned to each segment which yields a production field. All activities in the construction process are treated as a single process which results in a schedule similar to the LOB's objective chart. Each production zone can produce a number of production vectors based on the probability distributions. The major downside to this method is that it requires complex linear programming methods to come up with a solution, which proves time consuming on large projects. The article concludes by stating that the method cannot provide optimal solutions on its own, but is a tool that can expose possible consequences of a decision that has associated probabilities.

Crandall (1976) explains that the reason for accepting deterministic time durations is the relationship between probabilistic durations and their associated costs. The desire to unite budget and time control has allowed deterministic analysis to become the norm despite the actual variability of activities on construction projects. Crandall suggests that integrating Monte-Carlo simulation into the scheduling method would offer the best solution.

Ashley (1980) states that the stochastic nature of repetitive-unit construction cannot currently be modelled by linear scheduling methods; however, process interaction simulation could overcome this limitation. Using the general purpose simulation system (GPSS) simulation language, eight sample runs of a project are created; one control case and seven other cases that vary labor availability, resource availability and labor allocation. The downside to this method is that in order to create models the user must understand a simulation language and the method outputs the data to create the schedule rather than creating the schedule itself. The fact that a new program must be written for each network/project limits this methods potential (Kavanagh 1985).

Trofin (2004) undertook a study to assess the impact of uncertainty on the construction process. The goal of the study was to demonstrate that a deterministic approach to linear scheduling is not necessarily the optimal solution in terms of project performance as well as show the impact of uncertainty on important project parameters. The Palisade MonteCarlo@Risk software was used to introduce variance into the activities and Microsoft Excel was used to plot the resulting linear schedules. It was found that the deterministic project duration not only underestimates the actual project duration, but is unlikely to be the optimum schedule in terms of project duration and idle times. Trofin's research emphasizes the need for a stochastic approach to linear scheduling and furthers the idea that software utilizing Monte Carlo simulation may be the answer.

Duffy et al. (2011) introduce a linear scheduling model with varying production rates (LSM_{VPR}) that is along the same lines as the research done by Dressler (1974). This model refers to the entire linear schedule as the time-location chart, made up of working windows. A working window is a time-space rectangle with a homogenous set of variables that affect production. Color can be added to the working windows to indicate activity performance. The color index shows the relationship between the user defined production rate and a production rate based on historical data. However, the color is activity dependent and the same working window could be different colors depending on which activities are being considered. Also, the method does not account for discrete activities.

3.0 REPETITIVE SCHEDULING AND SIMULATION

This chapter explores the research done into creating repetitive scheduling programs/software and highlights the advantages and disadvantages of each. Section 3.1 – 3.10 introduce the programs and present the advantages and disadvantages of each. Section 3.11 summarizes the findings of the comparison.

3.1 SIREN (Simulation of Repetitive Networks)

SIREN (Kavanagh 1985) requires the user to input a precedence diagram for the repetitive unit of construction (i.e. one floor of a skyscraper) and any additional "sub networks" that are not a part of the repetitive sequence. The input program, coded in GPSS, is menu driven and utilizes prompting and data verification to eliminate errors; there is no need to learn a computer language. A deterministic analysis is performed first to obtain crew/equipment utilization and then a stochastic analysis is executed which provides confidence intervals. Activity durations and weather can only be modeled as Erlang, Uniform or Normal distributions. A bar chart schedule and cumulative cost curve are produced as well as crew and equipment utilization graphs. No linear schedule is produced.

SIREN does not take into account time and space buffers between activities or perform any type of critical path analysis. No information is gathered on activity criticality. In addition, Kavanagh (1985) recognizes that the objective of a user-friendly system for site personnel was hindered. Kavanagh goes on to say that no firm conclusions regarding SIREN's applicability can be made until it has been used to model a number of projects.

3.2 SYRUS – System for Repetitive Unit Scheduling

SYRUS (Arditi and Psarros 1987) is a menu driven program that integrates a network diagram with linear scheduling methods. The user inputs project information (name, start date, number of activities, target production rate, etc.) and activity information (man hours required, minimum time buffer, men per crew, etc.). The program performs network analysis on the unit network and then uses that information to perform the linear analysis. The linear schedule is plotted for the critical activities or the user can choose which activities they wish to see on the schedule.

The applicability of SYRUS to real world situations is limited as the system does not include discrete activities, milestones, the effects of the learning curve and uncertainty in activity durations. In addition, SYRUS can only handle a maximum of 21 activities and 50 repetitive networks (Tokdemir 2003).

3.3 REPCON (Representing Construction)

REPCON (Russell 1990) was developed with the intention to marry network analysis and line of balance techniques, which resulted in a family of planning structures. Menus are a key ingredient in the user interface and input information can come from external sources or through a common database. Output information can be a hard copy report, visual displays or information transferred to a database. REPCON produces a linear schedule, bar chart and progress chart to accommodate various cognitive styles (Russell and Wong 1993). The program allows for various advanced functions such as subcontractor control, cash flow management and procurement.

Although a comprehensive system, REPCON does not incorporate stochastic analysis. In addition, although the user interface is menu driven, training of site personnel would be required to utilize this program.

3.4 RUSS – Repetitive Unit Scheduling System

RUSS (Suh 1993) is a menu driven program with inputs such as project data, activity data time/space dependencies, crew sizes and cost information. Once all of the data is entered, the program generates paths, calculates durations, considers milestones and optimizes the schedule. The linear schedule is produced as well as weekly bar-charts and cost information.

The current version of RUSS assumes that the number of crews remains constant throughout the project (Arditi et al. 2001) and does not incorporate any uncertainty into project durations. In addition, RUSS can only handle a maximum of 100 activities and 100 repetitive networks (Tokdemir 2003).

3.5 SCaRC - Space-Constrained and Resource-Constrained

SCaRC (Thabet and Beliveau 1997) is a program designed to schedule repetitive floors in multistory projects. The user must input a logical network for a typical floor, any vertical construction logic constraints, estimates of normal and maximum duration values, resource requirements, space demand and availability. The system is comprised of a database, schema structures and knowledge modules. The database provides an interactive user interface to define scheduling data and view the resulting schedule. The schema structures and knowledge modules are responsible for extracting the input data and generating the schedule.

The idea of making working space a resource for scheduling projects with limited working space has merit. However, SCaRC does not include discrete tasks, perform critical path analysis, apply the effects of the learning curve or integrate uncertainty in a way that would be meaningful to the industry.

3.6 LCPM – Linear Construction Project Manager

LCPM (El-Sayegh 1998) was developed using Microsoft Access and has two major functions; company specific tables that store information about equipment, materials and crews and scheduling specific projects. The software uses various forms to facilitate data entry and produces a variety of reports in addition to the linear schedule.

The method behind the program was created to schedule continuous linear projects such as highways and would be of limited use for discrete linear projects such as high-rises. In addition, Yamin (2001) points out that LCPM does not represent which activities are critical and when they become critical. Yamin goes on to say that the LCPM method does not calculate the effects of linear continuous activities. This is due to the assumptions that are made such as not incorporating space buffers and discrete tasks being scheduled separately. Uncertainty is incorporate into the model; however it takes a long time to run the stochastic models (El-Sayegh 1998).

3.7 PULSS - Purdue University Linear Scheduling Software

PULSS (Harmelink and Yamin 2001) was developed for the Indiana Department of Transportation as a method of producing linear schedules for use in planning and managing highway construction projects. The prototype software was developed within a Computer Aided Design (CAD) environment, but the authors note that this may not be the best environment as it is costly to contractors. Computerized spreadsheets are used to convert dates into Julian dates (i.e. the number of elapsed days since January 1, 1900) and this Julian date is used as the Y coordinate. The other attributes represented by the CAD environment are summarized in Table 3-1.

Table 3-1 Attribute Relationships of PULSS (adapted from (Harmelink and Yamin 2001))

Activity Characteristic	CAD Equivalence	Description
Name	LAYER name of object	Each activity exists on a unique layer that can be made visible/invisible and have a unique color
Start/End Location	X coordinate in drawing	
Start/End Date	Y coordinate in drawing	
Duration	Delta Y coordinate	Difference in Y coordinates of the start and end points of activities
Productivity	Slope of line between start and end point	Measured in units of space per units of time
Resources	Metadata (extended data)	Entity extensions that can be associated with particular activities. Include the number and type of resources an activity consumes

PULSS uses the method introduced by Harmelink and Rowings (1998) to perform critical path analysis, however, the program cannot calculate the critical path if there is a period of time when no activity is being executed. PULSS also assumes that all activities are continuous, even if they are intermittent.

Harmelink & Yamin (2001) suggest future work such as solving visualization issues when activities occur at the same location, improving the user interface and onscreen presentation of the linear schedule, and incorporating uncertainty in order to perform statistical analysis. PULSS does not incorporate user defined buffers, milestones or the effects of the learning curve. In addition, it is solely up to the user to ensure that technological precedence is maintained.

3.8 ALISS – Advanced Linear Scheduling System

ALISS (Tokdemir 2003) is a result of research aimed to improve the previous versions of SYRUS and RUSS, specifically incorporating a more realistic learning model. The program has three different user interfaces; a visual basic program, a Microsoft Access database and a web interface. It also includes an administrative interface to add new users and modify security levels. The program produces the linear schedule, bar charts and cost curves.

ALISS improves on SYRUS and RUSS by incorporating a more realistic learning model, as well as including a server and web module. However, Tokdemir (2003) notes that the programs required to run the server version of the program are quite expensive. ALISS does not include uncertainty in any form and also only incorporates discrete activities on the bar charts.

3.9 Velocity 1.0

Velocity 1.0 (Duffy 2009) is a framework for linear scheduling that accounts for variances in production rates developed specifically for the pipeline industry. It is a Microsoft Excel based program that implements the LSM_{VPR} algorithm explained earlier in Section 2.4.

Velocity 1.0 does not include milestones, discrete activities or uncertainty in activity durations. No formal critical path analysis is performed, however, the colors added to the working windows provides an indication of activity performance. Its focus on the visual aspects of the schedule and the pipeline industry hinder its ability to be useful on a variety of repetitive construction projects.

3.10 OTHER

Moselhi & Hassanein (2003) introduce a model implemented in prototype software that operates in a Microsoft Windows environment. The purpose of the model is to minimize total construction cost, construction duration or their combined impact. The model accounts for transverse obstructions (i.e. a river), utilizes resource-driven scheduling, incorporates continuous and discrete activities, allows for multiple predecessors and successors, accounts for variations in quantity of work and the impact of inclement weather. Microsoft Access is utilized to store data related to owned equipment/labor, assignment dates of equipment/labor and equipment rental firms. The model outputs the linear schedule, bar charts and various other reports.

The model incorporates time buffers, but not space buffers and also excludes milestones, critical path analysis and uncertainty in activity durations. In addition, the linear schedule created is not visually useful as the activities are not labelled. This is most likely due to the fact that crew assignment to achieve the optimum schedule is the focus of the program, not the linear schedule itself (Moselhi and Hassanein 2003).

3.11 Summary

From the comparison of the various existing linear scheduling programs/software it is evident that the major gap in almost all of the programs is their inability to perform stochastic analysis. The few programs that do incorporate uncertainty either restrict the distributions that can be modelled and/or take an extended period of time to run the models and obtain results. Another issue seen in many of the programs is the quality of the user interface or a requirement of the user to be trained in a simulation language to operate the program. Finally, many of the programs are designed for a specific purpose (i.e. vertical construction, pipelines, high rises etc.) and would not be of use for other types of repetitive construction. Table 3-2 provides a summary of the comparison.

Table 3-2 Summary of Comparison

	SIREN	SYRUS	REPCON	RUSS	SCaRC	LCPM	PULSS	ALISS	Velocity 1.0	Other
Natural Rhythm		✓	✓	✓	✓	✓	✓	✓	✓	✓
Dependencies		✓	✓	✓	✓			✓	✓	
Resources	✓		✓	✓	✓	✓	✓	✓	✓	✓
Milestones	✓		✓					✓		
Continuous/ Discrete	✓		✓			✓	✓			✓
Critical		✓	✓				✓	✓	✓*	
Learning Curve	✓		✓					✓	✓	✓
Scale		✓	✓	✓	✓	✓	✓		✓	✓
User Friendly		✓	✓	✓	✓	✓		✓	✓	
Uncertainty	✓*					✓				

*limited

4.0 LINEAR SCHEDULING TOOL

When the need for industry acceptance is combined with the limitations of existing software, the naturally arising solution is a framework built upon simulation concepts. Applying simulation to the traditional linear scheduling problem would allow for: (1) The creation of models with no knowledge of computer programming languages; (2) The ability to model activities, resources, buffers, milestones, etc. in a natural way; and (3) The ability to utilize Monte Carlo simulation for stochastic analysis. This chapter provides an in depth look at the development of a framework to achieve this. Section 4.1 summarizes the requirements of the framework. Section 4.2 presents the algorithm for the framework. Section 4.3 introduces Simphony as the program in which to implement the framework as a special purpose simulation template. Sections 4.4 – 4.8 outline the elements that make up the template. Section 4.9 outlines the logic and error tracking incorporated into the template. A user manual for the template can be found in the Appendix.

4.1 Requirements

It has previously been discussed that a major reason for the lack of linear scheduling software is the complicated nature of the programs. Arditi et al. (2002) discuss what a computerized system for linear scheduling would need to address in order to be of value for managers of repetitive construction. These requirements are (Arditi et al. 2002):

1. All activities start using one crew and operate at a production rate equal to their natural rhythm. Once the schedule is created, it can be compared to contractual deadlines and accelerated if need be
2. Time and space dependencies must be incorporated
3. Resources required for activities, resource usage histograms and resource limitations must be included
4. Important milestones must be incorporated. Once the schedule is completed, milestones are identified. If required, activities before a milestone can be accelerated to meet the deadline
5. Both nonlinear (production rate varies from location to location) and discrete activities (one-off activities) must be incorporated
6. Critical and near-critical activities must be singled out in a method unique to linear
7. The effects of a learning curve must be incorporated due to the repetitive nature of the construction
8. An adequate number of lines must be selected as well as an appropriate scale to clearly communicate the information on the schedule

After reviewing existing programs, a user friendly interface, the integration of uncertainty and the ability to model various types of repetitive projects are also important features. These requirements will be the guidelines in the development of the framework.

4.2 Algorithm

As previously mentioned, applying simulation in the development of the framework would be beneficial. Discrete event simulation consists of a simulation clock, action events and virtual travelers through these events called entities. When the clock is advanced, the system is scanned to determine whether flow unit movement should take place (Halpin and Riggs 1992). Figure 4-1 shows an example of discrete event simulation. For more information on discrete event simulation refer to (Lu 2003).

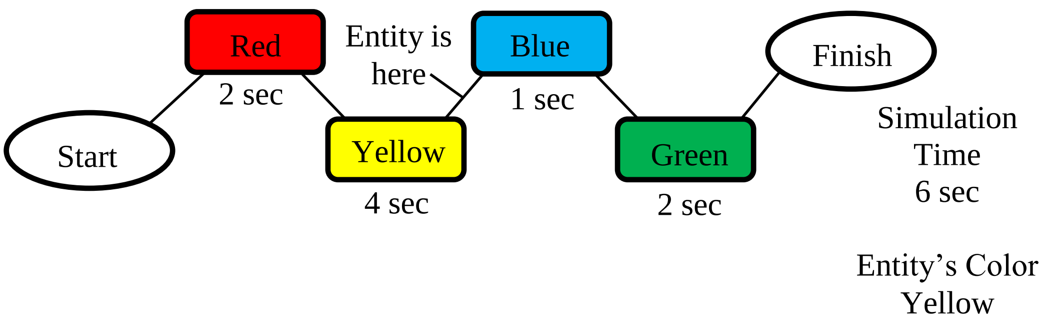


Figure 4-1 Discrete Event Simulation (Source: (Sabha 2012))

In the proposed framework, the entities passing through the system will represent the units (i.e. floors in a high-rise building, meters in a highway, lengths of pipe in a pipeline etc.) in the construction process being modelled. These units will be stamped with an ordinal number (representing which unit they are in the project) and will pass through the action events (construction activities) one at a time in sequential order. The remainder of this section will focus on addressing the requirements laid out by Arditi et al. (2002) as well as the requirements found by analyzing existing programs.

4.2.1 Natural Rhythm & Learning Curve

More often than not the production rate of an activity varies throughout the length of a project. For example, on a highway project, there may be light clearing and grubbing at the beginning and heavier brush towards the end; on a high rise building, a certain number of floors may have custom flooring that takes longer to install. Rather than averaging the production rate for the entire activity, the user will be able to input the production rate for each unit individually or for all of the units. Once the schedule is created, the user can modify the original inputs in order to meet deadlines if required. This will also allow the user to manually apply the effects of the learning curve to the durations; earlier units will have lower production rates than later units.

4.2.2 Time and Space Dependencies

Time dependency will be incorporated by allowing simulation time to pass before the unit can begin the next activity. Because the units pass through one at a time, the amount of time each unit is delayed must take into account the duration of the previous activity. This can be done in one of two ways:

1. If the previous activities duration is larger than the time required between the two activities, the unit will not be delayed as the required lead time is incorporated in the previous duration.
2. If the previous activities duration is smaller than the time required the unit will be delayed. The length of the delay will be equal to the difference between the delay required and the previous activities duration.

Space dependency will be incorporated by maintaining a queue of units. For example if the length of space required between two activities is 3 units, then units 1, 2 and 3 would be queued. When unit 4 arrives the first unit that entered the queue (in this case unit 1) will be allowed to continue on to the next activity. This example is shown in Figure 4.2. To avoid units not making it through the entire simulation, the last unit in the project will release all of the units left in the queue, regardless of the queue length.

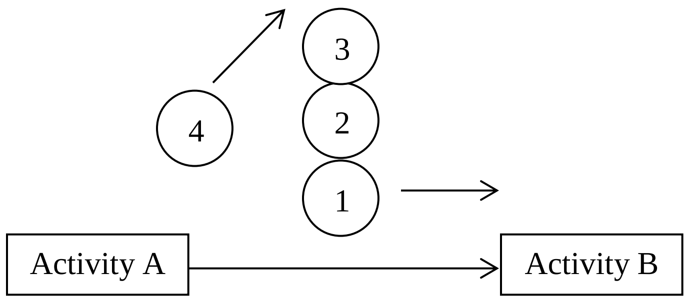


Figure 4-2 Space Dependency

4.2.3 Resources

On nearly all construction sites there is a limit to the amount of material that can be stored on site; material is often delivered to site on a ‘just in time’ basis. This will be incorporated by allowing the user to specify a maximum amount of material that can be accommodated on site. The user can also specify the frequency of material delivery and how much material is delivered each time.

4.2.4 Milestones

Milestones will be incorporated right on the schedule. The user can specify the time at which a milestone should occur as well as a name for that milestone. The milestone will be plotted on the linear schedule and the schedule can then be accelerated if need be.

4.2.5 Continuous & Discrete Activities

The algorithms for incorporating both continuous and discrete activities can be seen in Figure 4-3 and Figure 4-4 respectively.

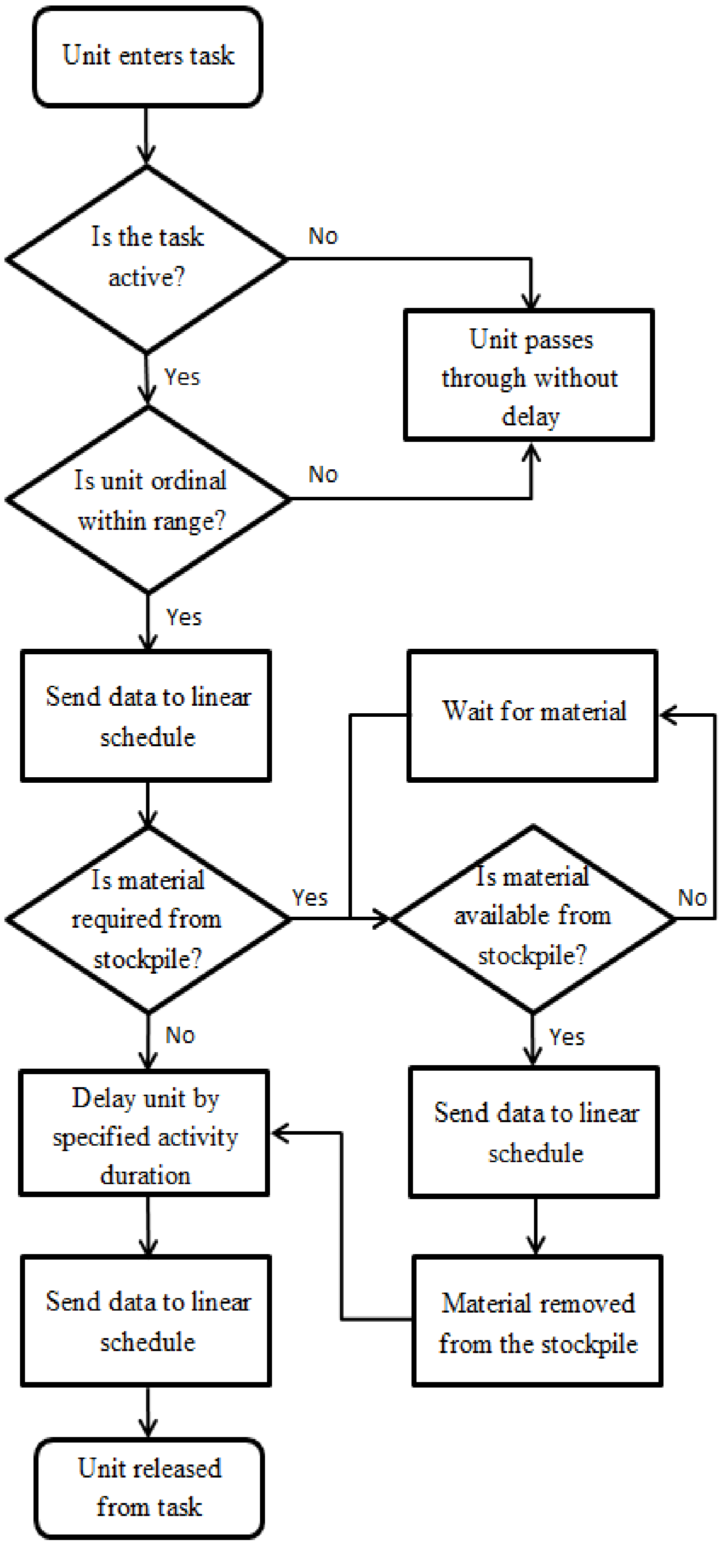


Figure 4-3 Continuous Task Algorithm

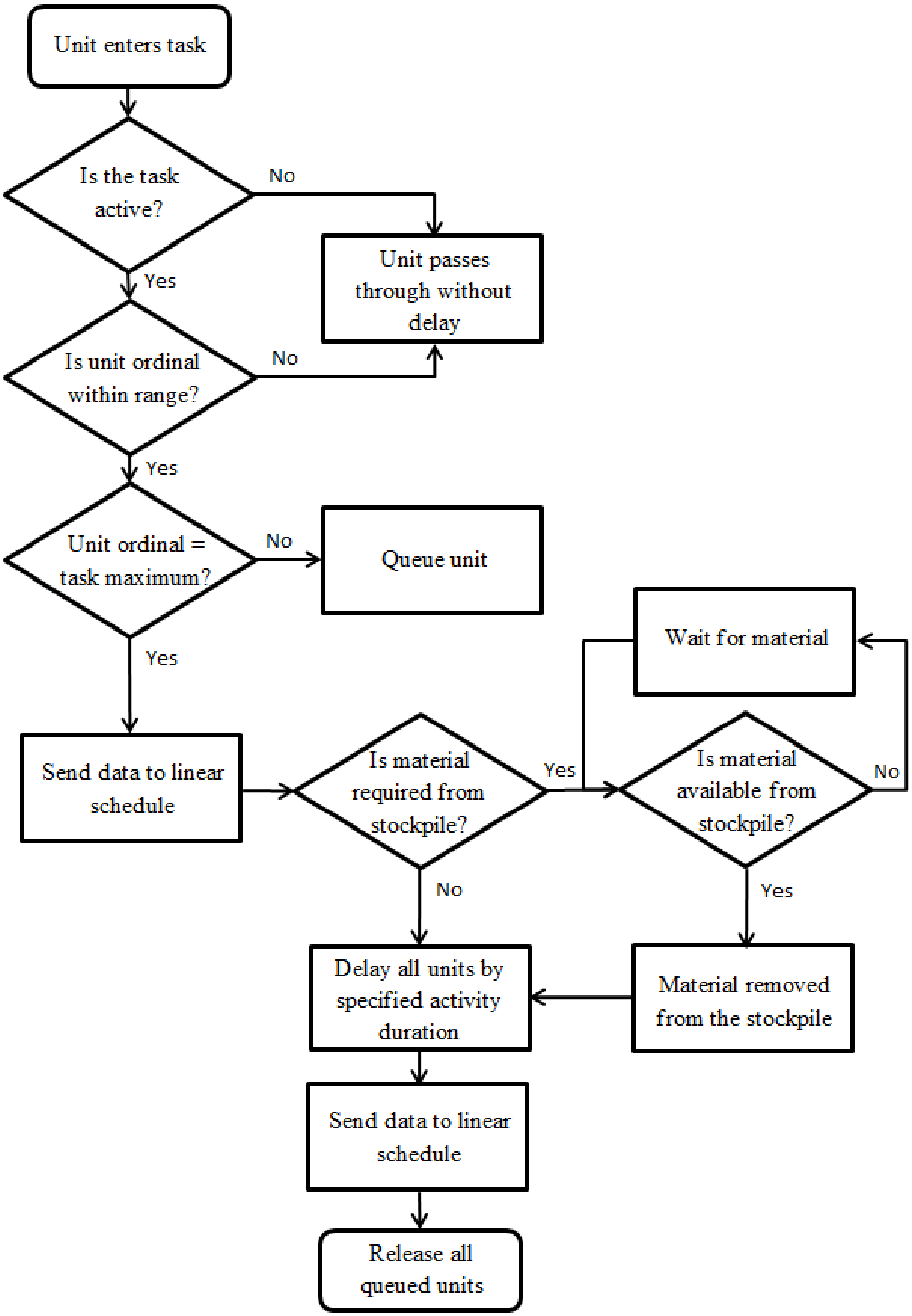


Figure 4-4 Discrete Task Algorithm

The data that will be sent to the linear schedule is the ordinal number of the unit(s) and the simulation time. The linear schedule will plot the point(s) for that activity as a single series (continuous activity) or single rectangle (discrete activity). The purpose of collecting more than one data point is to be able to see the breaks in work continuity due to material delivery delay, imbalances in production rates, etc. on the schedule.

4.2.6 Number of Lines/Scale

The user will be able to specify a different color for each task as well as adjust the transparency of the lines on the schedule. If the color of the task is changed, the schedule should reflect this change. A legend will assist the user in identifying which lines/rectangles represent which activities.

4.2.7 Uncertainty & Criticalness

As mentioned before, using simulation allows for the use of the Monte Carlo method to incorporate uncertainty. The user will be able to specify time inputs as distributions and run the simulation numerous times. Each run will produce its own schedule and a cumulative schedule will also be produced. This cumulative schedule will show the maximum and minimum times values and shade the area in between to represent the possible duration of the activity. The user can then look at this cumulative schedule and visually identify areas of criticality/concern based on the level of overlap between two activities.

4.2.8 Various Types of Projects

In order to accommodate vertical projects, the user will be able to specify when the next unit can begin construction. For example, if a high rise building is made up of 10 floors, the first unit will begin construction and the other 9 units will be queued. When the first floor has completed enough activities to make it possible to start the second floor, the second unit will be released for construction. This allows the user to model both vertical and horizontal construction processes while maintaining technological precedence.

4.3 Simphony

All of the requirements laid out have been addressed thus far except for a simple user interface. As mentioned previously, Simphony is a Microsoft based program that allows for the development of special purpose simulation (SPS) tools. The major benefit of SPS is that the user does not need to learn a programming language to utilize the tool. In addition, Simphony has a simple, visual user interface that resembles a drawing board. Modelling elements are dragged and dropped into the modeling environment and connected by clicking and dragging between output/input ports. Elements can also be renamed.

A screenshot of the Simphony modeling environment is shown in Figure 4-5. The various templates installed can be seen on the left hand side of the screen and when expanded, show the modelling elements available for that template as seen in Figure 4-6 (a). The properties of each scenario in the model can be seen on the right hand side of the model. The user can specify run times, run counts and the unit of time used in the model. It is crucial that all time properties entered in the model are of the same time unit (i.e. all minutes, hours, days etc.). A screenshot of the scenario is shown in Figure 4-6 (b). The trace and error window can be seen along the bottom of the screen. The error window is where the model communicates logical errors within the model to the user. Section 4.9 further discusses error detection.

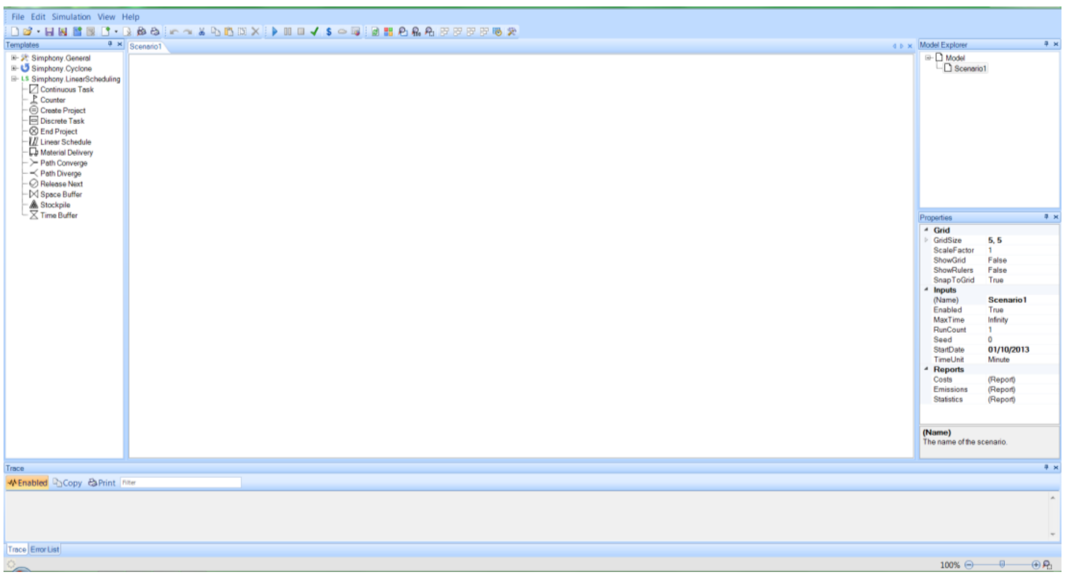


Figure 4-5 Simphony Modelling Environment

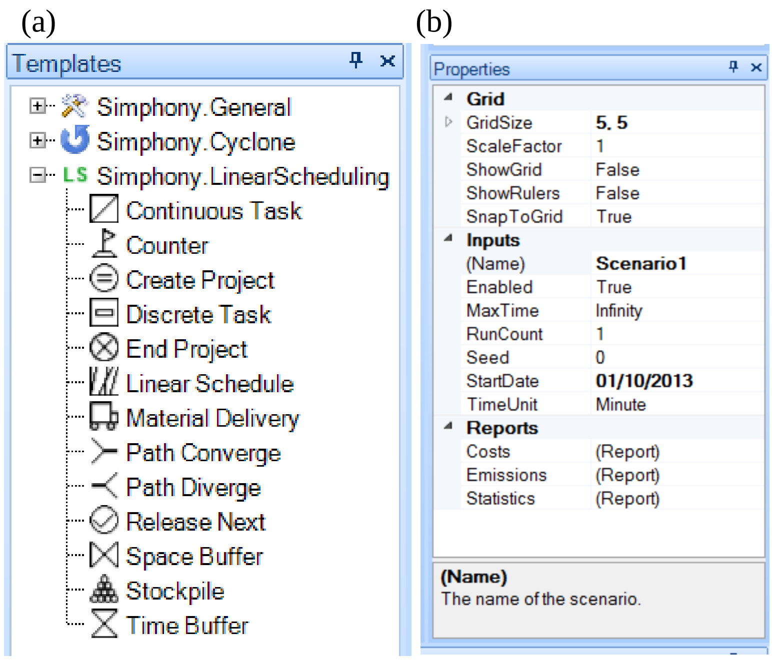


Figure 4-6 Simphony (a) Templates & Elements (b) Scenario Properties

4.4 Task Elements

The task elements are the building blocks of the model as each task represents a different activity in the project.

4.4.1 Continuous Task

The continuous task element (Figure 4-7) represents a single repetitive activity that occurs along a portion of the project, or along the entire length of the project. The user inputs for this element are:

- Active – Should this task be included in the schedule
- Duration
  - Activity duration – Represents the length of time it takes to complete this task for one unit within the specified range. The activity duration can be entered as either a constant value or as a distribution

- Range – Represents the range of units that this activity occurs over. Multiple ranges can be specified for the same task with different durations which allows the user to account for variations in the project (i.e. different soil types) and the learning curve (i.e. increasing the production rate as the project progresses). There can be no gaps in the ranges (i.e. 1-10, 15-20). If a gap is required, it is recommended that separate continuous task elements be utilized for each section

- Line color – The user may choose which color they would like the task to appear as on the schedule. All tasks must have a different line color, unless they are all black
- Stockpile – The stockpile from which the task should request resources
- Stockpile quantity – The number of resources required per unit to complete this task

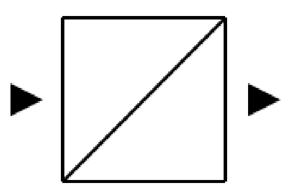


Figure 4-7 Continuous Task Element

4.4.2 Discrete Task

The discrete task element (Figure 4-8) represents a single non-repetitive activity that occurs at a certain location for a certain amount of time. The user inputs for this element are:

- Active – Should this task be included in the schedule
- Activity duration – Represents the length of time it takes to complete this task. The activity duration can be entered as either a constant value or as a distribution
- Line color – The user may choose which color they would like the task to appear as on the schedule. All tasks must have a different line color, unless they are all black
- Maximum/Minimum – The range of units that this task occurs on. A range of 2-5 would imply that this task occurs at units 2, 3, 4 and 5. A range of 2-2 would imply that this task occurs only at unit 2.
- Stockpile – The stockpile from which the task should request resources
- Stockpile quantity – The total number of resources required to complete this task

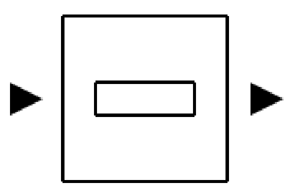


Figure 4-8 Discrete Task Element

4.5 Buffer Elements

The buffer elements allow the user to model time and space dependencies between activities.

4.5.1 Space Buffer

The space buffer element (Figure 4-9) models the minimum space dependency between two activities. The inputs for this element are:

- Buffer length – The number of units required between two activities

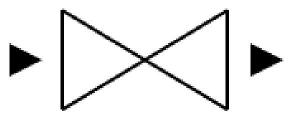


Figure 4-9 Space Buffer Element

4.5.2 Time Buffer

The time buffer element (Figure 4-10) models the minimum time dependency between two activities. The inputs for this element are:

- Buffer Time – The amount of time required between two activities. The buffer time can be entered as either a constant value or as a distribution

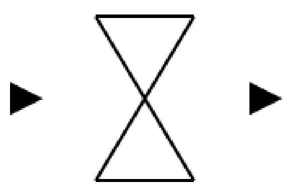


Figure 4-10 Time Buffer Element

4.6 Project Elements

The project elements allow the user to define project parameters as well as view the linear schedule created by the model.

4.6.1 Create Project

The create project element (Figure 4-11) is where the project is initialized and the units are created. Only one create project element can be added to the model, and there must be a create project element in the model for it to run. The user inputs for this element are:

- Initial Units – The number of units to be released at the beginning of the project
- Milestones
    - Name – The name of the milestone. This name must be different for each milestone and must not be the same as any of the task names
    - Time – The time at which this milestone occurs
- Release First – The time at which the initial units are made available. If no value is specified, the number of initial units specified will be released at time zero
- Time between units – The time required between successive units. If no value is specified, the units will be released with no delay between them
- Unit –The units that one entity represents. For example, m, lengths of pipe, floors etc.
- Units in Project – The number of units in the project. For example, if a pipeline project was 150m long and each length of pipe was 10m, the user would enter 15 into the units in project field.

The create element outputs the number of units created at the time the simulation stops.

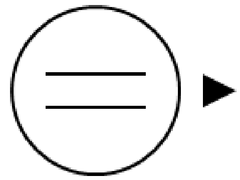


Figure 4-11 Create Project Element

4.6.2 End Project

The end project element (Figure 4-12) represents the end of the project. There can only be one end project element in the model. This element does not require any input from the user nor does it produce any outputs.

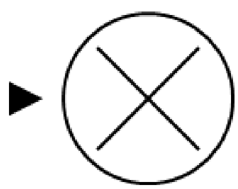


Figure 4-12 End Project Element

4.6.3 Linear Schedule

The linear schedule element (Figure 4-13) displays the schedule produced by the model. There can only be one linear schedule element in the model and there must be a linear schedule element in the model for it to run. The inputs for this element are:

- X-Axis title – The title of the time axis on the linear schedule
- Y-Axis title – The title of the units axis on the linear schedule

The linear schedule has the following outputs:

- Schedule – The linear schedule produced by the model. The linear schedule for each individual run can be viewed as well as the linear schedule for all runs, which shows the minimum and maximum start and finish times for all activities. The schedule that incorporates all runs can be used to visually identify critical activities and areas of concern. The user can change the transparency of the lines and export the data to a Microsoft Excel file. The exported file contains the x and y (time and unit) values for all of the activity lines on the linear schedule.

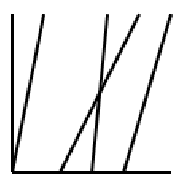


Figure 4-13 Linear Schedule Element

4.7 Resource Elements

The resource elements allow the user to model material resource requirements and limitations.

4.7.1 Material Delivery

The material delivery element (Figure 4-14) allows the user to model resource availability. The inputs for this element are:

- Delivery quantity – The number of resources in each delivery
- First delivery time – The time at which the first delivery is made
- Time between deliveries – The time before the next delivery arrives

- Total quantity – The total number of resources that need to be delivered

The material delivery element outputs the number of resources delivered at the time the simulation stops.

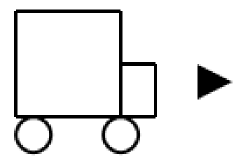


Figure 4-14 Material Delivery Element

4.7.2 Stockpile

The stockpile element (Figure 4-15) allows the user to model resources on site. The inputs for this element are:

- Initial material – The number of resources available on site at the beginning of the simulation
- Maximum material – The maximum number of resources that the stockpile can hold. A value of zero indicates no maximum

The stockpile element outputs the amount of material left in the stockpile when the simulation stops.

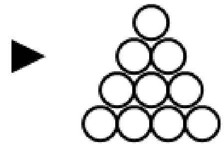


Figure 4-15 Material Stockpile Element

4.8 Other Elements

The remaining elements allow the user to model more complex, real world situations.

4.8.1 Path Diverge

The path diverge element (Figure 4-16) allows a path to split into two separate paths. There must be an equal number of path converge and diverge elements in the model. There are no inputs or outputs for this element.

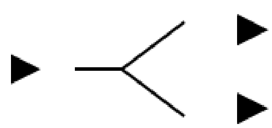


Figure 4-16 Path Diverge Element

4.8.2 Path Converge

The path converge element (Figure 4-17) allows two paths to rejoin into a single path. There must be an equal number of path converge and diverge elements in the model. There are no inputs or outputs for this element.

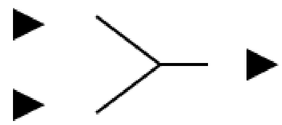


Figure 4-17 Path Converge Element

4.8.3 Release Next

The release next element (Figure 4-18) is used specifically for modelling vertical construction operations and indicates when the next unit is available for construction. For example in a multistory building, the first level floor must be installed, the walls framed and the roof built before the floor can be installed on the second level. If this scenario was modelled, there would be a task element for installing the floor, framing the walls and building the roof followed by a release next element. There can only be one release next element in the model. There are no inputs or outputs for this element.

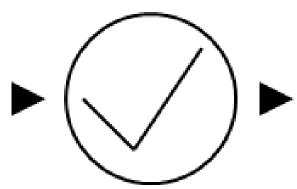


Figure 4-18 Release Next Element

4.8.4 Counter

The counter element (Figure 4-19) allows the user to track units through the model. The user inputs for this element are:

- Initial value – The initial value of the count

- Limit – The limit of the counter. If the count reaches the limit, the simulation will stop. A value of 0 implies no limit

The counter outputs the total number of units observed and the time at which the last unit was observed when the simulation stops.

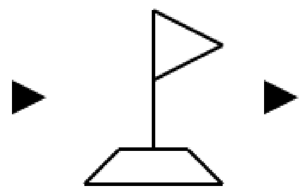


Figure 4-19 Counter Element

4.9 Error Detection

There are two types of messages that Simphony displays for the user; Error messages and warnings. If errors are present, the model will not run, whereas warnings caution the user on proceeding, but will still allow the model to run. Each error/warning displays a message and the element that is causing the issue to assist the user in solving the problem. The following is a list of errors that the linear scheduling special purpose template can produce:

- "A Create Project/Linear Schedule element must be added to the model" – In order for any linear scheduling model to run, one and only one create project and linear schedule element must be present
- "A model may only contain one End Project element" – There can be only one end project element in a model
- "Invalid Stockpile name" – All stockpile elements must have unique names

- “Element input/output points can only have one relationship” – Each input and output port of an element can only be connected to one input/output port of another element. This is to prevent unexpected and unwanted behaviour. If more than one path is required between elements, the path converge and diverge elements can be utilized
- “Ranges cannot overlap” – The unit ranges in the continuous task cannot contain the same values
- “Maximum cannot be larger than the number of units in the project” – For the continuous and discrete tasks, the maximum value for the range of units must be equal to or less than the number of units in the project
- “All Task line colors must be different, unless they are all black” – As stated earlier, all tasks must have different line colors. This is done to make activities easier to identify on the linear schedule
- “The initial value must be smaller than the limit of the counter” – Logically, it does not make sense for the initial value of the counter to be higher than the limit
- “The limit of the counter must be equal to or smaller than the number of units in the project” – It is possible that if the limit of the counter is higher than the number of units in the project the simulation won’t stop
- “A model may only contain one Create Project/End Project/Linear Schedule element”
- “This element has an unconnected output point. Connect it to an End Project element if necessary” – Having unconnected output points could

lead to unexpected results. All paths through the model must lead to the single end project element

- “Milestones must have unique names/different names than Tasks” – The linear schedule plots each milestone as its own series in addition to the tasks. Because of this, each milestone must have a different name than any other milestone or task
- “Initial units released must be larger than the sum of the Space Buffer lengths before the Release Next element” – If the initial number of units is less than the sum of the lengths of the space buffers some units will remain in the space buffer and the simulation will not end. Unexpected results will be produced
- “Material Delivery elements must be connected to a stockpile element” – Material delivery elements can only be connected to stockpile elements and vice versa. The only exception to this is a counter element can be place between a material delivery and stockpile element to track deliveries
- “Number of Path Converge elements must be the same as Path Diverge” – Because the path converge element waits to release a unit until both of the clone units have arrived, there must be an equal number of path converge and diverge elements in the model to ensure that all units are released
- “A model may only contain at most one Release Next element” – There can only be one release next element in a model

- “Buffers cannot be connected to other Buffers” – Whether connected to the same type of buffer or a different type of buffer, buffers cannot be connected to each other
- “The Buffer length must be smaller than the number of units in the project” – Logically it does not make sense to have a space buffer larger than the number of units in the project
- “Maximum material amount not enough for task” – The amount of material to be delivered to a stockpile is not enough to complete the tasks that require material from it
- “Total material will not be enough for all tasks” – There is not enough material in a stockpile to complete all of the tasks that require material from it

The following is a list of warnings that the linear scheduling template can produce:

- “It is possible for the duration to be negative” – When utilizing distributions for activity durations it is sometimes possible to sample a negative duration. To avoid sampling a negative number it is recommended that the distribution be modified. Allowing the model to sample a negative value would produce unexpected results
- “Buffers should be placed on both paths before/after a converge/diverge rather than once after/before” – To avoid unexpected results, it is better to model the buffers on the side of the path converge/diverge with two output points

5.0 APPLICATIONS OF LINEAR SCHEDULING TEMPLATE

This chapter is intended to show the applications of the developed template through a text book sewer line example. This case study will also be used to validate the outputs of the linear scheduling template. Section 5.1 provides the problem statement. Section 5.2 explains the linear scheduling template model. Section 5.3 describes the CYCLONE (Cyclic Operations Network) template model created to validate the linear scheduling model. Section 5.4 presents the results and an analysis of the two models. Section 5.5 provides further examples of applications of the template.

5.1 Overview and Problem Statement

Verification and validation are critical aspects of the development of a template as they determine the degree of accuracy of the templates results and the level of user confidence (Ekyalimpa et al. 2012). Verification of the linear scheduling template was done: (1) using the trace window in Simphony to track the logical sequence of events; (2) using counter elements to track the flow of entities through the developed elements; (3) and by confirming the graphical outputs of the linear schedules of numerous test models. AbouRizk & Halpin (1990) explain that validation of a template is best done by comparing the simulation results of the developed template with: (1) results obtained from other models (when such models exist); (2) results obtained from analytical techniques (when such techniques can estimate the output parameters of concern); and (3) historical or published data. For this research, the template will be validated by comparing the outputs of the linear scheduling template model to the outputs of a CYCLONE

(Halpin and Riggs 1992) template model. The example to be modeled is a sewer line project adapted from Halpin & Riggs (1992).

5.1.1 Problem Statement

A 100ft sewer line is to be constructed beneath the route of an existing city street. The pipe sections are 10 feet long large diameter concrete units. The work tasks are as follows:

1. Pavement removal – A pavement breaker proceeds ahead of the excavation equipment and breaks up 10ft sections of the existing pavement
2. Excavation – A backhoe shovel excavates and clears the trench for the pipe
3. Shoring – A prefabricated trench box is placed in the excavated trench to protect workers in the event of a collapse. The trench box is 10 feet long.
4. Haul – Debris and broken pavement is removed from site by trucks. Trucks return with good fill material. One truck load is equal to 10ft of excavated material or filled length. The site layout only allows for one truck load of good fill material to be dumped at any point in the project. There is no fill material initially available on site.
5. Manual labor – One crew performs the final grading of the bed, aids in placing the pipe and installs a rubber gasket around the pipe joint. A second crew does hand backfill and compaction and then forms and pours the concrete paving

6. Placing the pipe – The pipe is placed and finished by the labor as described above. There is 1 pipe unit initially on site and this is the maximum number of unused pipes that can be present on the work site. Trucks have a maximum carrying capacity of 1 pipe.
7. Backfill – Aside from a small amount done by the laborers, the backfilling operation is performed by the backhoe shovel in 10ft sections
8. Compaction and preparation for paving – This is done by a small roller and manual labor in 10ft sections
9. Concrete repaving of the street – This is done in 10ft sections

A schematic diagram of this work site is shown in Figure 5-1 and the durations of all of the activities are shown in Table 5-1.

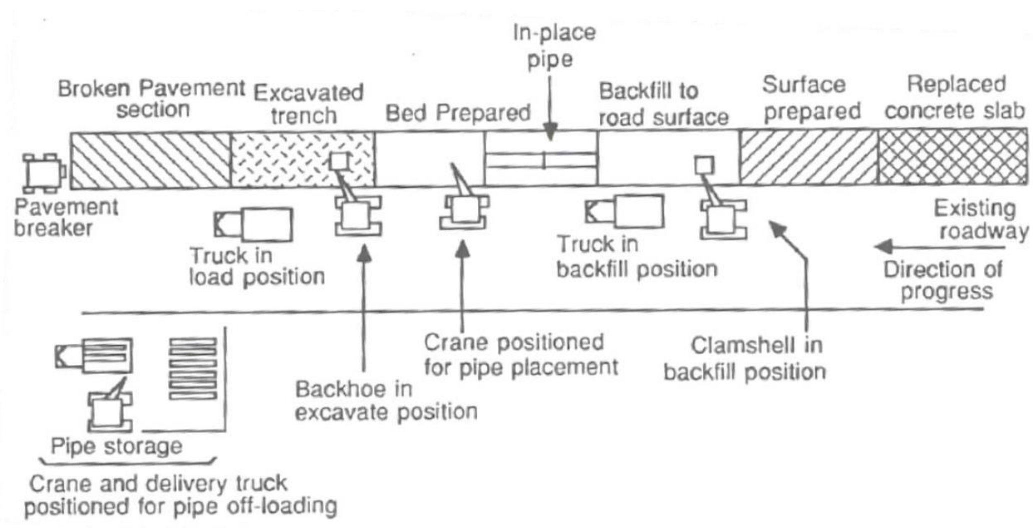


Figure 5-1 Schematic of Work Site (Source: (Halpin and Riggs 1992))

Table 5-1 Activity Durations for Sewer Line Example

Activity	Duration
Break & remove pavement	5min/section
Excavate and load material	11.25min/section
Install/remove trench box	5min/section
Grade trench bed	22.5min/section
Lay pipe	5min/pipe
Connect, seal & finish pipe	5min/pipe
Backfill	15min/section
Compact material	22.5min/section
Form & place rebar	30min/section
Mix & pour concrete	52.5min/section
Travel to/return from dump site	8.7min/12.2min
Dump excavated material	1.5min
Load good fill material	4.9min
Unload good fill material	0.7min
Travel to/return from pipe source	1.7min
Load pipe onto truck	2.6min
Unload pipe into stockpile	2.9min

5.2 Linear Scheduling Template Model

The model developed with the linear scheduling template is shown in Figure 5-2.

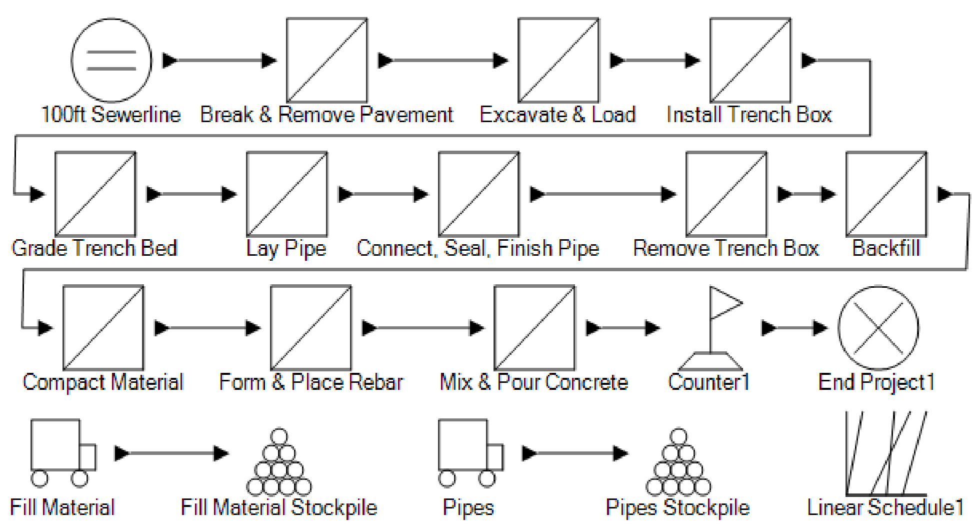


Figure 5-2 Linear Scheduling Template Model

The unit used for the model is 10ft sections of pipe and each activity is represented by a continuous task element. The deterministic linear schedule produced can be seen in Figure 5-3.

5.3 CYCLONE Template Model

The model developed with the CYCLONE template can be seen in Figure 5-4. Initially, 100ft of road is generated and the existing pavement is broken up. The length of the project is then consolidated into 10ft sections. The counter stops the simulation when 10, 10ft sections have arrived.

5.4 Validation of Template

The last time property of the counters in both models is shown in Table 5-2.

Table 5-2 Model Output

Model	**Counter Last Time**
Linear Scheduling	651.25 minutes
CYCLONE	651.25 minutes

The overall simulation time of the deterministic models is identical, validating the output of the linear scheduling template. Now that the results are validated, the template can be used to produce various scenarios and to perform stochastic analysis to determine the optimum schedule.

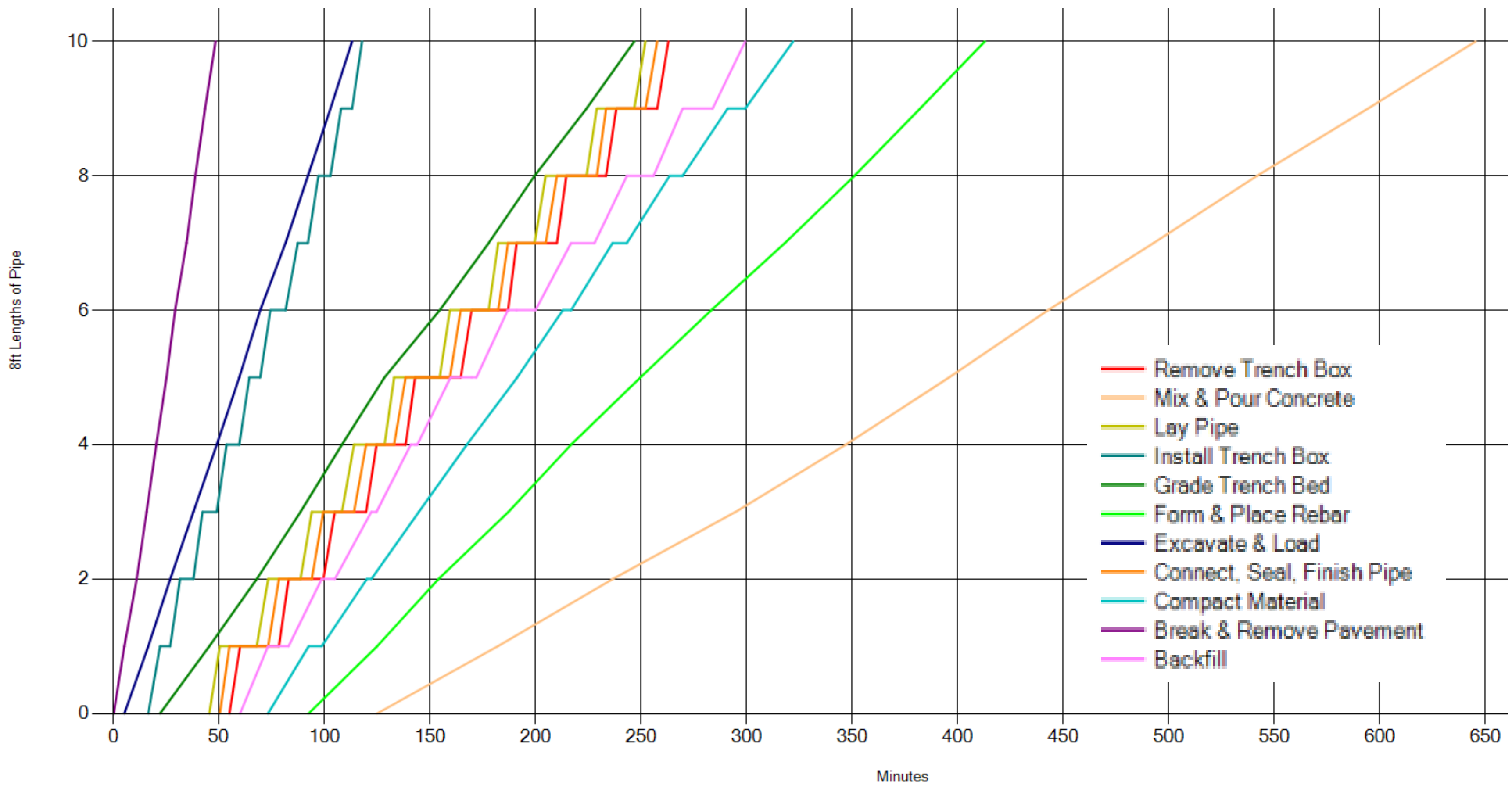


Figure 5-3 Sewer Line Schedule

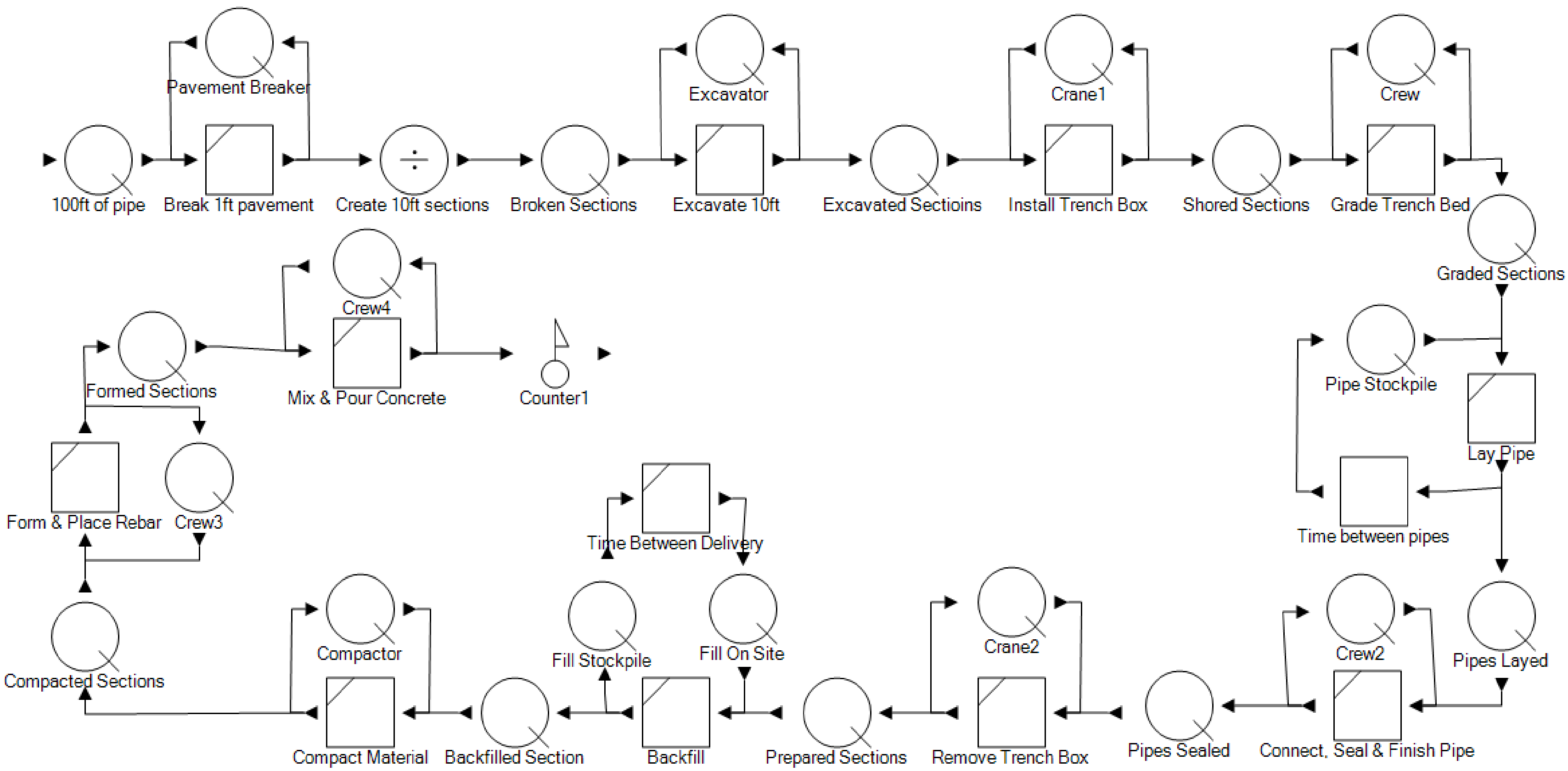


Figure 5-4 CYCLONE Template Model

5.5 Further Applications

In order to highlight all of the linear scheduling templates features, multiple variations of the above example will be presented.

5.5.1 Stochastic Analysis

Assume that all of the activity durations can be modelled by a triangular distribution that is ±20% of the average values shown in Table 5-1. After simulating the model 1000 times, the linear schedule for all of the runs can be seen in Figure 5-5. The overlap in activities indicates areas of concern and allows the user to visually identify the areas that activities become critical. Being able to incorporate uncertainty can uncover information that changes how the project is planned. Having the ability to view various schedules ahead of time has numerous advantages including avoiding material delivery delays, integrating float to avoid activities delaying subsequent activities and the ability to maintain work continuity. Being able to incorporate uncertainty into how the schedule is planned can uncover inform

For example, in this sewer line case study, many of the activities overlap significantly with each other. This could be a major cause for concern due to the fact that a delay in an activity early on would cause delays in the rest of the project. Some activities can be accelerated (excavate & load and grade trench bed) by adding another crew or choosing different equipment. Space and time buffers can be added to the schedule to incorporate float and decrease the amount of overlap between activities. By doing these two things up until the backfill activity, the majority of the overlap can be eliminated. This revised schedule can be seen in Figure 5-6.

It can now be seen on the schedule that the delivery of back fill material is delaying the project. Having the ability to see this outcome ahead of time allows the manager to arrange for more frequent deliveries, well ahead of when they are actually required. After adjusting the delivery frequency, the remainder of the project was analyzed. The resulting schedule can be seen in Figure 5-7. It should be noted that the project’s maximum duration is now slightly higher than the 651.25 minutes previously modelled, but there is considerably less overlap between activities and work continuity is maintained. The model that produced the final revised schedule can be seen in Figure 5-8.

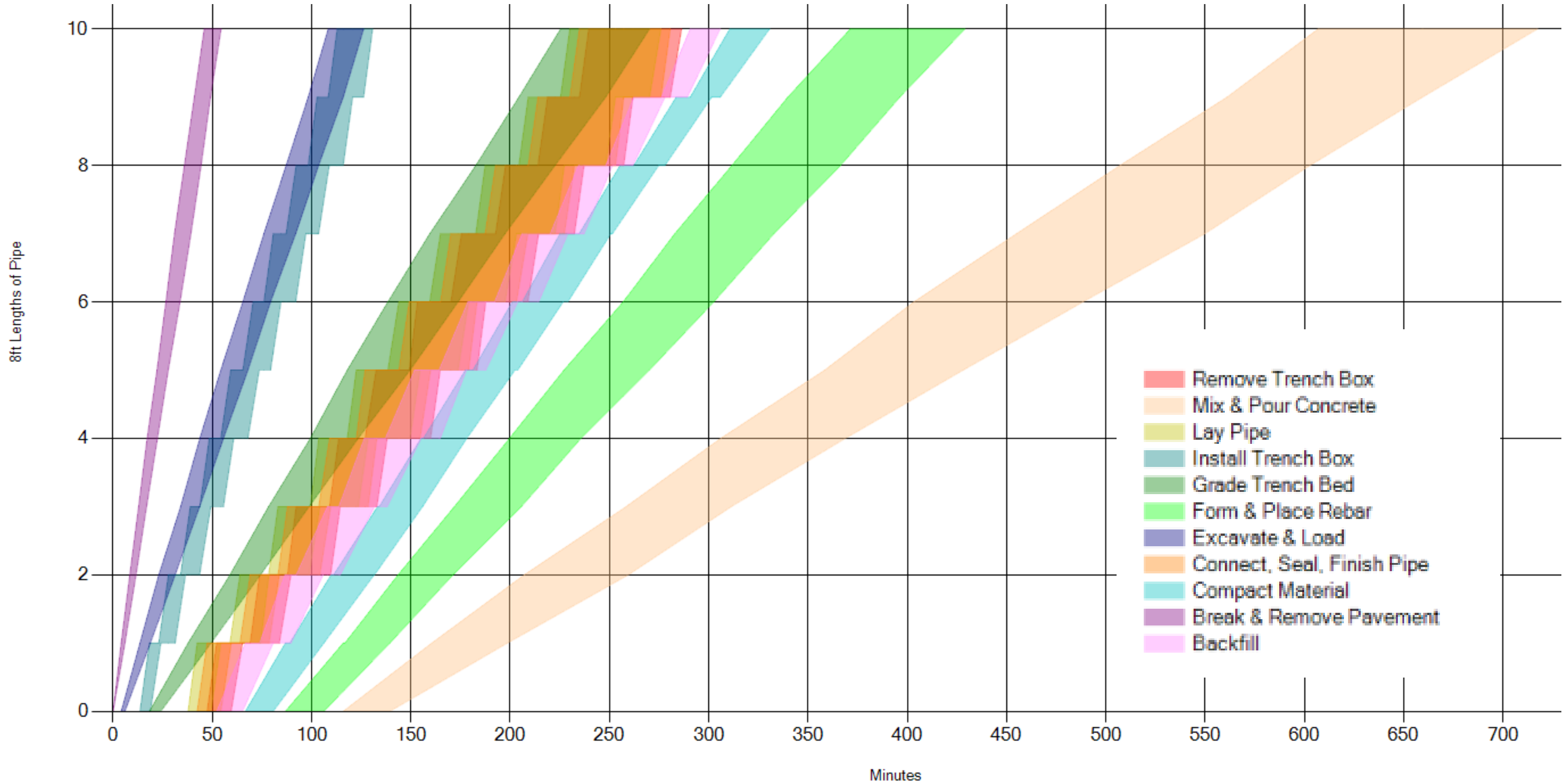


Figure 5-5 Sewer Line Schedule: Cumulative

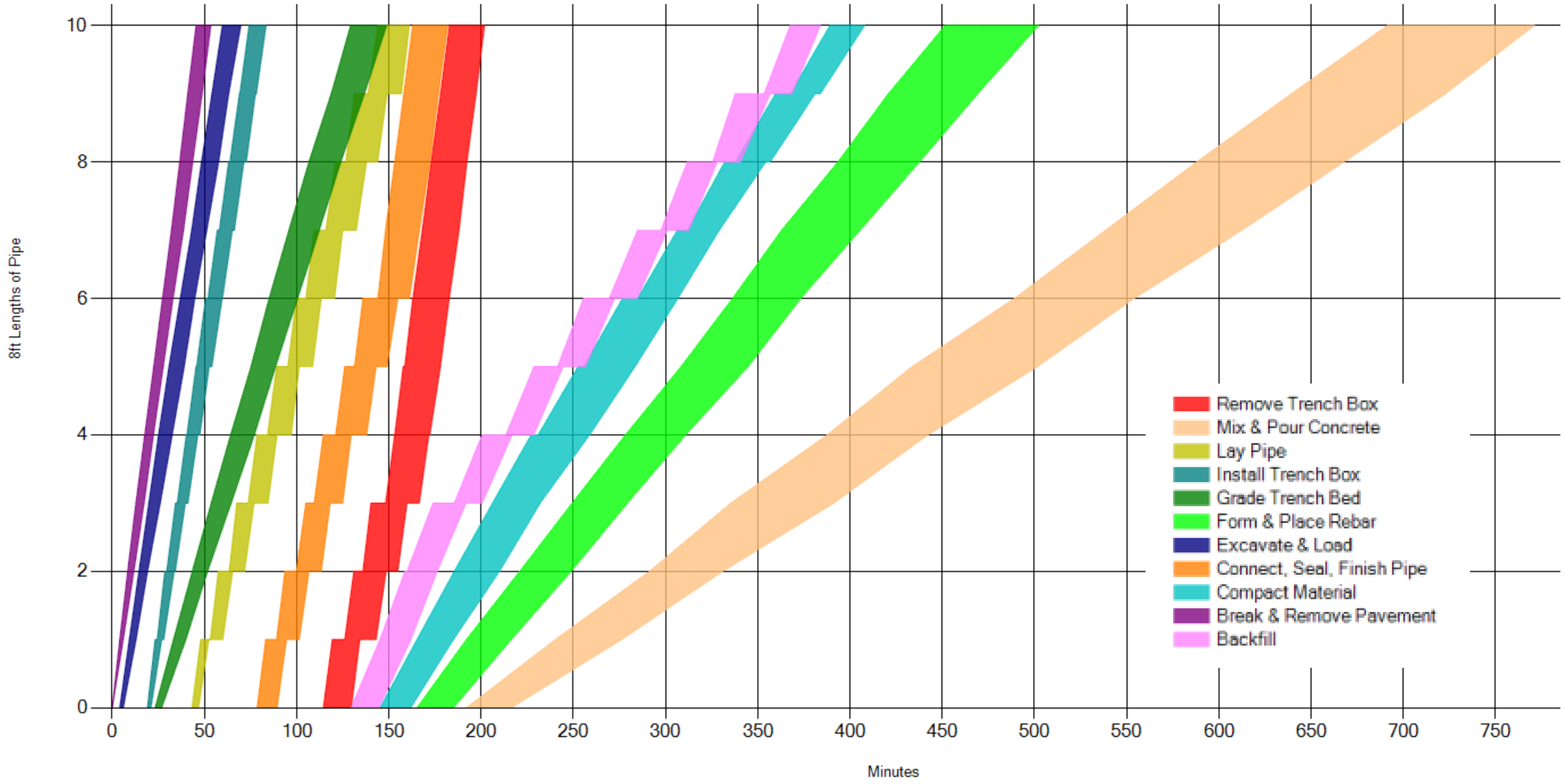


Figure 5-6 Sewer Line Schedule: Revised Cumulative

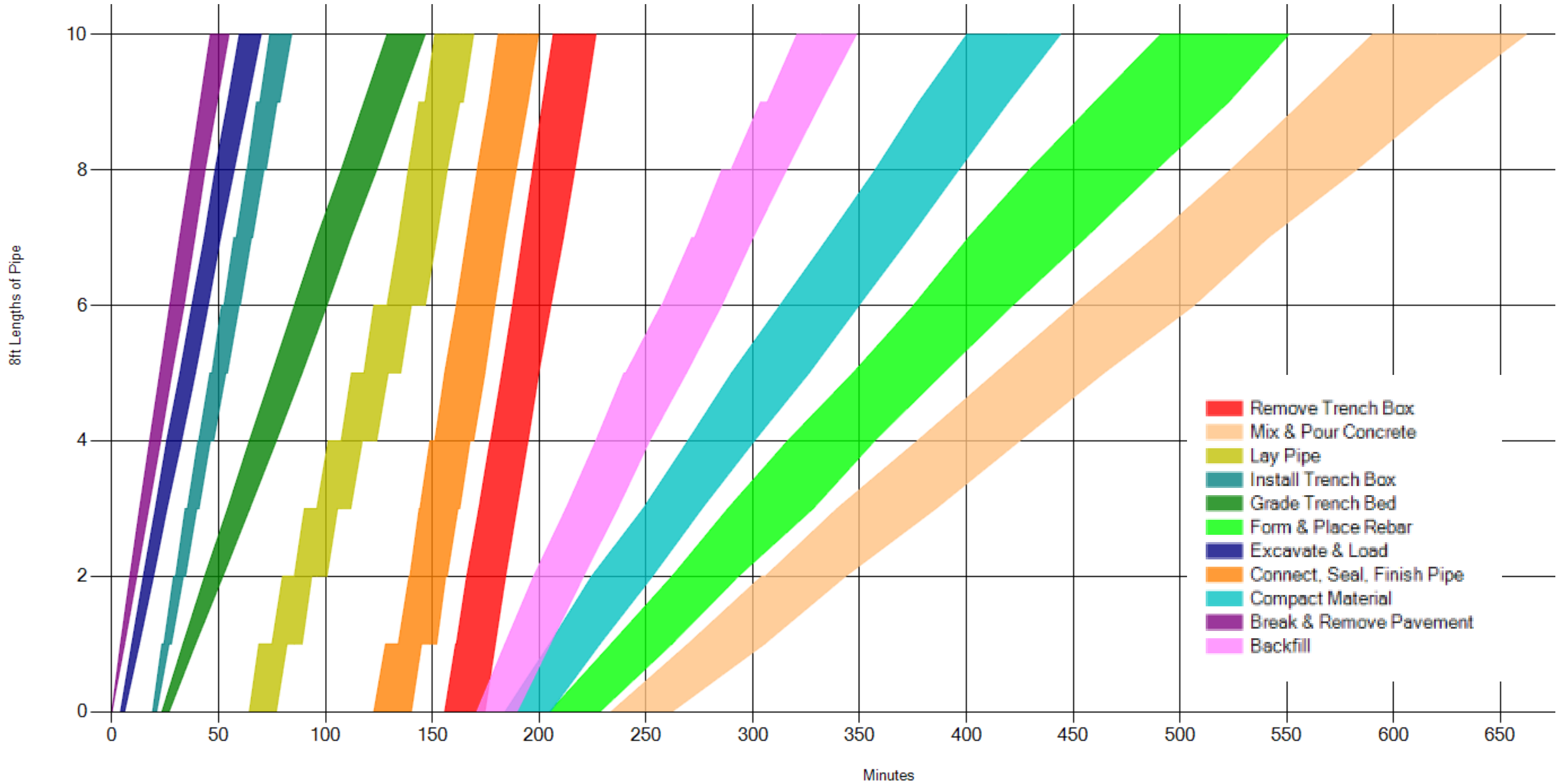


Figure 5-7 Sewer Line Schedule: Final Cumulative

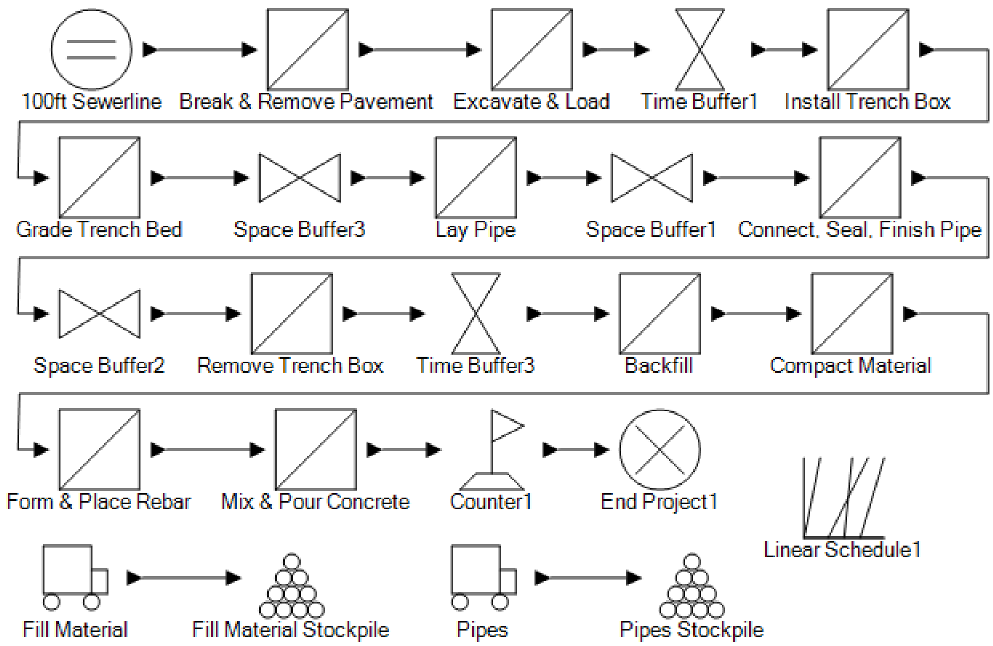


Figure 5-8 Sewer Line: Final Revised Model

5.5.2 Alternate Number of Activities

One of the major advantages of a linear schedule is the ease with which it can be interpreted; however, it is crucial that an appropriate number of activities be selected in order to fully take advantage of this benefit. The deterministic scheduled produced for the sewer line case study (Figure 5-3) is generally easy to read. The only place it becomes difficult to distinguish between activities is the lay pipe, connect, seal & finish and remove trench box activity. The project manager could choose to combine these activities to make the schedule easier to read for site staff. The schedule produced for this scenario is shown in Figure 5-9. With the three activities combined into one, the project duration remains the same (as expected) and the schedule is easier to read.

.

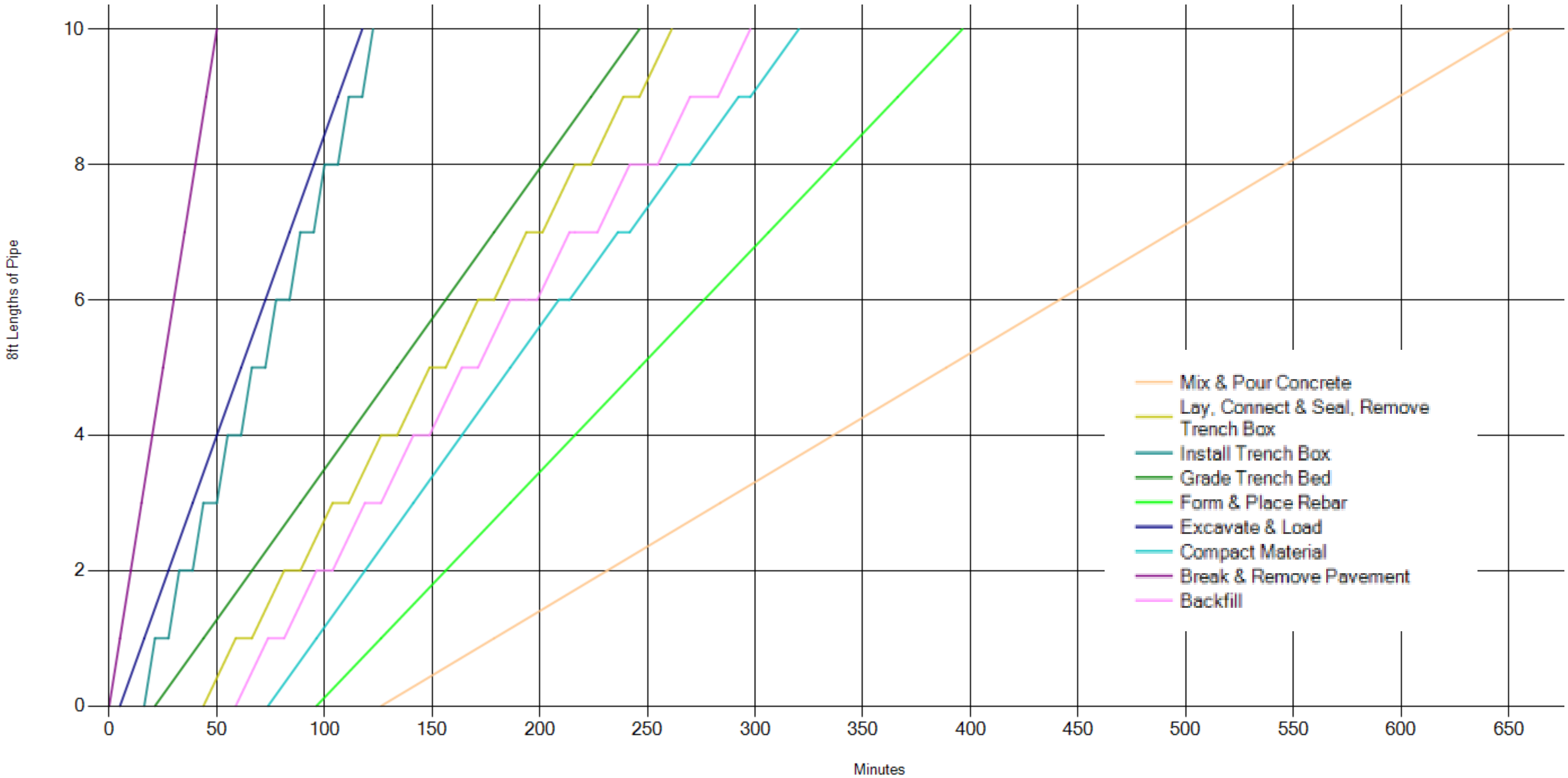


Figure 5-9 Sewer Line Schedule: Alternate Number of Activities

5.5.3 Discrete Task

In this variation, a part of the construction process is a pipe connection that will occupy section 5, 6, and 7 of the project. It will take 2.5 hours ± 10% for the crew to complete this connection and the construction of the connection begins once section 5 of pipe is laid and sealed, but before the trench box is removed. The schedule for this variation can be seen in Figure 5-10. The discrete task appears as a block on the schedule and the effects of this activity on the continuity of the project can be seen. To improve work continuity, the manager could create various models, in a similar fashion to the analysis done in section 5.5.1.

5.5.4 Milestones

For this variation, assume that the contract documents state that the first 50ft (5 sections) of pipe must be finished in 250 minutes and the entire project must be finished in 700 minutes. The project manager can utilize the milestones property in the create element to determine if these dates will be met based on the current rate of production. The model including these milestones can be seen in Figure 5-11. It can be seen from the schedule that not all of the activities will have completed the 5th section by the required time, but that the entire project will be completed well before the deadline. The project manager can then use the template to model various options and decide at what rate the project needs to be accelerated to meet the first deadline.

5.5.5 Progress/As-Built Analysis

The template can also be used to track progress and determine a course of action if the project is behind schedule. Assume that after 200 minutes, the following information was collected on site:

- Break and remove pavement began at minute 0 and was completed at minute 50
- Excavate and load began at minute 5 and was completed at minute 120
- Install trench box began at minute 17 and was completed at minute 125
- Grade trench bed began at minute 20 and is 90% complete
- Lay pipe began at minute 44 and is 70% complete
- Connect, seal & finish pipe began at minute 50 and is 60% complete
- Remove trench box began at minute 54 and is 50% complete
- Backfill began at minute 60 and is 40% complete
- Compact material began at minute 80 and is 30% complete
- Form & place rebar began at minute 100 and is 20% complete
- The mix and pour concrete activity has just begun

Using the progress reported from site the model can be updated to produce an as-built schedule. This schedule can be seen in Figure 5-12. Using the template, the manager can analyze this schedule and choose the best course of action to correct the schedule.

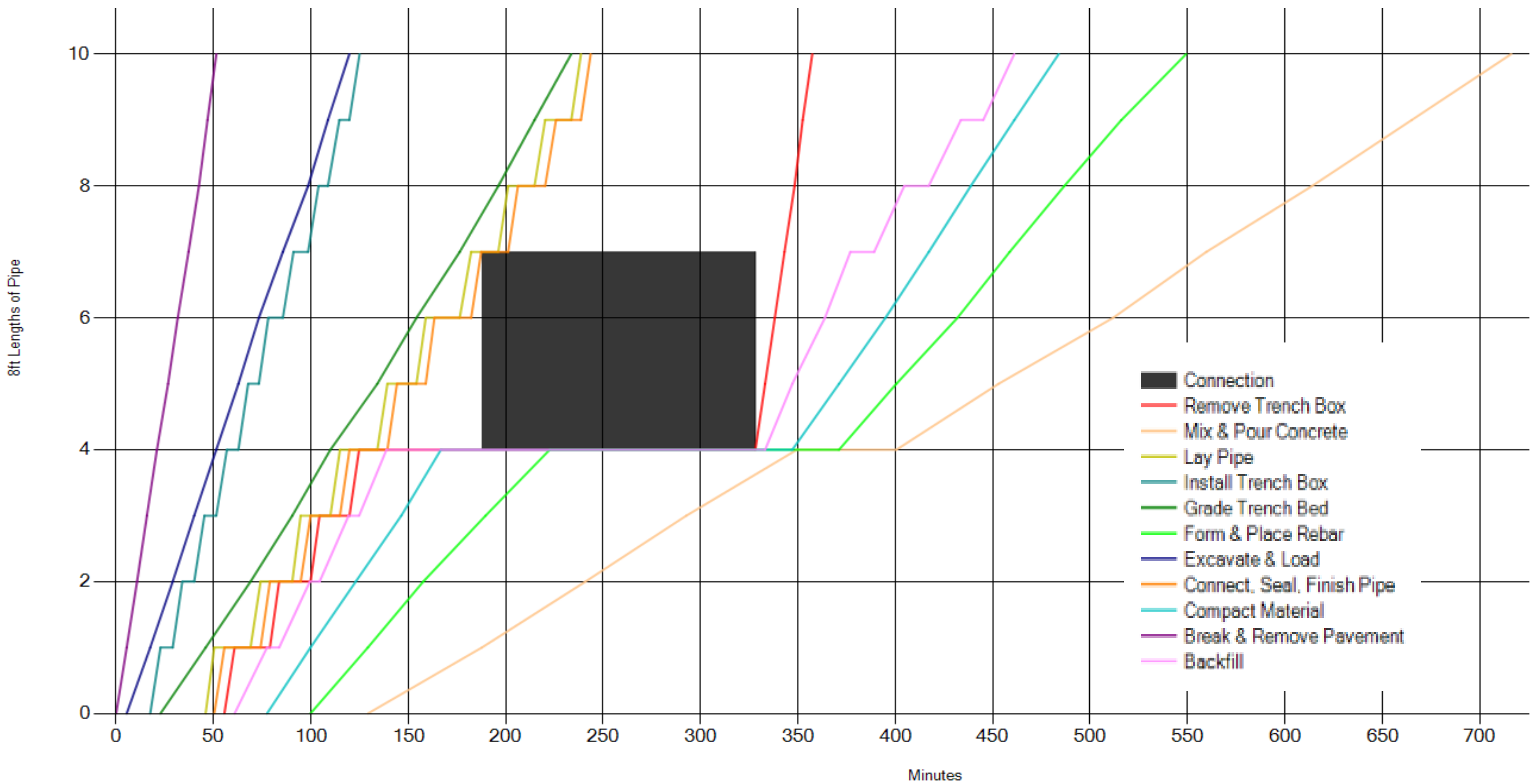


Figure 5-10 Sewer Line Schedule: Discrete Activity

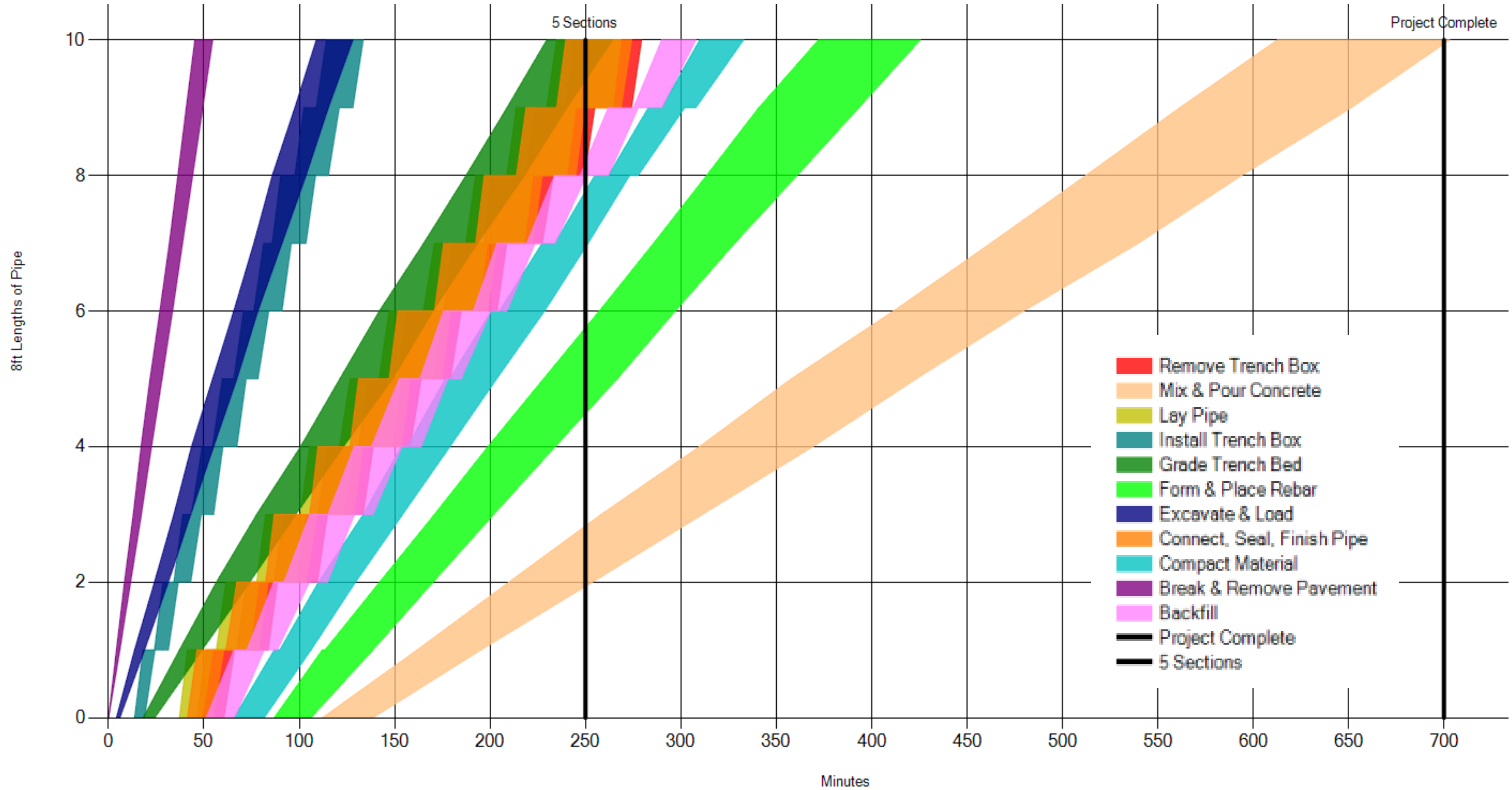


Figure 5-11 Sewer Line Schedule: Milestones

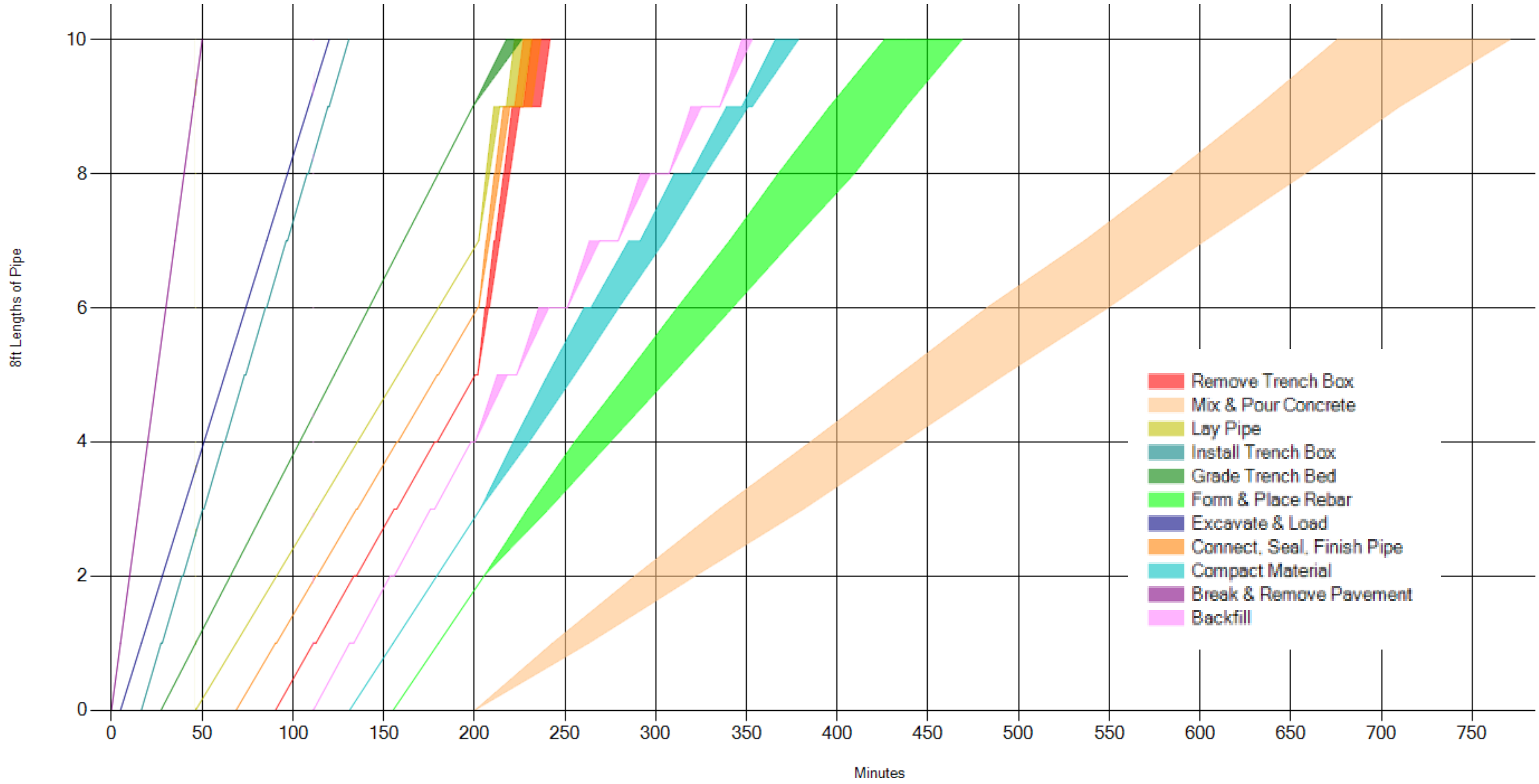


Figure 5-12 Sewer Line Schedule: Progress/As-Built

6.0 CASE STUDY

To further show the applications of the developed template, it will be applied to a real world case study. This chapter presents the use of the linear scheduling template to model a real world case study of the construction of a mechanically stabilized earth (MSE) wall. Section 6.1 introduces the project. Section 6.2 provides an overview of the construction method used to build the wall. Section 6.3 presents the methods of data collection used. Section 6.4 explains the simulation model created using the linear scheduling special purpose template. Section 6.5 analyzes the results of the simulation model and compares those results to the real world results.

6.1 Introduction and Background

As a part of an oil sands project in northern Alberta, Ledcor was contracted to build two MSE walls. This was the first time that Ledcor Site Services Ltd. had undertaken such major MSE structures and a steep learning curve was encountered. The construction of both walls was significantly delayed. After speaking with the engineers responsible for the project, it was determined that three major issues were the cause of this delay. The first issue was material delivery as there was no approved source of lean oil sand (LOS) when the project began. In addition, as the project was delayed into the winter, material would freeze before being placed and would have to be removed and placed again. The second issue was rework encountered several times as reinforcing straps were placed improperly resulting in the removal of the straps and re-installation at the correct location. The third and most significant issue was that the decision making process was hindered due to inconsistent performance, constant changes and other unforeseen delays. The engineers on site had difficulty determining how far along a lift installation needed to progress before the next lift could begin.

6.2 Mechanically Stabilized Earth (MSE) Wall Construction

The construction of a MSE wall begins with the installation of the shear key which is essentially a hole dug where the wall is to be constructed and then filled with competent material. Next drainage trenches are dug and lined with geo-fabric before perforated pipe is laid. The trenches are then backfilled with gravel. At the same time as the drainage is being installed, a concrete slab is poured at the front of the wall. The slab which is referred to as the levelling slab has a very high tolerance and will support the concrete panels that are installed at the bottom of the wall. After the levelling slab a layer of sand, referred to as the drainage blanket is laid in 8 inch lifts. Laying a lift involves material being dumped, spread, wetted by a water truck and then compacted by a roller. Stabilizing straps are installed between each layer and the process repeats for the next lift. Surveyors must check the alignment of the straps before the next lift goes in. In addition, the soil is tested and must be approved after a certain volume is installed. While the drainage blanket is installed concrete panels are placed on top of the leveling slab. The panels are braced so that they don`t move when the drainage blanket is compacted.

After the drainage blanket and concrete panels are installed, the construction process changes slightly as the wall is built up in 0.5 meter lifts. Mesh baskets are installed along the front of the wall. The baskets are tied to one another with steel wire and a woven geo-tech fabric is placed between the soil and the baskets. The straps are then attached to the baskets. Once the straps have been surveyed and approved, a layer of LOS is dumped, spread, wetted and compacted. The LOS layer reaches from the back of the wall up to 2 meters away from the baskets. The last two meters is referred to as the chimney drain and is a layer of sand that is also dumped, spread, wetted and compacted. The chimney drain must be compacted with plate tampers as a roller would cause the baskets to deform. Once the chimney drain is installed, the next lift can begin. This process is shown in Figure 6.1.

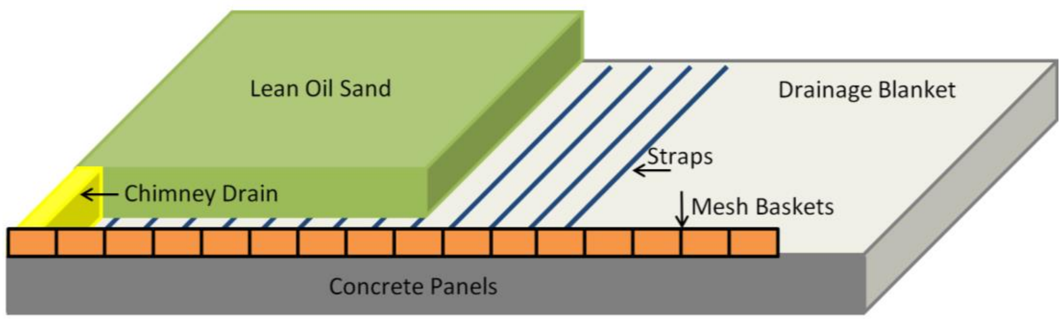


Figure 6-1 MSE Wall Construction Process

Finally, a 1 meter layer of sand is placed and watered, followed by a 1.1 meter layer of gravel and a final 0.3 meter layer of finer gravel. To finish off the wall, concrete approach slabs are poured, a maintenance slab is poured and electrical conduits are installed. A chain link safety fence is put in place and 797 truck tires are placed at the edge of the wall, two high.

6.3 Data Collection

Initially, the data required to model the construction of one MSE wall was to be obtained from video recordings of site. Four camera systems composed of a camera and network video recorder (NVR) mounted on a light tower were installed on site. Due to site restrictions, no live streaming was allowed. The videos had to be downloaded manually and project control staff on site were instructed how to do so.

There were numerous issues encountered with the camera systems. The NVR's were powered by the light tower that they were mounted on. Every time the towers had to be moved or fueled they were shut off, which in turn, shut of the NVR. When the light tower was turned back on, the NVR did not automatically turn back on. Also, the cameras drew their power from the NVR through power over Ethernet (PoE). As it got colder outside, the PoE capability slowly stopped working and the cameras drooped until they were pointing directly at the ground. Because of these technical issues and the inability of site staff to download the videos daily due to time constraints, no useable videos were obtained.

In lieu of the videos, drawings and construction records were provided that contained the quantity installed, man hours spent and percent complete for installing the baskets and straps as well as for placing and compacting sand, gravel and LOS. The data provided only allows for the modelling of the construction process after the concrete panels and drainage blanket were installed. It does not allow for modelling of the effects of material delivery delays or delays due to rework (i.e. relaying the straps).

6.4 Simulation Models

The issue with creating a linear schedule for an MSE wall is that the wall is linear in both the horizontal and vertical direction. In the horizontal direction, the construction process is much like that of a highway project; placing and compacting material. In the vertical direction, the construction process is much like that of a high rise building; lift one must be built before lift two, which must be built before lift three etc. The question then becomes should lifts be used as the unit of construction (vertical) or should meters along the wall be used (horizontal). Both models are presented below.

6.4.1 Vertical Construction – Lifts as the Unit

The model created for vertical construction can be seen in Figure 6-2.

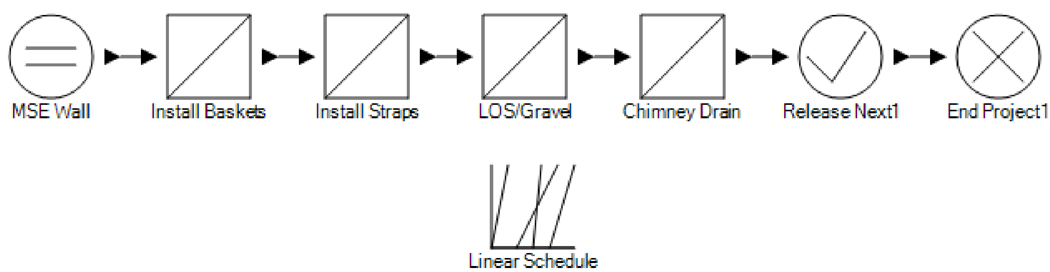


Figure 6-2 Vertical Construction Model

The create project element (MSE Wall) creates the 32 lifts and releases the first lift for construction. The baskets are installed on the first lift, the straps are installed, the LOS is placed and the chimney drain is placed. Then, lift one passes through the release next element which releases the second lift from the create project element and the process repeats itself until all 32 lifts are completed. The activity durations (hours/lift) for each of the four continuous activities were determined from the construction records. It was assumed that if a crew could complete a lift in 60 hours or in 90 hours they could complete a lift at any time between those two values. Because of this assumption and the lack of data, uniform distributions were used to model the activity durations for the stochastic model. For the deterministic model, the average activity duration was used.

6.4.2 Horizontal Construction – Meters as the Unit

The model created for horizontal construction can be seen in Figure 6-3.

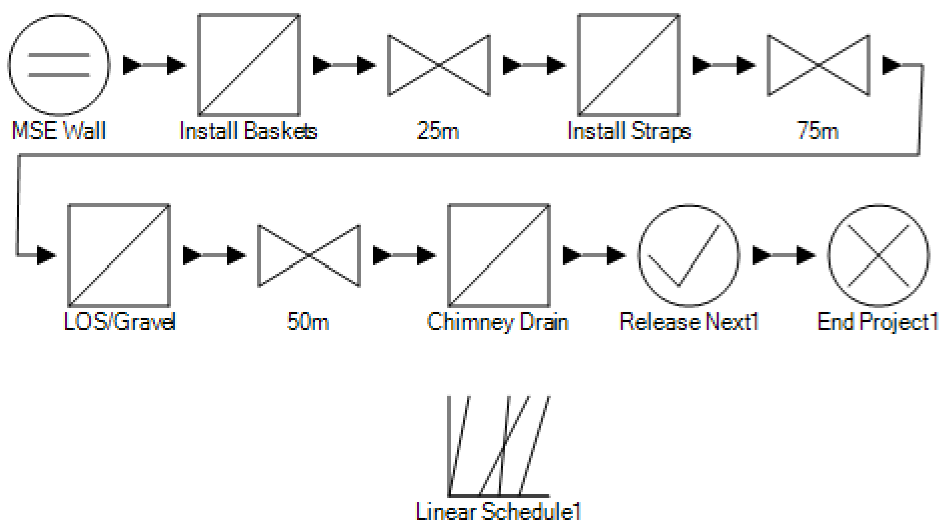


Figure 6-3 Horizontal Construction Model

The create project element (MSE Wall) creates the 6884 horizontal meters (determined from the drawings) and releases the first 283 meters (length of the first lift) for construction. Each meter has the baskets installed, the straps installed, the LOS/gravel placed and the sand placed before it releases the next queued meter for construction. For example when meter 1 reaches the release next element, the create element will release meter 284, which represents the first meter in the second lift. There is a space buffer of 25 meters between installing the baskets and installing the straps as both are done by ground crews and a significant amount of space is not required. Ledcor uses a 50-10 rule for approaching equipment which means that no one can approach closer than 50 meters without radio contact/visual communication and within 10m the equipment operator must be out of the cab. Because the crew installing the straps is a ground crew and the crew placing/compacting the LOS/gravel is an equipment crew, a buffer of 75 meters was used between the two for added protection. The buffer between placing the LOS/gravel and placing the chimney drain is 50 meters as both crews would be equipment crews. Once again the activity durations (hours/meter) for each of the four continuous activities were determined from the construction records. This model is a deterministic model due to the limited information in the construction records.

6.5 Results and Analysis

The data received indicated that the construction took place over 27 weeks or 4536 hours (7 days/week, 24 hours/day). The deterministic vertical construction model indicates that if each lift was completely installed before the next lift began, the project would have taken 8064 hours or 48 weeks to complete. The horizontal construction model indicates that if each activity on a lift begins after a certain buffer length, the project would have taken 3796 hours or 22.5 weeks to complete. The deterministic vertical and horizontal linear schedules are shown in Figure 6-4 and Figure 6-5 respectively.

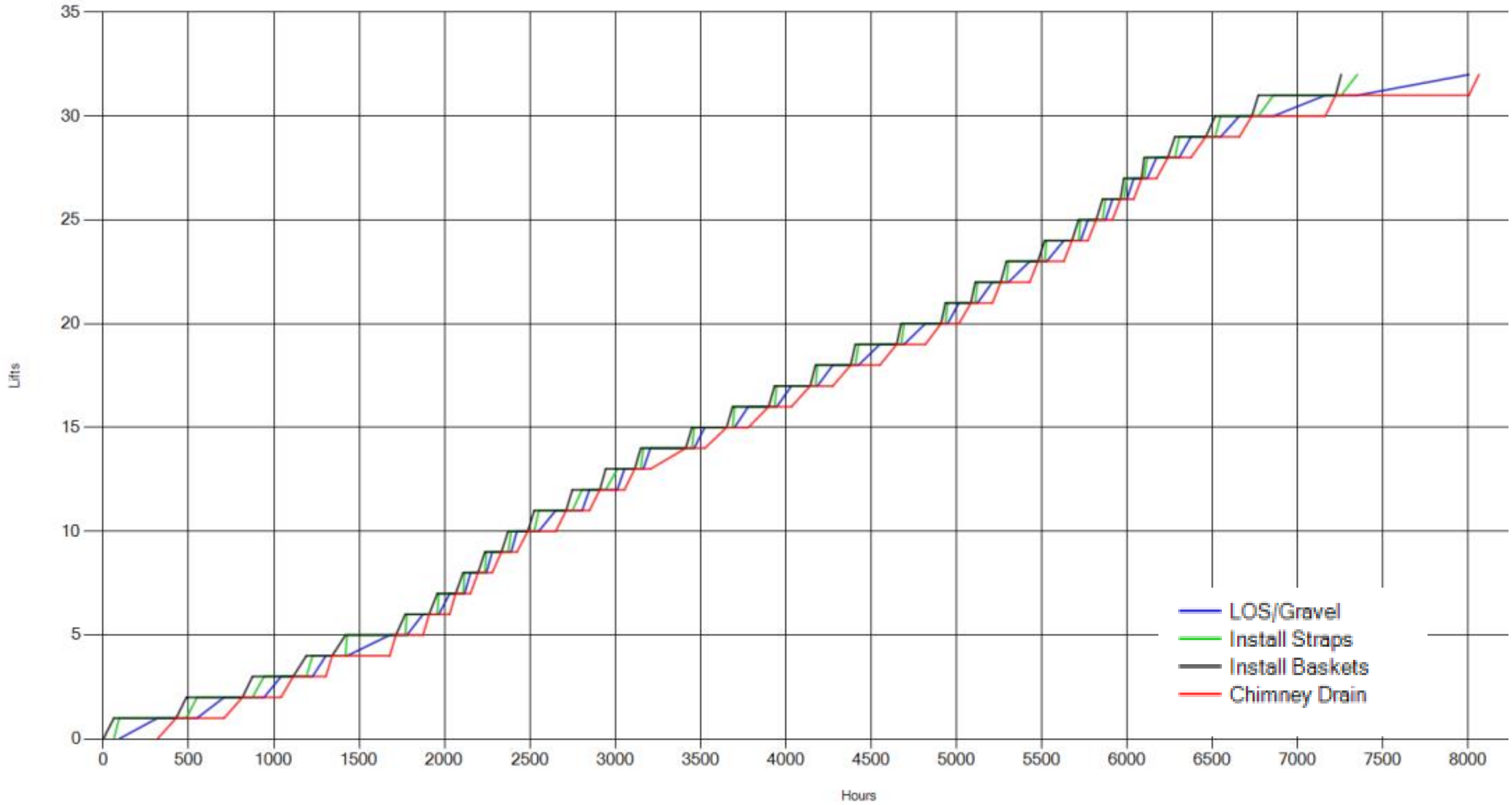


Figure 6-4 Vertical Construction – Lifts as the Unit

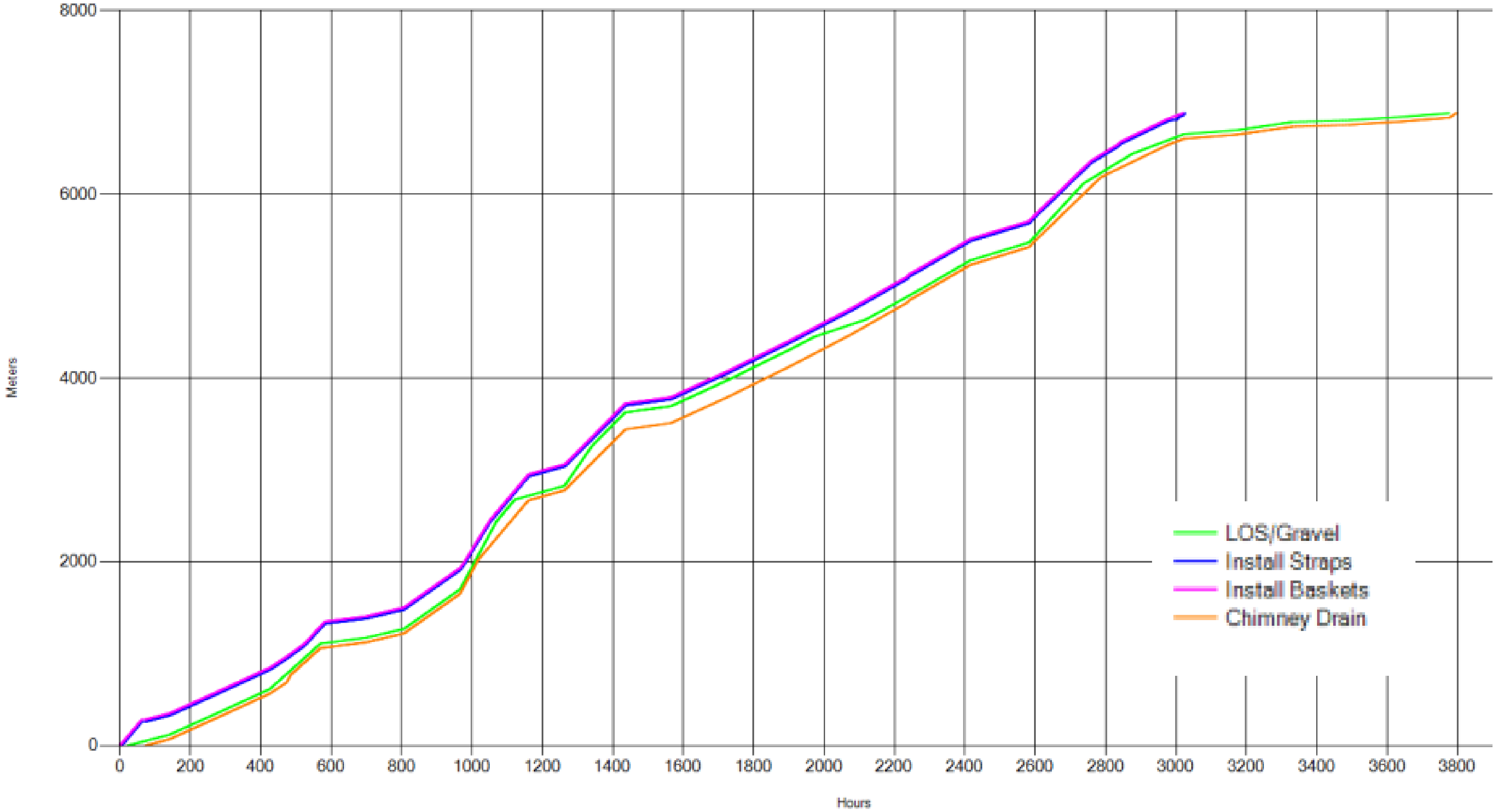


Figure 6-5 Horizontal Construction – Meters as the Unit

It can be seen on the vertical construction linear schedule that waiting for one lift to be finished before the next begins, results in constant work interruption (horizontal segments of activity lines) and extended project duration. In addition, when the stochastic model was simulated 1000 times, the overall duration varied from 6214 hours (37 weeks) to 10446 hours (62 weeks). Figure 6-6 shows the resultant cumulative linear schedule of the stochastic model for the first five lifts.

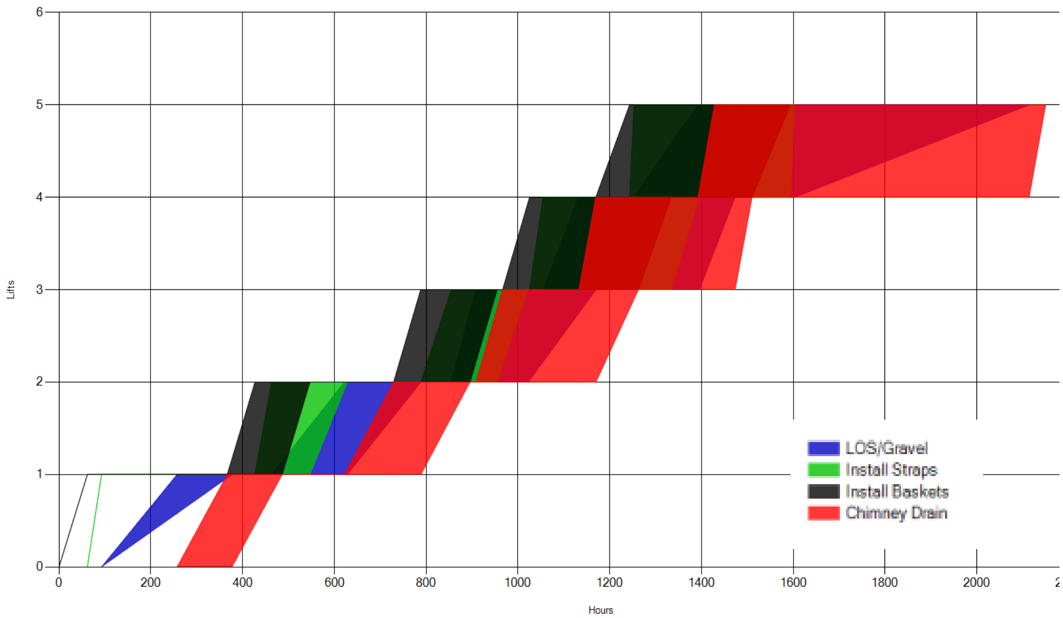


Figure 6-6 Cumulative Linear Schedule

Looking at the cumulative linear schedule, it becomes immediately apparent that if any activity is delayed, it will impact the succeeding activities, making all activities critical. Being able to model this project ahead of time would allow the project managers to see the work continuity issues with completing a lift before starting the next one, as well as the criticality of all of the activities.

The horizontal construction schedule provides much better work continuity. The actual duration of the project was closer to the horizontal schedule than the vertical schedule, which is expected as the staff on site used their best judgment as to when to begin the next lift. If the staff had access to this tool at the beginning of the project, the above analysis could have been performed and the timing of the lifts would have been known.

7.0 CONCLUSIONS AND RECCOMENDATIONS

This chapter provides a summary of the work conducted. Section 6.1 presents the conclusions of this research. Section 6.2 provides recommendations for future template development.

7.1 Conclusions

As construction companies are becoming more and more specialized, the need for specialized scheduling software is growing, especially for repetitive construction projects. Although simulation has been widely accepted in the research community, industry has been slow to accept simulation as a useful tool in modelling construction operations. This is mostly due to the complexity of modelling construction operations with general purpose simulation tools and the time required to train staff on how to use various simulation languages. The goal of this research was the development of a framework for repetitive scheduling that could be used to develop a tool that met industry requirements while maintaining simplicity.

In order to develop this framework, a thorough review of the development of the linear scheduling method was conducted. It was found that linear scheduling has various advantages over network methods when scheduling repetitive projects. These include the ability to show the interdependencies of activities, the ease of which they are understood by all levels of staff on site, the ability to monitor progress and work continuity, and the simplicity of preparing and updating the schedule. Although the obvious benefits, there was one major disadvantage uncovered; to date, very little has been done to incorporate uncertainty into linear scheduling. Construction operations almost always include an element of uncertainty and therefore it is crucial to be able to model the effects of this uncertainty on the schedule.

A review of current linear scheduling software was conducted to determine the strengths and weaknesses of existing programs. It was found that although some programs were quite comprehensive and produced a variety of useful visual tools, almost all of them did not incorporate uncertainty. The few that did either limited the distributions that could be modelled or the models took an extended amount of time to run.

The framework developed incorporated the industry needs identified from the literature and the gaps in existing repetitive scheduling software. This framework was used to develop a special purpose simulation tool for Simphony. The template developed consists of thirteen modelling elements and allows the user to:

1. Model activities with production rates equal to the natural rhythm and explore various options if acceleration of the schedule is required
2. Model minimum time and space requirements between activities
3. Model the effect of material requirements and limitations
4. Model milestones and explore various options if milestones are not met
5. Model both continuous and discrete activities
6. Visually identify critical activities on the linear schedule that incorporates the results of all stochastic runs

7. Model varying production rates for the same activity, which allows the user to incorporate a learning curve if desired
8. Produce a linear schedule that allows the transparency of the lines to be changed to make overlapping activities easier to see
9. Model uncertainty in all durations
10. Model both horizontal and vertical repetitive construction operations
11. Detect errors in logic within the model and aid in resolving the issue

The developed model was verified using a fictional sewer line case study. The sewer line case study was further used to demonstrate the applications of the template. After the template was verified, it was used to model the construction of a real life MSE wall case study. Due to issues encountered in data collection, the model did not include the entire construction process or model material delivery. It did, however, show the benefits of scheduling the MSE wall as a horizontal project with space buffers over scheduling the wall as a vertical project.

7.2 Future Recommendations

The template developed is intended as the first step in the development of an industry accepted tool for linear scheduling. The following are recommendations for future developments:

1. Create a person/equipment pool element – This will allow the template to more accurately model real world scenarios by modelling individual resources rather than a combination. The user should be allowed to define a crew/equipment pool and then define which resources each task requires.

A priority input could be utilized if more than one task requires the same resources

2. Include resource histograms – For both material and crew/equipment, histograms would assist managers in keeping track of what/who should be where and when
3. Convert the time axis to dates and incorporate a calendar – The user should be allowed to specify generic dates (how the template outputs now) or calendar dates. If calendar dates are chosen, the user will specify a start date and a calendar (i.e. Monday to Friday, 24hrs/day etc.) and the linear schedule will display the actual calendar dates on the time axis
4. Schedule screenshots – The user should be able to view the entire linear schedule, or have the option to view certain ranges of it (i.e. units 1-5 or day 300-350)
5. Computer calculated learning curve – Although the user is able to apply their own learning curve by entering different durations for different units, it may be advantageous to allow the user to specify a learning curve and have the template apply it. This would help reduce user error
6. Auto highlight critical areas – Although the user can visually identify activity overlap, it would be very useful if the template itself located areas where minimum and maximum durations overlap, or space/time buffers may not be met when using stochastic durations
7. Enhance the counter element – Statistics, such as productivity can be added to the counter to assist the user in the decision making process

8. Integrate cost data – This would allow the user to analyze various options on more than just a time basis
9. Optimization – Currently, the template can be used to manually identify the optimal schedule. An optimization routine that allows the user to choose which property they wish to optimize (time, cost, work continuity etc.) would be useful.
10. Random events – To increase the real world applications of the template the effects of events such as breakdowns and weather on the schedule should be modelled.

8.0 REFERENCES

AbouRizk, S. (2010). "Role of Simulation in Construction Engineering and Management." *J.Constr.Eng.Manage.*, 136(10), 1140-1153.

AbouRizk, S., and Halpin, D. (1990). "Probabilistic Simulation Studies for Repetitive Construction Processes." *J.Constr.Eng.Manage.*, 116(4), 575-594.

AbouRizk, S., Halpin, D., Mohamed, Y., and Hermann, U. (2011). "Research in Modeling and Simulation for Improving Construction Engineering Operations." *J.Constr.Eng.Manage.*, 137(10), 843-852.

AbouRizk, S., and Mohamed, Y. (2000). "Simphony - An Integrated Environment for Construction Simulation." *Proceedings of the 2000 Winter Simulation Conference,* 1907-1914.

AbouRizk, S. M., and Hajjar, D. (1998). "A Framework for Applying Simulation in Construction." *Can.J.Civ.Eng.,* 25(3), 604-617.

Al Sarraj, Z. (1990). "Formal Development of Line-of-Balance Technique." *J.Constr.Eng.Manage.*, 116(4), 689-704.

Arditi, D. (1985). "Construction Productivity Improvement." *J.Constr.Eng.Manage.*, 111(1), 1-14.

Arditi, D., and Albulak, M. (1986). "Line-of-Balance Scheduling in Pavement Construction." *J.Constr.Eng.Manage.*, 112(3), 411-424.

Arditi, D., and Albulak, M. (1979). "Comparison of Network Analysis with Line of Balance in a Linear Repetitve Construction Project." *Proceedings of the 6th Internet Congress 1979,* Garmisch-Partenkirchen, Munchen, Germany, 13-25.

Arditi, D., and Psarros, M. (1987). "SYRUS - System for Repetitive Unit Scheduling." *Proc., Nordnet/Internet/PMI'87 Conference on Project Management*, The Icelandic Project Management Society, 55-61.

Arditi, D., Tokdemir, O. B., and Suh, K. (2001). "Scheduling system for repetitive unit construction using line-of-balance technology." *Engineering, Construction and Architectural Management*, 8(2), 90-103.

Arditi, D., Tokdemir, O., and Suh, K. (2002). "Challenges in Line-of-Balance Scheduling." *J.Constr.Eng.Manage.*, 128(6), 545-556.

Ashley, D. B. (1980). "Simulation of Repetitive-Unit Construction." *Journal of the Construction Division*, 106(CO2), 185-194.

Birrell, G. S. (1980). "Construction Planning - Beyond the Critical Path." *Journal of the Construction Division,* 106(CO3), 389-407.

Carr, R. I., and Meyer, W. L. (1974). "Planning Construction of Repetitive Building Units." *Journal of the Construction Division,* 100(CO3), 403-412.

Chrzanowski, E., and Johnston, D. (1986). "Application of Linear Scheduling." *J.Constr.Eng.Manage.,* 112(4), 476-491.

Crandall, K. C. (1976). "Probabilistic Time Scheduling." *Journal of the Construction Division,* 102(CO3), 415-423.

Dressler, J. (1974). "Stochastic Scheduling of Linear Construction Sites." *Journal of the Construction Division,* 100(CO4), 571-587.

Duffy, G. (2009). "Linear Scheduling of Pipeline Construction Projects With Varying Production Rates". Doctor of Philosophy. Oklahoma State University, Stillwater, Oklahoma.

Duffy, G., Oberlender, G., and Seok Jeong, D. (2011). "Linear Scheduling Model with Varying Production Rates." *J.Constr.Eng.Manage.,* 137(8), 574-582.

Ekyalimpa, R., AbouRizk, S., and Farrar, J. (2012). "Effective Strategies for Simulating One-of-a-Kind Construction Projects." *Proceedings of the Winter Simulation Conference,* Winter Simulation Conference, 260:1-260:13.

El-Sayegh, S. M. (1998). "Linear Construction Planning Model (LCPM): A New Model for Planning and Scheduling Linear Construction Projects". Doctor of Philosophy. Texas A&M University, .

Halpin, D. W., and Riggs, L. S. (1992). *Planning and Analysis of Construction Operations.* John Wiley & Sons, Inc., .

Harmelink, D. J. (1995). "Linear Scheduling Model: The Development of a Linear Scheduling Model With Micro Computer Applications for Highway Construction Project Control". Doctor of Philosophy. Iowa State University, Ames, Iowa.

Harmelink, D. J., and Yamin, R. (2001). "Development and Application of Linear Scheduling Techniques to Highway Construction Projects." *Rep. No. FHWA/IN/JTRP-2000/21,* Joint Transportation Research Program, Indiana Department of Transportation and Purdue University, West Lafayette, Indiana.

Harmelink, D., and Rowings, J. (1998). "Linear Scheduling Model: Development of Controlling Activity Path." *J.Constr.Eng.Manage.,* 124(4), 263-268.

Harris, R., and Ioannou, P. (1998). "Scheduling Projects with Repeating Activities." *J.Constr.Eng.Manage.,* 124(4), 269-278.

Ipsilandis, P. (2007). "Multiobjective Linear Programming Model for Scheduling Linear Repetitive Projects." *J.Constr.Eng.Manage.,* 133(6), 417-424.

Johnston, D. (1981). "Linear Scheduling Method for Highway Construction." *Journal of the Construction Division,* 107(CO2), 247-261.

Kavanagh, D. (1985). "Siren: A Repetitive Construction Simulation Model." *J.Constr.Eng.Manage.,* 111(3), 308-323.

Liu, W., Flood, I., and Issa, R. R. A. (2005). "Simulation and Optimization of Linear Construction Projects." *International Conference on Computing in Civil Engineering,* ASCE, Cancun, Mexico, 1-11.

Lu, M. (2003). "Simplified Discrete-Event Simulation Approach for Construction Simulation." *J.Constr.Eng.Manage.,* 129 537-546.

Lutz, J. D. (1990). "Planning of Linear Construction Projects Using Simulation and Line of Balance". Doctor of Philosphy. Purdue University, West Lafayette, Indiana.

Mattila, K., and Abraham, D. (1998). "Linear Scheduling: Past Research Efforts and Future Directions." *Engineering Construction and Architectural Management,* 5(3), 294-303.

Mattila, K., and Park, A. (2003). "Comparison of Linear Scheduling Model and Repetitive Scheduling Method." *J.Constr.Eng.Manage.,* 129(1), 56-64.

Moselhi, O., and El-Rayes, K. (1993). "Scheduling of Repetitive Projects with Cost Optimization." *J.Constr.Eng.Manage.,* 119(4), 681-697.

Moselhi, O., and Hassanein, A. (2003). "Optimized Scheduling of Linear Projects." *J.Constr.Eng.Manage.,* 129(6), 664-673.

Naaman, A. E. (1974). "Networking Methods for Project Planning and Control." *Journal of the Construction Division,* 100(CO3), 357-372.

O'Brien, J. (1975). "VPM Scheduling for HIgh-Rise Buildings." *Journal of the Construction Division,* 101(CO4), 895-905.

Peer, S. (1974). "Network Analysis and Construction Planning." *Journal of the Construction Division,* 100(CO1), 203-210.

Reda, R. (1990). "RPM: Repetitive Project Modeling." *J.Constr.Eng.Manage.,* 116(2), 316-330.

Russell, A. (1990). "REPCON: An Innovative Construction Managment System." *Managing Projects (Int. Symp. on Building Economics and Constr. Mgmt),* International Council For Building Reseach Studies and Documentation, Australia, 405-417.

Russell, A., and Caselton, W. (1988). "Extensions to Linear Scheduling Optimization." *J.Constr.Eng.Manage.,* 114(1), 36-52.

Russell, A., and Wong, W. (1993). "New Generation of Planning Structures." *J.Constr.Eng.Manage.,* 119(2), 196-214.

Russell, A. D. (1985). "Microcomputers, Management, and High-rise Construction: The Next Step." *Can.J.Civ.Eng.,* 12(2), 396-414.

Sabha, F. (2012). "Modeling Reclamation Earthwork Operations Using Special Purpose Simulation Tool". University of Alberta, .

Selinger, S. (1980). "Construction Planning for Linear Projects." *Journal of the Construction Division,* 106(CO2), 195-205.

Senouci, A., and Eldin, N. (1996). "Dynamic Programming Approach to Scheduling of Nonserial Linear Project." *J.Comput.Civ.Eng.,* 10(2), 106-114.

Stradal, O., and Cacha, J. (1982). "Time Space Scheduling Method." *Journal of the Construction Division,* 108(CO3), 445-457.

Suh, K. (1993). "A Scheduling System for Repetitive-Unit Construction Using Line-Of-Balance Technology". Doctor of Philosophy in Civil Engineering. Illinois Institute of Technology, Chicago, Illinois.

Tavakoli, A., and Riachi, R. (1990). "CPM Use in ENR Top 400 Contractors." *J.Manage.Eng.,* 6(3), 282-295.

Thabet, W., and Beliveau, Y. (1997). "SCaRC: Space-Constrained Resource-Constrained Scheduling System." *J.Comput.Civ.Eng.,* 11(1), 48-59.

Tokdemir, O. B. (2003). "ALISS: Advanced Linear Scheduling System". Doctor of Philosophy in Civil Engineering. Illinois Institute of Technology, Chicago, Illinois.

Trofin, I. (2004). "Impact of Uncertainty on Construction Project Performance Using Linear Scheduling". Master of Science. University of Florida, Florida.

Vorster, M., and Bafna, T. (1992). "Discussion of “Formal Development of Line-of-Balance Technique” by Zohair M. Al Sarraj (December, 1990, Vol. 116, No. 4)." *J.Constr.Eng.Manage.*, 118(1), 210-211.

Wang, C., and Huang, Y. (1998). "Controlling Activity Interval Times in LOB Scheduling." *Constr.Manage.Econ.*, 16(1), 5-16.

Yamin, R. (2001). "Cumulative Effect of Productivity Rates in Linear Schedules". Doctor of Philosophy. Purdue University, .

Yamín, R., and Harmelink, D. (2001). "Comparison of Linear Scheduling Model (LSM) and Critical Path Method (CPM)." *J.Constr.Eng.Manage.*, 127(5), 374-381.

APPENDIX: USER MANUAL

The Linear Scheduling Template is a special purpose simulation tool that allows the user to model and produce the linear schedule for repetitive construction projects. The use of this template does not require knowledge of simulation language, however, working knowledge of the linear scheduling method and the construction process being modeled are recommended. The thirteen modelling elements that make up the Linear Scheduling Template will be described as follows:

Element Name

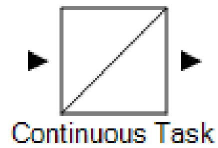
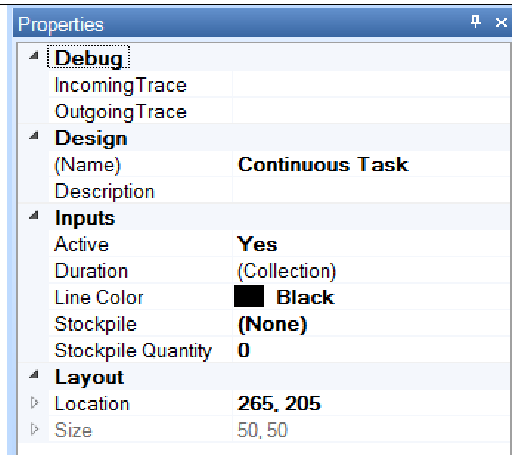
Descriptions

Element symbol	The symbol that represents the element. Each element shape is unique.
Properties	The property grid contains all of the input and output parameters. The user can specify values of various input variables and the simulator will display all outputs after the simulation
Input Parameters	The inputs that the user can specify.
Output Parameters	The outputs the simulation displays.

TASK ELEMENTS

Continuous Task

The continuous task element is used to represent repetitive activities in the project. The duration of the activity can be uniform or vary for all of the units in the project. Each active continuous task element will appear on the linear schedule.

Element symbol	
Properties	
Input Parameters	*Active* – Should the activity be included in the current simulation and plotted on the linear schedule *Duration* – The length of time it takes to complete the activity on a single unit within a specified range. Durations can be either constant values or distributions. Figure A-1 shows the input window for the duration property *Line Color* – The color of the activity line on the linear schedule. All tasks must have a different line color unless they are black *Stockpile* – The stockpile from which the activity requires resources. This property is optional *Stockpile Quantity* – The amount of resources required by the task to complete a single unit
Output Parameters	N/A

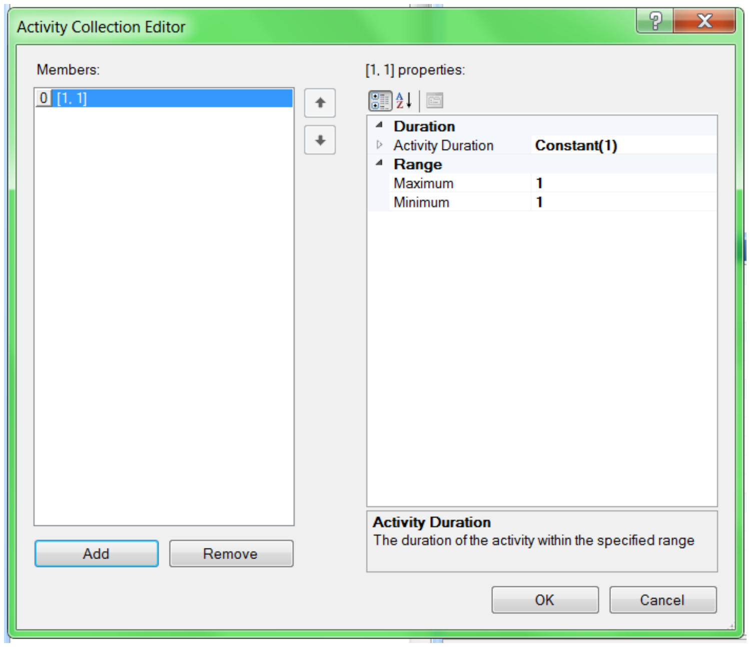
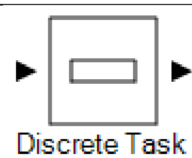
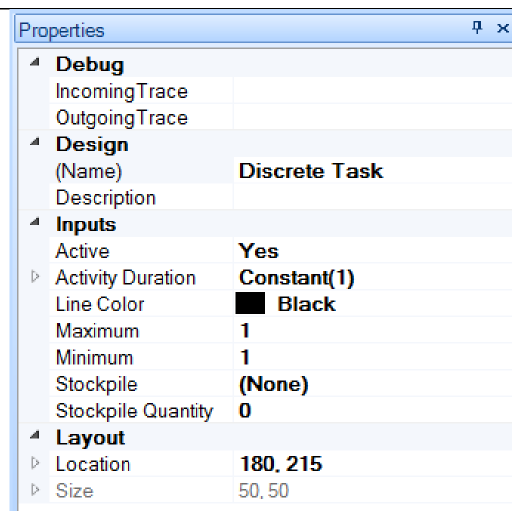


Figure A-1 Continuous Task Duration Property Input

Discrete Task

The discrete task element is used to represent one off activities in the project. These activities typically occupy more than one unit/location at a time. Each active discrete task element will appear on the linear schedule.

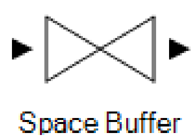
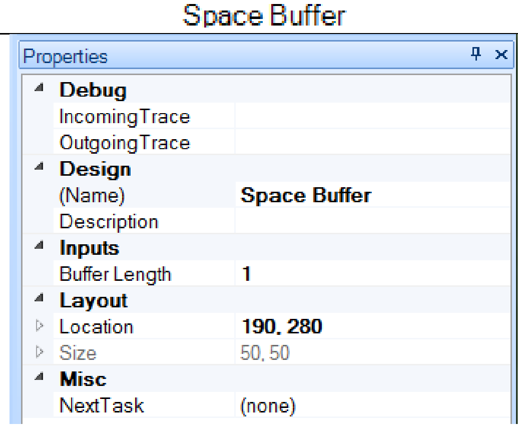
Element symbol	
Properties	
Input Parameters	*Active* – Should the activity be included in the current simulation and plotted on the linear schedule

	Activity Duration – The length of time it takes to complete the activity. Duration can be either a constant value or a distribution *Line Color* – The color of the activity line on the linear schedule. All tasks must have a different line color unless they are black *Maximum* – The maximum of the range of units this task occupies *Minimum* – The minimum of the range of units this task occupies *Stockpile* – The stockpile from which the activity requires resources. This property is optional *Stockpile Quantity* – The amount of resources required to complete the task
Output Parameters	N/A

BUFFER ELEMENTS

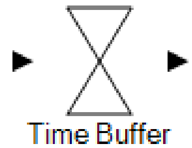
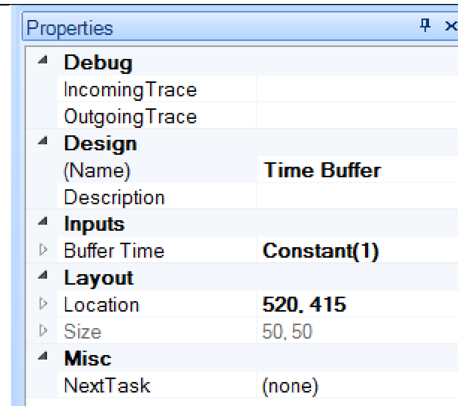
Space Buffer

The space buffer element models minimum space dependencies between activities.

Element symbol	
Properties	
Input Parameters	*Buffer Length* – The length of space required between two activities. Must be in the same units as the units of the project
Output Parameters	N/A

Time Buffer

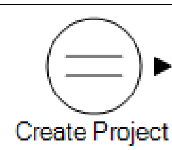
The time buffer element models minimum time dependencies between activities. It also takes into consideration the duration of the preceding activity.

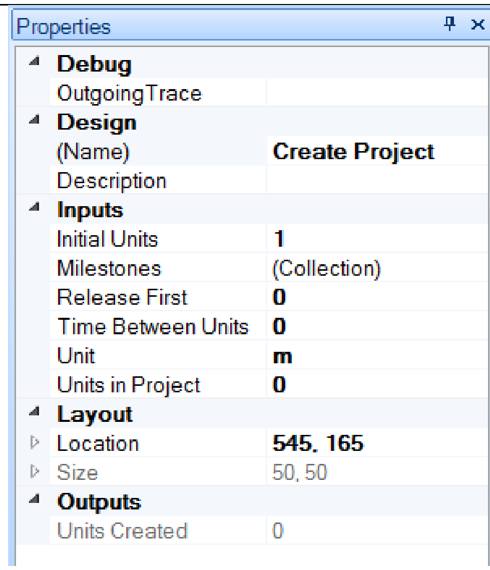
Element symbol	
Properties	
Input Parameters	*Buffer Time* – The length of time required between two activities. Can be constant or a distribution
Output Parameters	N/A

PROJECT ELEMENTS

Create Project

The create project element is where the project is initialized and the milestones are defined. There can only be one create project element in a model.

Element symbol	

Properties	
Input Parameters	*Initial Units* – The number of units available for construction at the start of the project *Milestones* – The name of a time which a milestone occurs can be inputted. Milestones are optional. Figure A-2 shows the input window for milestones
Output Parameters	*Units Created* – The number of units the create project has created at the time the simulation stops

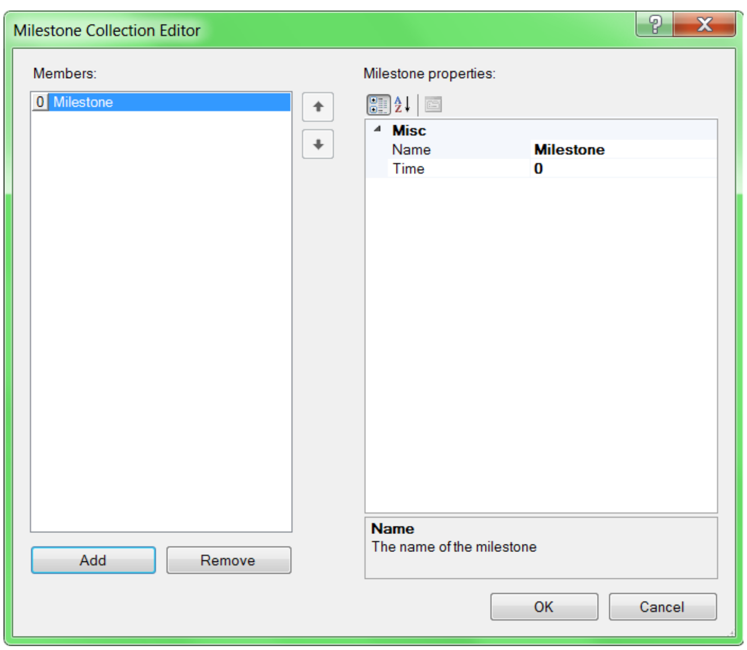
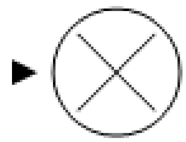


Figure A-2 Milestones Input Screen

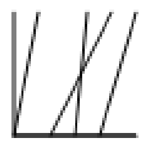
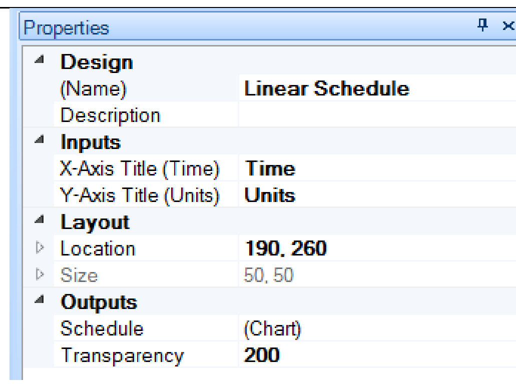
End Project

The end project element signifies the end of the project. There can only be one end project element in the project and all paths through the project must end at the end project element.

Element symbol	
Properties	N/A
Input Parameters	N/A
Output Parameters	N/A

Linear Schedule

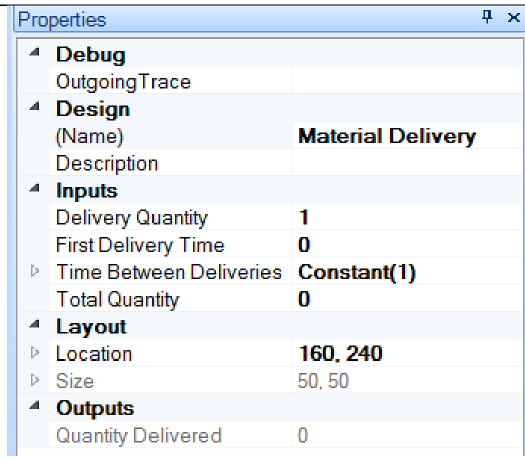
The linear schedule element, when double clicked, displays the schedule.

Element symbol	
Properties	
Input Parameters	*X-Axis Title* – The x-axis title that will be displayed on the schedule *Y-Axis Title* – The y-axis title that will be displayed on the schedule
Output Parameters	*Schedule* – The linear schedule produced by the simulation *Transparency* – The transparency of the lines on the schedule. Adjusting this property can assist in making the schedule easier to interpret

RESOURCE ELEMENTS

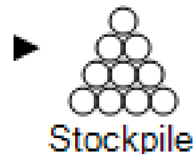
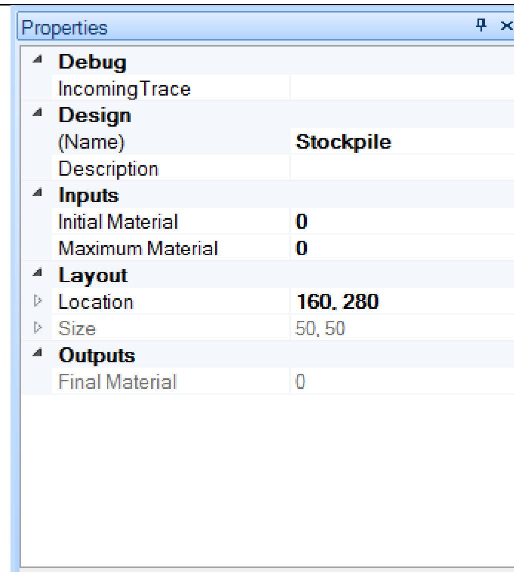
Material Delivery

The material delivery element models the delivery of consumable resources to the project.

<table>
<tr><td>Element symbol</td><td>
</td></tr>
<tr><td>Properties</td><td>
</td></tr>
<tr><td>Input Parameters</td><td>Delivery Quantity – The number of resources delivered at one time
First Delivery Time – The time at which the first delivery is sent
Time Between Deliveries – The inter-arrival time of deliveries. Can be constant or a distribution
Total Quantity – The total number of resources to be delivered to site</td></tr>
<tr><td>Output Parameters</td><td>Quantity Delivered – The number of resources delivered to site at the time the simulation stops</td></tr>
</table>

Stockpile

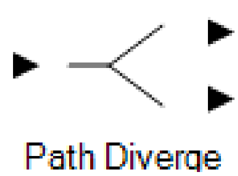
The stockpile element models space requirements on site.

<table>
<tr><td>Element symbol</td><td>
</td></tr>
<tr><td>Properties</td><td>
</td></tr>
<tr><td>Input Parameters</td><td>Initial Material – The initial number of resources on site
Maximum Material – The maximum number of resources allowed on site. If zero, there are no restrictions</td></tr>
<tr><td>Output Parameters</td><td>Final Material – The number of resources left in the stockpile at the time the simulation stops</td></tr>
</table>

OTHER ELEMENTS

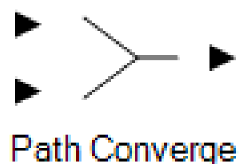
Path Diverge

The path diverge element allows the user to model tasks that occur simultaneously. It can also assist the user in what if analysis by allowing for variations in the scheduled to be modelled simultaneously.

<table>
<tr><td>Element symbol</td><td>
</td></tr>
<tr><td>Properties</td><td>N/A</td></tr>
<tr><td>Input Parameters</td><td>N/A</td></tr>
<tr><td>Output Parameters</td><td>N/A</td></tr>
</table>

Path Converge

The path converge element allows the user to model tasks that occur simultaneously. There must be an equal number of path converge and diverge elements in the model.

Element symbol	
Properties	N/A
Input Parameters	N/A
Output Parameters	N/A

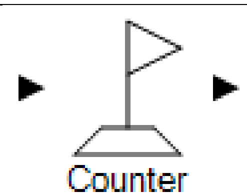
Release Next

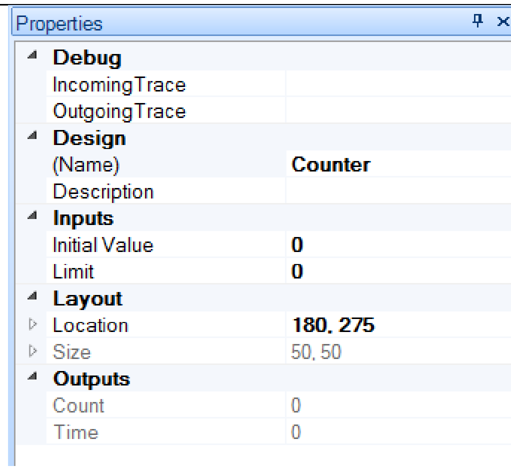
The release next element allows the user to model vertical construction. The release next element is placed after a task, which once completed, means it is viable to begin the next unit of construction. For example, the first floor must be installed before the second floor can begin.

Element symbol	
Properties	N/A
Input Parameters	N/A
Output Parameters	N/A

Counter

The counter element allows the user to track units though the model and assists in the detection of errors.

Element symbol	

<table>
<tr><td>Properties</td><td>
</td></tr>
<tr><td>Input Parameters</td><td>Initial Value – The initial value of the count
Limit – The limit of the count. When the count reaches the limit, the simulation will be stopped. If zero, the simulation will continue until all units have been processed</td></tr>
<tr><td>Output Parameters</td><td>Count – The number of units that have passed through the counter at the time the simulation stops
Time – The time at which the last unit through the counter was observed</td></tr>
</table>