Guidelines for Basic Truck-Shovel Simulation Modeling using Arena

Hooman Askari-Nasab & Mohammad Tabesh
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

1. Introduction

The purpose of this lab is to introduce the student to the Arena simulation environment in a step-by-step manner. Figures have been included to assist in the learning process. We will begin by utilizing some of the basic Arena functions and settings in order to simulate a basic truck-shovel system.

The decision making system used for assigning trucks to various operations in a mine is usually referred to as the dispatching system. The problem definition in this lab is based on example number 13 from the book 'Mine design: examples using simulation' by Sturgul (2000). One of the first papers discussing the dispatching system is by Arnold and W (1983) where principles of a dispatching system are presented by simple examples. The proposed system in their paper, known as DISPATCH, allocates trucks to shovels and also determines the optimum number of trucks required in the mine. One of the simple examples from the paper is going to be modeled in this lab.

2. Learning Objectives

By the end of this lab you should be able to:

- Learn how to capture a process with simulation.
- Understand what it takes to move from a description of your process to a simulation of your process.
- Understand why you need to simulate.
- Understand the kinds of questions and answers you can get from the use of Arena.
- Leave with sound practical advice for how to succeed with simulation.

3. Discrete Event Simulation

A model of a real or proposed system for the purpose of evaluating the system's behavior under various conditions In discrete event simulation we need only consider the logic of what occurs at event times, where an event is that point in time when the system status may change. For example, consider a single server, single queue system where there are two events: an arrival and a departure.

- The most powerful analysis technique for real world systems in which the dynamic and stochastic nature can be captured.
- Permits managers to visualize the operation of a new or existing system under a variety of conditions.

Askari-Nasab, Hooman (2012), Mining Optimization Laboratory (MOL) – Report Four, © MOL, University of Alberta, Edmonton, Canada, Pages 340, ISBN: 978-1-55195-301-4, pp. 226-254.

- Understand how various components interact with each other and how they affect overall system performance.

3.1. Why Simulate?

- Provide general insight into the nature of a process.
- Provide an understanding of how performance measures vary over time.
- Identify specific problems or problem areas within a system.
- Develop specific policies or plans for a process.
- Test new concepts and/or systems prior to implementation.
- Improve the effectiveness of a system.
- Provide an "insurance policy" for system performance.

3.2. Simulation Considerations

- Cannot give accurate results if the data is inaccurate.
- Cannot describe system characteristics that have not been explicitly modeled.
- Cannot solve problems — it can only provide information.
- Cannot provide easy answers to complex problems.

3.3. Simulation Project Management: The Players

- Simulation project team.
- System design team.
- Data/information sources.
- Implementation team.
- Contractors.
- Decision makers/management.

3.4. Starting the Study

- Have clearly stated and accepted objectives
- Get input from everyone
- Make certain all agendas are understood

3.5. The Simulation Process

- Define — functional specification
- Formulate — the simulation model
- Verify/validate — input from all players
- Analyze — statistical evaluation
- Recommend — alternatives to the decision maker(s)

3.6. Why Have a Functional Specification?

- Defines the problem completely
- Requires system understanding from the start
- Provides vision of the task
- Defines how simulation will be used
- Defines all assumptions of the simulation model
- Identifies data requirements
- Identifies required output statistics and analysis

3.7. Model Formulation

- One or more analysts
- Data structure requirements
- Model control logic
- Level of detail
- Flexibility
- Statistical requirements

3.8. Verification and Validation

- Verification: Ensuring that the model behaves in the way it was intended.
- Validation: Ensuring that the model behaves the same as the real system.

Requires:

1. Involvement of all the players,
2. Use of animation and data, and
3. Reasonable and robust model.

3.9. The Analysis

- Addresses the project objectives.
- An iterative process.
- Helps the analyst understand the results.
- Establishes result accuracy and sensitivity.

3.10. The Final Presentation

- Keep it short and simple
- Answer the right questions
- Address the audience
- Provide reasons for the results

4. Concepts and Terminology

4.1. Using Basic Process Modules

Topics covered in this section:

Concepts:

- Entity flow and attributes
- Capacity constraining (Resources), and
- Queues.

Arena Constructs:

- Create logic module
- Process logic module
- Dispose logic module
- Entity spreadsheet module
- Resource spreadsheet module
- Run > Setup

5. Concepts

Before you can build your first simulation model with Arena, there are a few fundamental concepts with which you will need to become familiar.

5.1. Entities

The items (documents, customers, parts, etc.) that are being produced, served, or otherwise acted on by our process. Arena is an entity-driven language. The entities represent any person, object, or thing, whether real or imaginary, whose movement through the system causes a change in the status of the system. Examples: customers moving through a restaurant, or parts moving through a factory. Logical entities: failure control, system control, staff breaks, etc.

5.2. Attributes

Within a system, there may be many types of entities, each having unique characteristics called Attributes. Attributes represent user-defined values associated with individual entities, such as customer type, product size, time job entered the system, etc. All entities have the same set of attributes (e.g., priority, arrival time) with different values. An assignment made to an entity's attribute affects only that entity. Some of the attributes that Arena automatically assigns are listed below. You can access and reassign these attributes when building your model:

- Entity.Type — An integer value assigned to entities when they are created. Used to distinguish one entity type from another. It is automatically assigned inside each Create module.
- Entity.Picture — The picture that will be displayed in the simulation animation. It is automatically assigned in the Create module (default picture is Picture.Report).
- Entity.CreateTime — The simulated time that the entity was created. Used to collect cycle time information.
- Entity.Station — The current station or station to which an entity is being moved. This attribute is updated automatically by Arena during a simulation run.
- Entity.Sequence — Number of the entity's visitation sequence. Used when different entities need to visit the same set of stations but in a unique order.
- Entity.JobStep — The entity's position or index into the sequence.

5.3. Resources

Object used to model an area where there is constraint. Constraints may be caused by limited numbers of people to perform an action, restricted storage space, capacity of equipment, etc. The resource capacity is the number of identical resource units corresponding to a specific resource. Entities seize resources to get control of one or more units of the resource. Entities release resources when they are no longer required. Whenever an entity seizes a resource, it must also release the same resource at some other time in the model. If the entity that controls a resource does not release it, all other entities requiring the same resource will be placed in a queue and be unable to seize the resource that will never become available.

5.4. Queues

Area where an entity resides, while waiting for a resource to become available or while waiting to be batched.

5.5. Seize, Delay, and Release

In virtually every simulation model, contention for some resource must be modeled. This resource could be anything from a machine or an operator to a storage space or an intersection. These types of constraints would be modeled as resources. When using resources in Arena, you should be aware of the three main commands that are associated with the use of every resource.

Seize: Use this command whenever you want an entity to gain control of a resource. If the resource is free when an entity arrives at a Seize command, the entity will reduce the capacity of the resource by the number of resource units required in the Seize module. If the required resources are not available, the entity will wait in a queue until the resource becomes available.

Delay: Once a resource is seized, there is usually some type of action or process that must occur before an entity can release the resource. This is usually some type of delay, either explicitly defined or implied due to some other actions. Explicitly defined delays would consist of a Delay module that defines the delay for some type of action or process. This delay can be a constant value, a random quantity from a defined distribution, or a mathematical expression. A delay can even be held by an entity attribute value that has been assigned prior to the Delay module. Implicit delays are ones caused by other actions or conditions in the simulation model. These delays include waiting to seize another resource, waiting to be moved (or being moved) by a transporter or conveyor, or waiting for a predefined condition to occur elsewhere in the model.

Release: After the delay, the entity needs to release the resource so that it will be available for other entities within the simulation model. This action is achieved by using a Release module. Once an entity releases a resource, Arena will allow the next entity waiting for this particular resource to seize it.

6. Arena Constructs

6.1. Create module

This module is intended as the starting point for entities in a simulation model. Entities are created using a schedule or based on a time between arrivals. Entities then leave the module to begin processing through the system. The entity type is specified in this module.

6.2. Process module

This module is intended as the main processing method in the simulation. Options for seizing and releasing resource constraints are available. Additionally, there is the option to use a "sub-model" and specify hierarchical user-defined logic. The process time is allocated to the entity and may be considered to be value added, non-value added, transfer, wait or other. The associated cost will be added to the appropriate category.

6.3. Assign module

This module is used for assigning new values to variables, entity attributes, entity types, entity pictures, or other system variables. Multiple assignments can be made with a single Assign module.

7. Truck-Shovel simulation

In any truck-shovel mining operation, there are a number of shovels that excavate the material and load it into the trucks. Depending on the project specifications, the extracted material are separated into different types and sent to proper destination. However in a general case, always there are two main various materials, the ore and the waste. The ore is sent to the crusher (processing plant) and the waste is sent to the waste dump. The mining company makes revenue out of the number of ore loads that are delivered to the processing plant. On the other hand, using more trucks results in more variable costs corresponding to each individual truck added to the fleet. In addition, there is also a fixed cost, regardless of the number of trucks. Therefore, the profit is the revenue made from ore loads minus the fix and variable costs. One approach for determining the appropriate number of trucks is to use simulation. Different scenarios can be defined and tested by running various simulation models and comparing the objective values gained in each scenario.

8. Basic Truck-Shovel Model

The truck-shovel system in a mine is setup in a way that trucks are assigned to either ore or waste and will return to the same shovel they were first assigned to after each dumping. The dispatching system is defined as sending trucks to the shovel other than the one visited in the last cycle. The two system with and without dispatching are shown in Fig. 1. The directed arcs represent the permitted paths of trucks. The model has two shovels and two dumping regions. One shovel is loading ore trucks. These trucks travel to the crusher where they dump their loads. The other shovel is loading waste. These trucks go to the waste dump. After a truck dumps either waste or ore, it may return to either shovel. The possible feasible paths are shown by the lines of flow. There are other possible paths, such as a trucks going from shovel to another shovel, but these are not considered as being feasible (Sturgul, 2000).

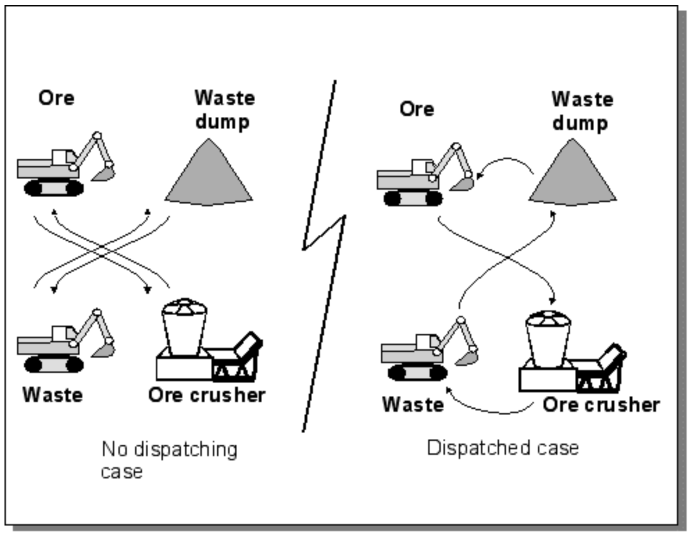


Fig. 1. Two different cases of the problem (Sturgul, 2000).

9. Average Constant Travel Time – Part A

9.1. Problem definition

There are two similar shovels working in a mine. The first one is working in the ore zone opposed to the second one which only digs waste. The number of trucks in the system is to be determined using a simulation model. The average travel times in the mine are presented in Table 1.

It is important to remember that a simulation model will not directly solve for the correct number of trucks to have in the mine but only indicate how the system will perform for any given number of trucks. For the data as given above, the different operational times are all constant so that there will be no queues if the correct number of trucks are selected. Consider the first scenario in Fig. 1.

The correct number of trucks can be found by assuming that one truck is at the ore shovel all the time. The number of trucks required to travel between crusher and shovel 1 to make sure there is always one truck at the shovel is equal to the summation of the travel times divided by loading time at the shovel as shown in Eq. 1.

Table 1. Average Travel and Operation Times (Minutes).

Operation	Mean
Shovel 1 to crusher	12
Crusher to shovel 1	8
Shovel 2 to dump	12
Waste dump to shovel 2	8
Crusher to shovel 2	4
Waste dump to shovel 1	3
Load at shovel 1	3
Dump at crusher	1
Load at shovel 2	3
Dump at waste	1

As the trucks at the waste dump don't interact with ore trucks they should be treated separately. Since the various times for loading, hauling, dumping and returning are identical, the correct number of trucks for hauling waste to the dump is also 8. Thus, the number of trucks to have in the mine is 16.

$$Num\ of\ Trucks = \frac{Total\ Travel\ Time}{Loading\ Time} + 1 \qquad (1)$$

$$Num\ of\ Trucks\ case\ 1 = \frac{(12+8+1)\min}{3\min} + 1 = 8 \rightarrow 8 \times 2 = 16 \qquad (2)$$

Following a similar line of reasoning, the correct number of trucks to have for the second case equals 12.33 trucks and so is rounded up to the next integer, namely, 13 to avoid queuing.

$$Num\ of\ Trucks\ case\ 2 = \frac{(12+4+12+3)\min}{3\min} + 2 = 12.33 \approx 13 \qquad (3)$$

9.2. Flow diagram

We need to develop two models: one for the current case and one with the dispatching system. Fig. 2 shows the flow diagram of the truck-shovel model with no dispatching, whereas Fig. 3 illustrates the truck-shovel model with the dispatching rule. The travel times are constant average values.

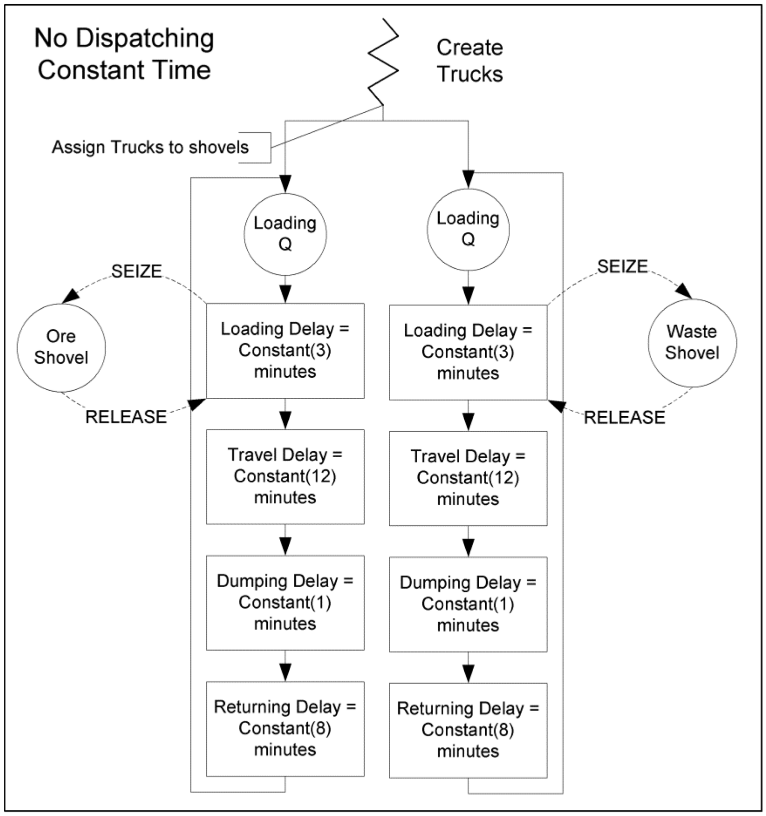


Fig. 2. Flow diagram Model 1, no dispatching, constant time.

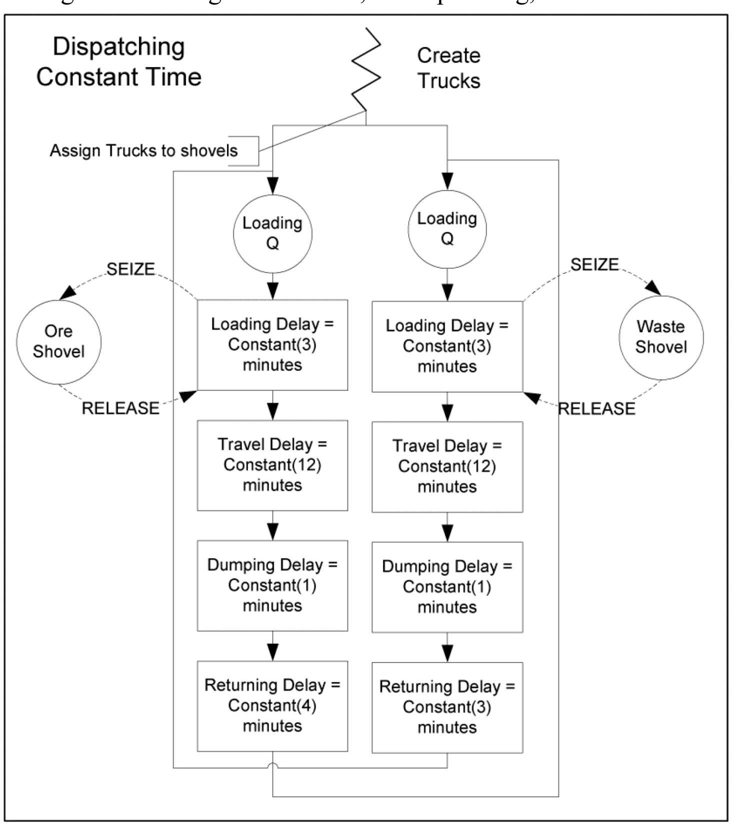


Fig. 3. Flow diagram, Model 2, with dispatching, constant time.

9.3. Definition of terminology

9.3.1. Entities

- entTruck: the entity representing trucks

9.3.2. Variables

- varOreLoadCount: used for counting the number of loads dumped into the crusher.
- varWasteLoadCount: used for counting the number of loads dumped at the waste dump.

9.3.3. Resources

- resShovel1: shovel working in ore
- resShovel2: shovel working in waste

9.3.4. Statistics

- sttOreLoadCount: reports the varOreLoadCount.
- sttWasteLoadCount: reports the varWasteLoadCount.
- sttOreTonnage: used for reporting the tonnage of ore loads at the end of the simulation.
- sttWasteTonnage: used for reporting the tonnage of waste loads at the end of the simulation.
- sttTotalTonnage: used for reporting the tonnage of waste loads at the end of the simulation.

9.4. Modeling approach and Pseudo code

- ***Create Trucks:*** use two separate create modules to create trucks for ore and waste mining
- ***Trucks are Loaded:*** shovels are seized and delayed until the trucks are loaded in waste or ore
- ***Trucks Travel to Their Destination:*** two process module are used to delay for the traveling time from mining areas to the destination, either crusher or waste dump
- ***Trucks Dump Material:*** two process modules are to delay trucks for the dumping time
- ***Trucks Return to the Assigned Shovel:*** two process modules to delay trucks for the traveling time back to the shovels. For the model without dispatching, this means returning to the same shovel the truck was load with. In contrast, trucks switch their shovel after each cycle.

9.5. A snapshot of the completed Arena model

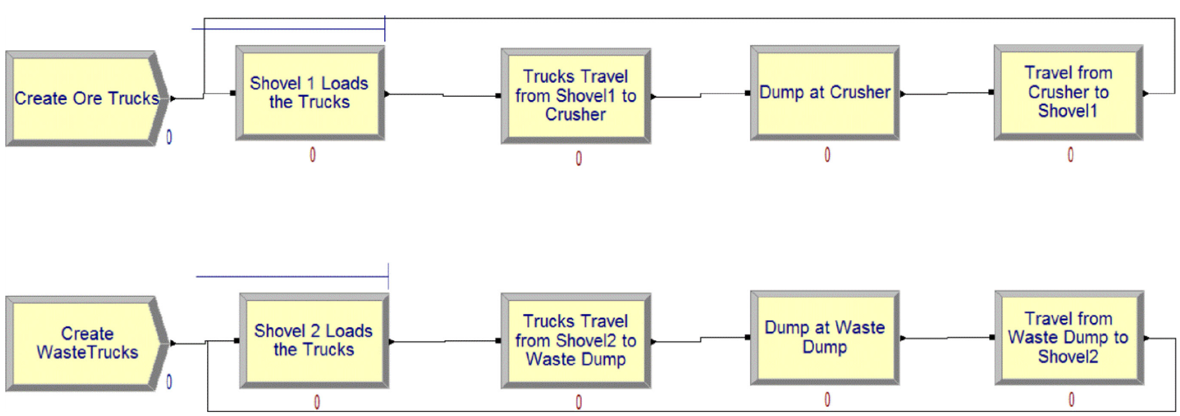


Fig. 4. Model 1 Snapshot (Current System without Dispatching).

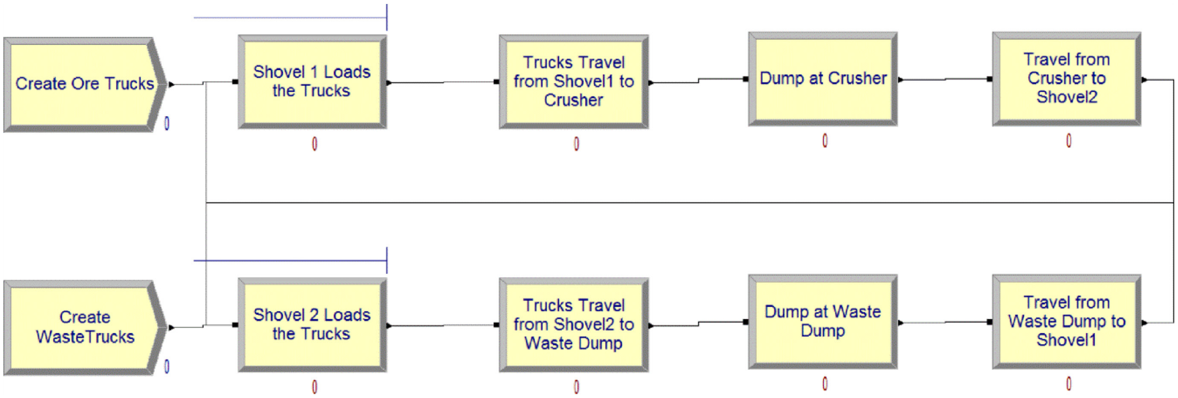


Fig. 5. Model 2 Snapshot (Proposed System with Dispatching).

9.6. Step by step modeling

9.6.1. Define Replication Parameters— Use Run > Setup

The model is going to be run for 5 days of 8 hours. The hours per day field is set to 8 and the base time unit is set to minutes.

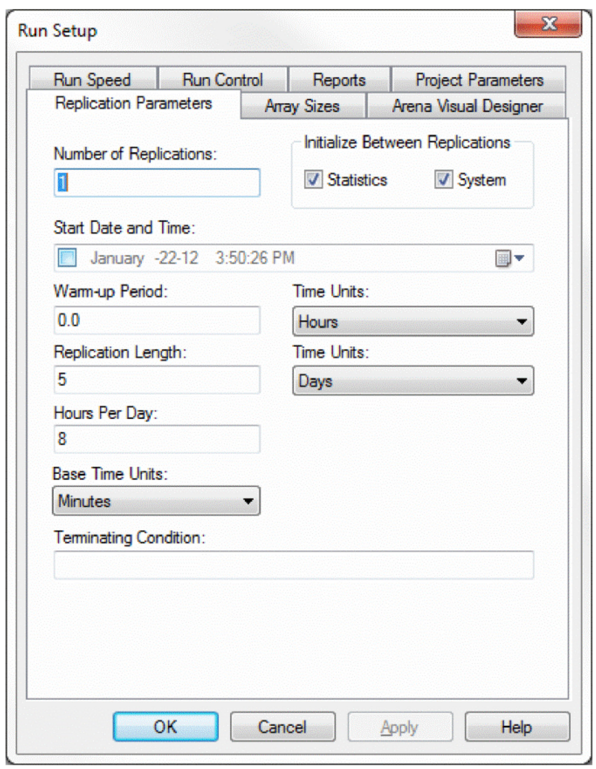


Fig. 6. Replication Parameters

Number of Replications	1
Warm-up Period	0 hours
Replication Length	5 Days
Hours per Day	8
Base Time Units	Minutes

9.6.2. **Create Two Sets of Trucks —** Use Two Create Modules

Two similar create modules are required to create truck entities. The only difference between the create modules is the maximum arrivals which is equal to the number of trucks needed for ore and waste. The number of trucks is assumed to be equal for ore and waste. We start with 5 trucks in each system and then increase the number of trucks to assess the impact of the truck numbers on the utilization of the shovels. Fig. 7 and Fig. 14 show the create modules for trucks in ore and waste areas respectively.

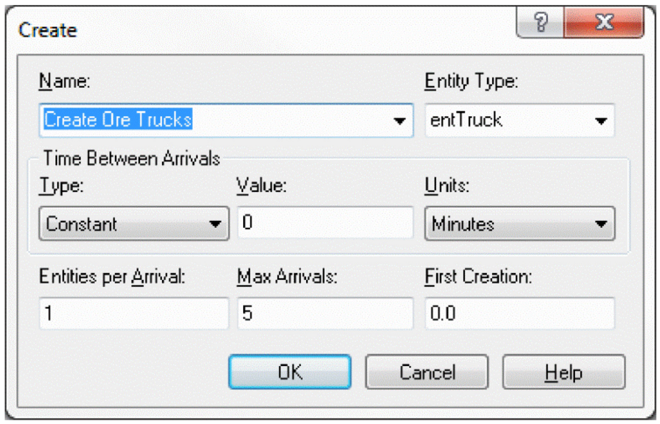
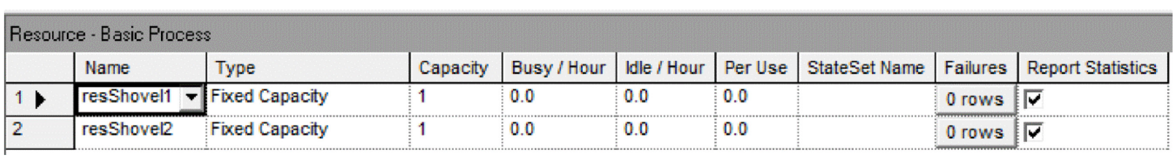


Fig. 7 .Create Module for Trucks in Ore.

Name		Create Ore Trucks
Entity Type		entTruck
Time Between Arrivals	Type	Constant
	Value	0
	Units	Minutes
Max Arrivals		5

9.6.3. **Define Resources** — Use Resource Data Module



Resource - Basic Process

	Name	Type	Capacity	Busy / Hour	Idle / Hour	Per Use	StateSet Name	Failures	Report Statistics
1	resShovel1	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑
2	resShovel2	Fixed Capacity	1	0.0	0.0	0.0		0 rows	☑

Fig. 8. Define Resources.

9.6.4. **Shovel 1 Loads the Trucks** — Use Process Module

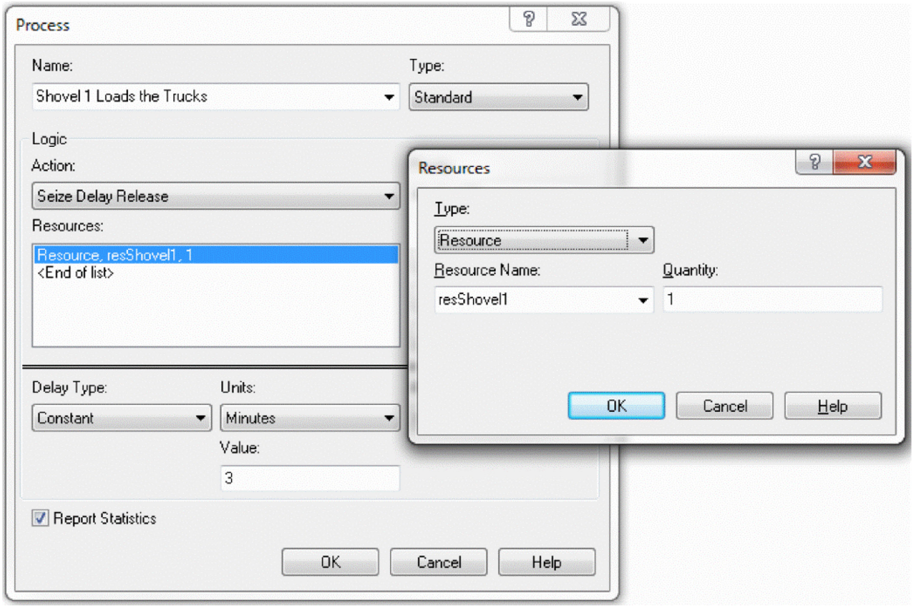


Fig. 9. Shovel 1 Loads the Trucks

Name		Shovel 1 Loads the Trucks
Action		Seize Delay Release
Resources	Type	Resource
	Resource Name	resShovel1
	Quantity	1
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	3

9.6.5. **Trucks Travel from Shovel1 to Crusher** — Use Process Module

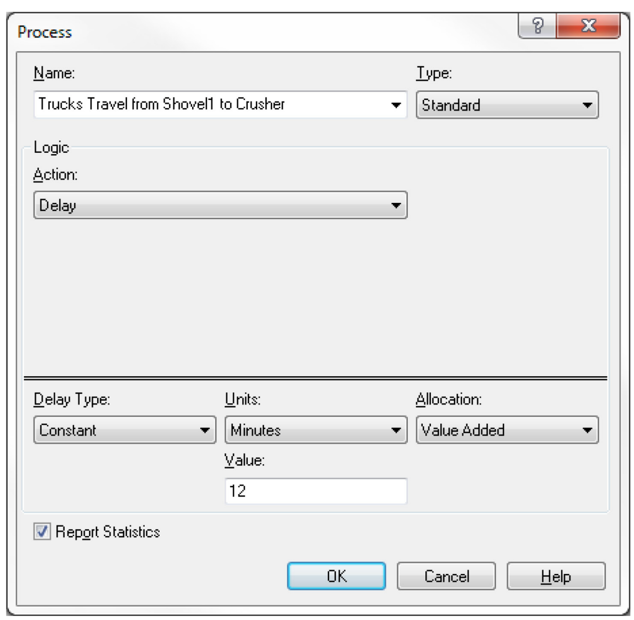


Fig. 10. Trucks Travel from Shovel1 to Crusher.

Name		Trucks Travel from Shovel1 to Crusher
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	12

9.6.6. **Dump at Crusher**— Use Process Module

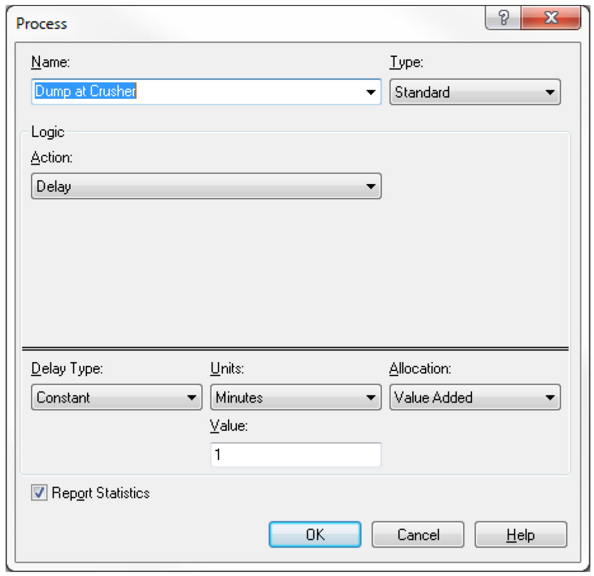


Fig. 11. Dump at Crusher.

Name		Dump at Crusher
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	1

9.6.7. **Travel from Crusher to Shovel1** — Use Process Module

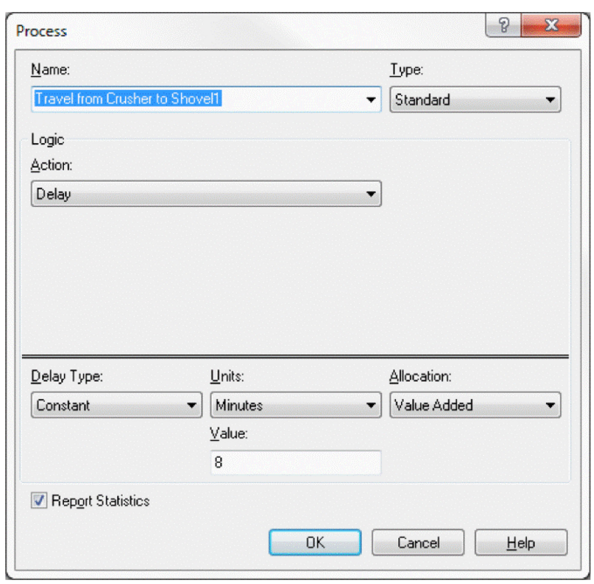


Fig. 12. Travel from Crusher to Shovel1.

Name		Travel from Crusher to Shovel1
Action		Delay
Delay Time	Delay Type	Normal
	Units	Minutes
	Value	8

9.6.8. **Form the Cycle** — Use Connect Button

Use the connect button from the ribbon to connect the output of the last process back to the inlet of the "Shovel 1 Loads the Trucks" process. The completed cycle should look like Fig 1.

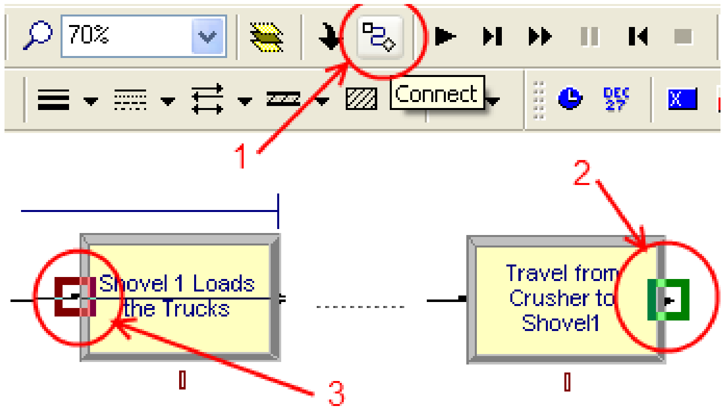


Fig. 13. Connect Button in the Ribbon

9.6.9. Create Waste Trucks — Use Create Module

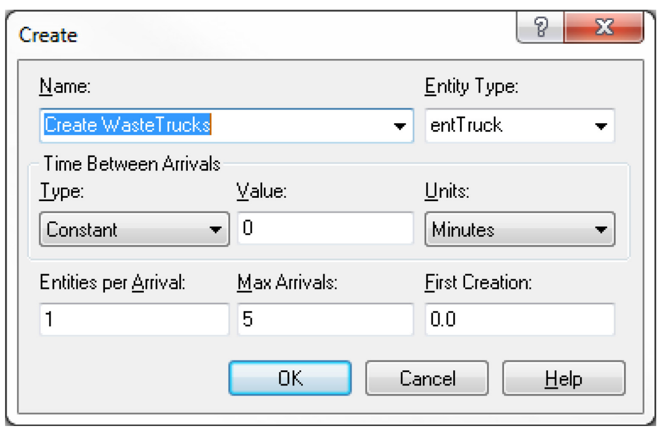


Fig. 14. Create Module for Trucks in Waste.

Name		Create Waste Trucks
Entity Type		entTruck
Time Between Arrivals	Type	Constant
	Value	0
	Units	Minutes
Max Arrivals		5

9.6.10. **Shovel 2 Loads the Trucks** — Use Process Module

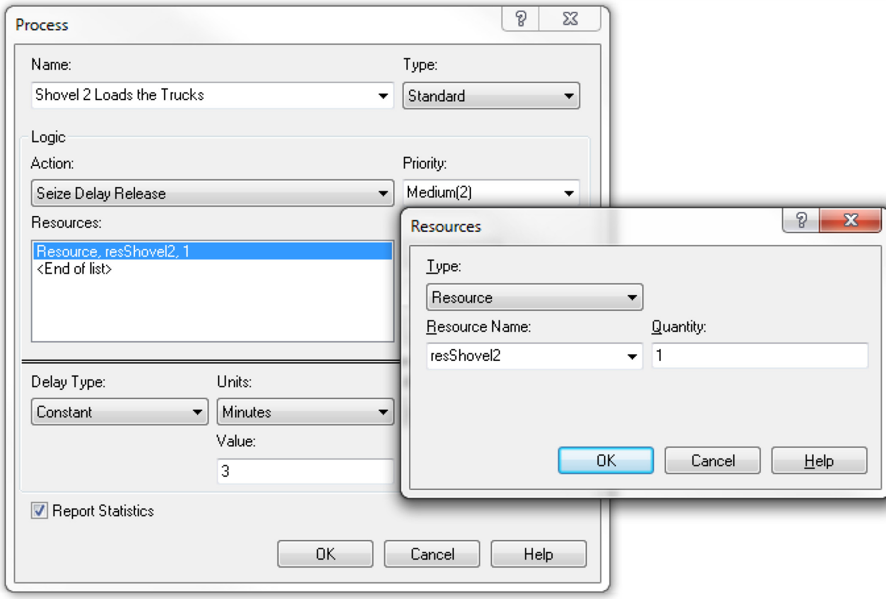


Fig. 15. Shovel 2 Loads the Trucks.

Name		Shovel 2 Loads the Trucks
Action		Seize Delay Release
Resources	Type	Resource
	Resource Name	resShovel2
	Quantity	1
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	3

9.6.11. **Trucks Travel from Shovel2 to Waste Dump** — Use Process Module

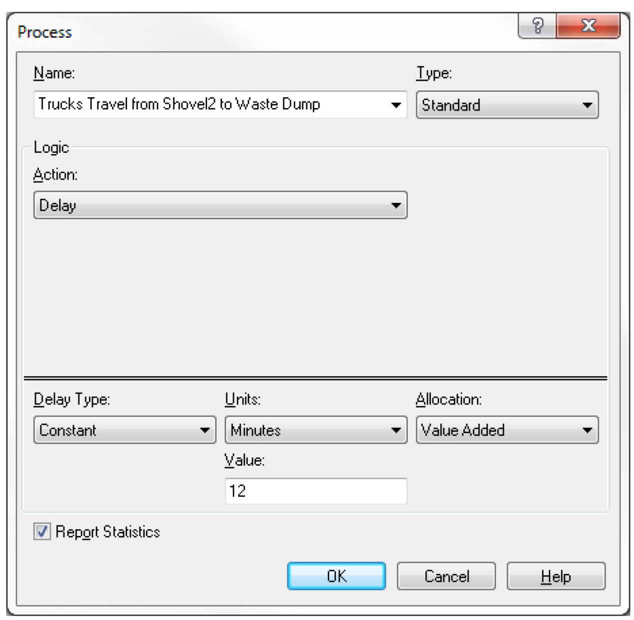


Fig. 16. Trucks Travel from Shovel2 to Waste Dump

Name		Trucks Travel from Shovel2 to Waste Dump
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value (Mean)	12

9.6.12. **Dump at Waste Dump** — Use Process Module

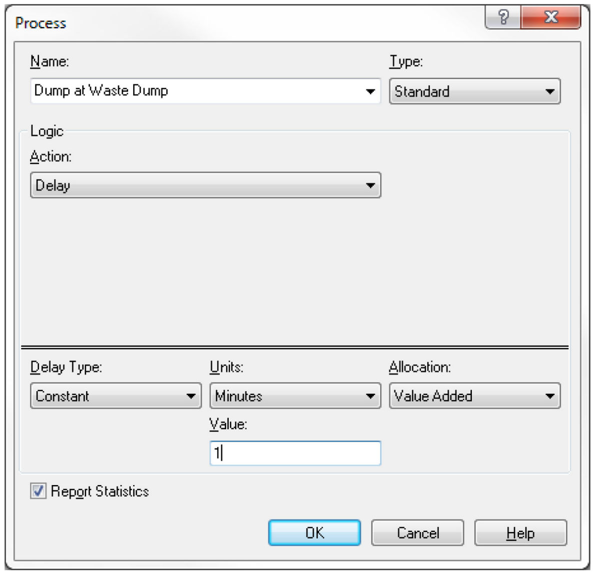


Fig. 17. Dump at Waste Dump.

Name		Dump at Waste Dump
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	1

9.6.13. **Travel from Waste Dump to Shovel2** — Use Process Module

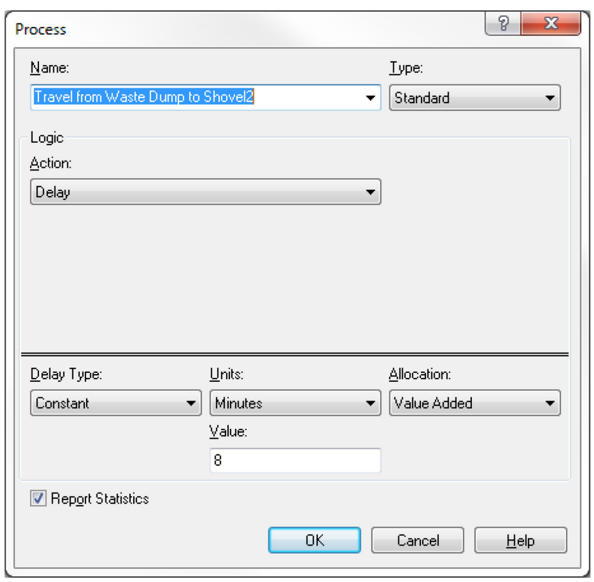


Fig. 18. Travel from Waste Dump to Shovel2.

Name		Travel from Waste Dump to Shovel2
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value (Mean)	8

9.6.14. **Form the Cycle —** Use Connect Button

Connect "Travel from Waste Dump to Shovel2" to "Shovel 2 Loads the Trucks" as shown in Fig. 13.

9.6.15. **Save the Model —** Use File > Save Button

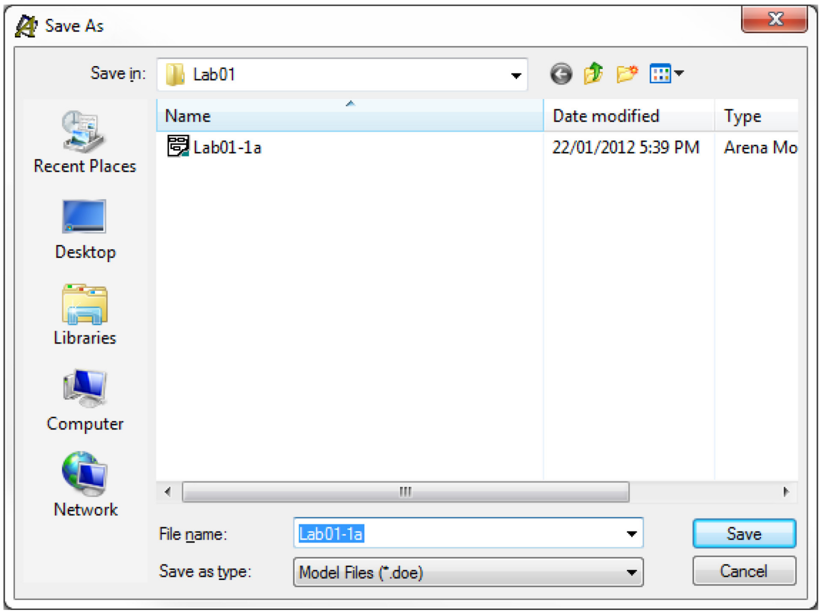


Fig. 19. Save the First Model.

9.6.16. **Save the Model as… —** Use File > Save As … Button

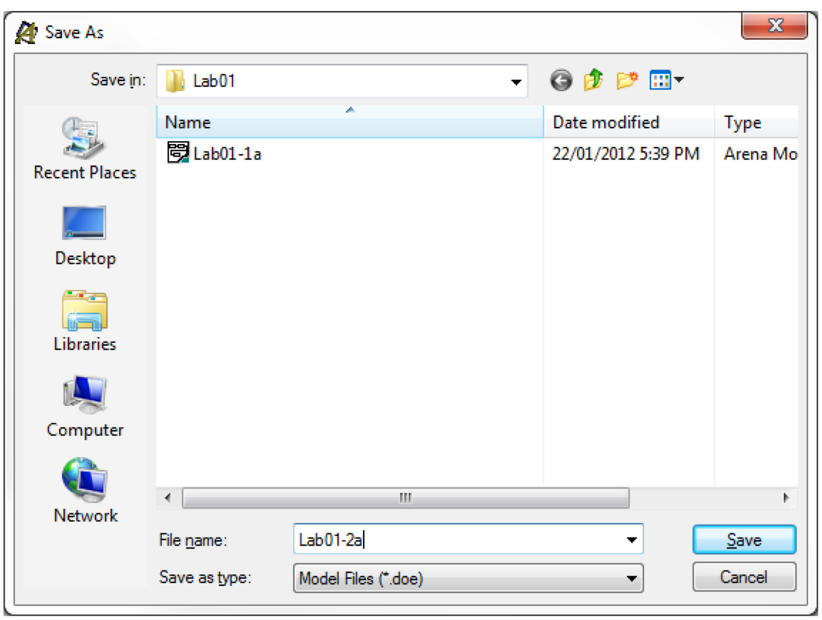


Fig. 20. Save with the New Name.

9.6.17. **Delete the Cycle Links** — Select the Link > Right Click > Delete Button

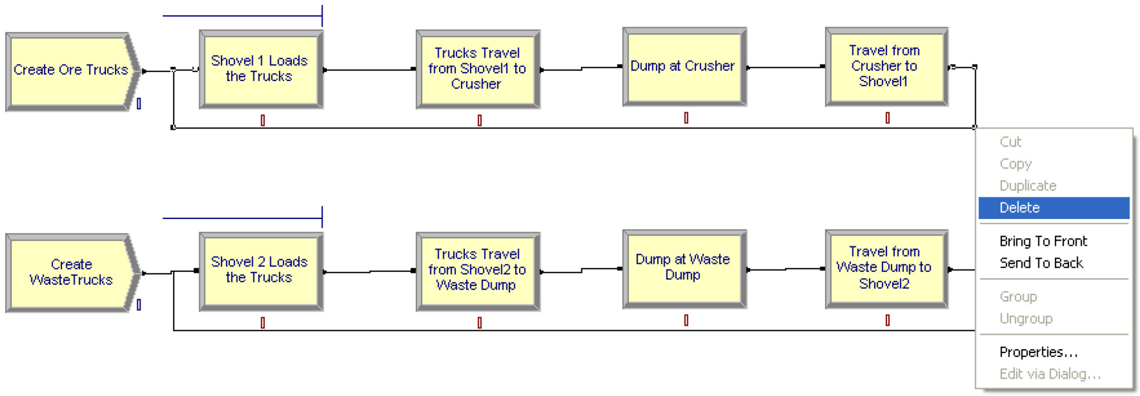


Fig. 21. Delete Cycle Links.

9.6.18. **Form New Cycles** — Use Connect Button

Connect "Travel from Crusher to Shovel1" to "Shovel 2 Loads the Trucks" as shown in Fig 19.

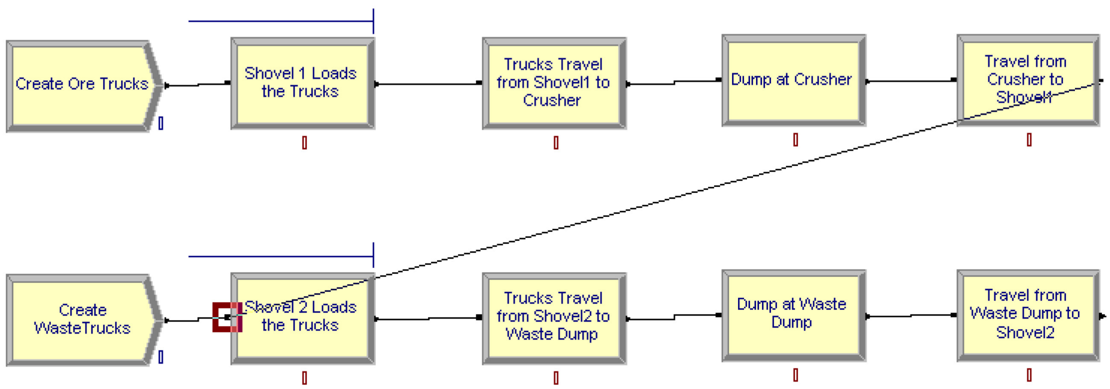


Fig. 22. Form New Cycles.

Connect "Travel from Waste Dump to Shovel2" to "Shovel 1 Loads the Trucks" as shown in Fig 20.

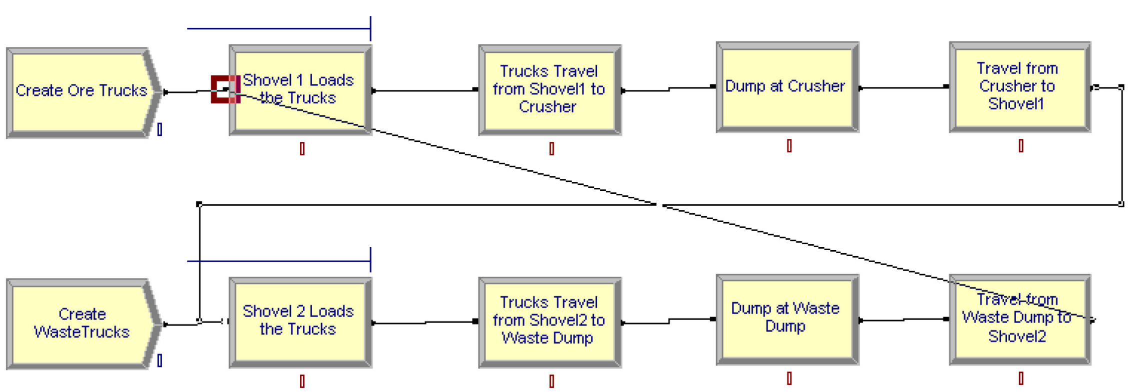


Fig. 23. Form New Cycles.

9.6.19. **Update Travel Time values from Crusher to Shovel1 — Double Click on the Process**

Update name and delay fields from data shown in Table 1 and Fig. 3.

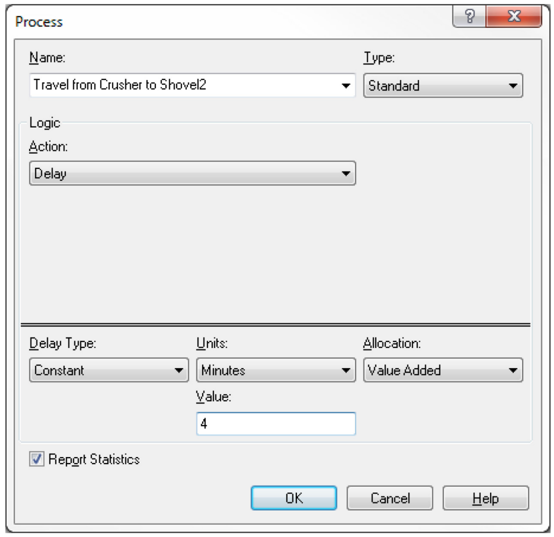


Fig. 24. Update Process.

Name		Travel from Crusher to Shovel2
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value	4

9.6.20. **Update "Travel from Waste Dump to Shovel2" Process —** Double Click on the Process Update name and delay fields from data shown in Table 1 and Fig. 3.

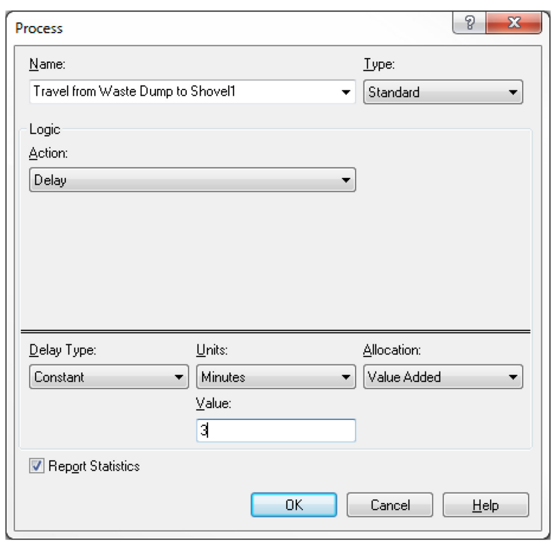


Fig. 25. Update Process.

Name		Travel from Waste Dump to Shovel1
Action		Delay
Delay Time	Delay Type	Constant
	Units	Minutes
	Value (Mean)	3

10. Variable Travel Time – Part B

10.1. Problem definition revised

In practice, the travel times in a mine are not constant but have to be determined from motion and time studies. It will be instructive at this time to determine if the solutions to the two previous problems change if the various times are given by different statistical distributions. As a first approximation, let us assume that all the times are normally distributed with a mean as given in and a standard deviation of 19% of the mean. Table 2 gives these times. Since we are now sampling from the normal distribution, this means that all relevant times for loading, hauling, dumping and returning will need to be changed.

Table 2. Process Times for the Problem (Minutes).

Operation	Mean	Std. Dev.
Shovel 1 to crusher	12	2.28
Crusher to shovel 1	8	1.52
Shovel 2 to dump	12	2.28
Waste dump to shovel 2	8	1.52
Crusher to shovel 2	4	0.76
Waste dump to shovel 1	3	0.57
Load at shovel 1	3	0.57
Dump at crusher	1	0.19
Load at shovel 2	3	0.57
Dump at waste	1	0.19

10.2. Flow diagram

We need to develop two models: one for the current case and one with the dispatching system similar to what we did for the constant average travel time. We are going to update the travel times of the current models that we have built so far with variable time. Fig. 26 and Fig. 27 show the flow diagram of the two models with variable travel times.

- Save a copy of the model Lab01-1a as Lab01-1b.
- Save a copy of the model Lab01-2a as Lab01-2b.

10.3. New Arena Constructs

10.3.1. Using Basic Process and Advanced Process Modules

Topics covered in this section:

Concepts:

- User defined variables

Arena Constructs:

- Variable Module
- Assign Module
- Record Module
- Statistic Spreadsheet Module (Advanced Process)

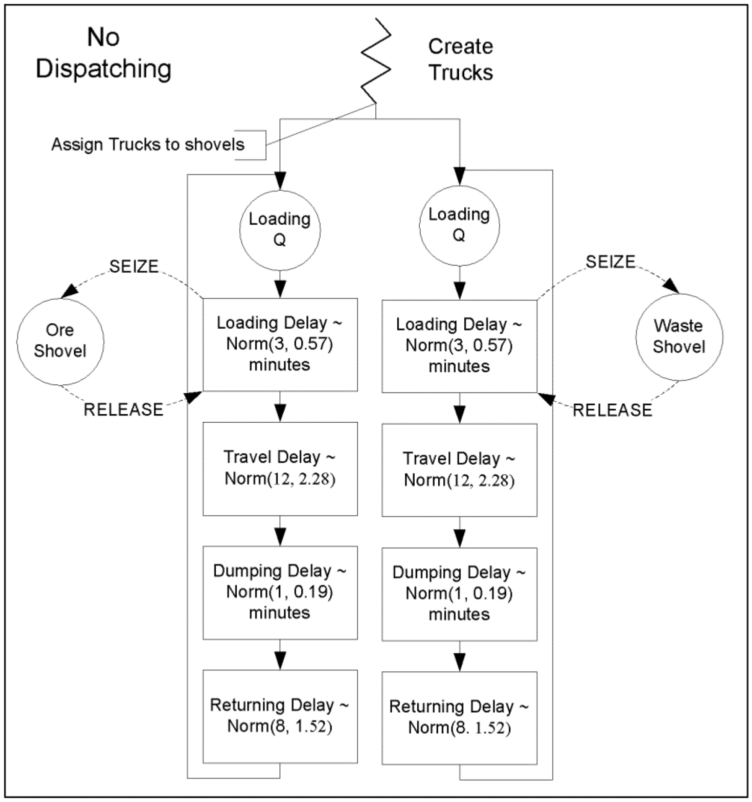


Fig. 26. Flow diagram Model 1, no dispatching, variable times.

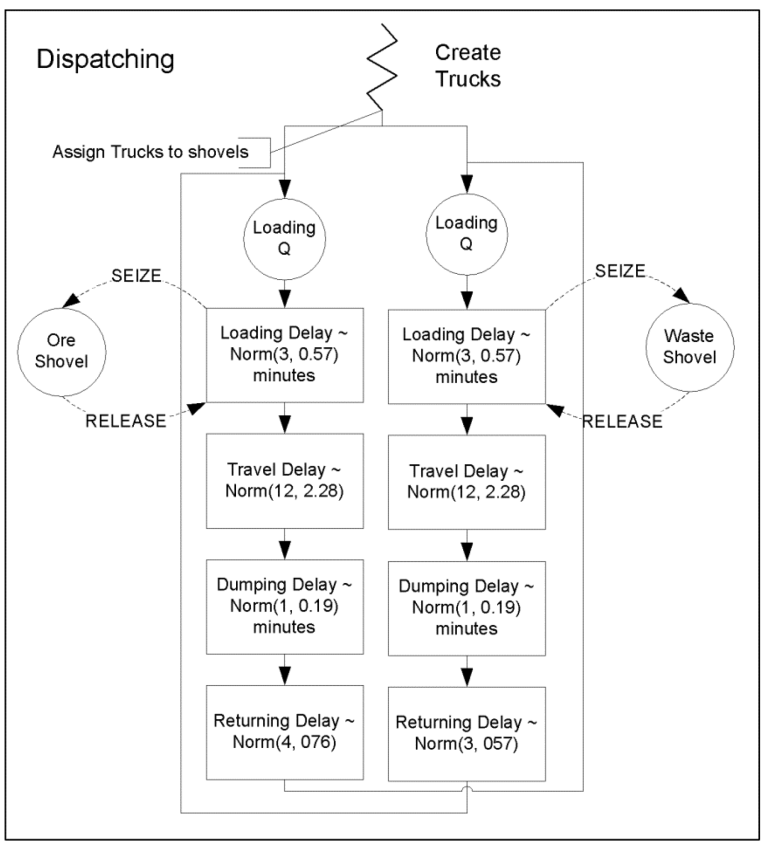


Fig. 27. Flow diagram, Model 2, with dispatching, variable time.

10.4. Terminology

We are going to update Model 1 and Model 2 with introducing new module with the following terminology:

10.4.1. Variable Spreadsheet Module (Basic Process)

This data module is used to define a variable’s dimensions and values. You can reference variables in other modules (for example, the Decide module), reassign new values to variables with the Assign module, and use variables in any expression. You can create single element variables, 1-D array variables, or 2-D array variables. To create a 1-D array variable, input a numeric value in the Rows cell, and leave the Columns cell blank. To create a 2-D array variable, input numeric values in both Rows and Columns cells. You can assign the variable's values by using a file, and you can specify the values manually in the Variable module.

10.4.2. Assign Module (Basic Process)

This data module is used to define a variable’s dimensions and values. You can reference variables in other modules (for example, the Decide module), reassign new values to variables with the Assign module, and use variables in any expression. You can create single element variables, 1-D array variables, or 2-D array variables. To create a 1-D array variable, input a numeric value in the Rows cell, and leave the Columns cell blank. To create a 2-D array variable, input numeric values in both Rows and Columns cells. You can assign the variable's values by using a file, and you can specify the values manually in the Variable module.

10.4.3. Record Module (Basic Process)

This module is used to collect statistics in the simulation model. Various types of observational statistics are available, including time between exits through the module, entity statistics (time, costing, etc.), general observations, and interval statistics (from some time stamp to the current simulation time). A count type of statistic is available as well. Tally and Counter sets can also be specified.

10.4.4. Statistics Module (Advanced Process)

The Statistic module is used to define additional statistics to collect during the simulation and also to specify output data files. While summary statistics (for example, Average and Maximum) are automatically generated for each statistic, if you specify an output file, then each individual observation (for example, each tally recorded) is written to an output file. The types of statistics you can define in the Statistics module are time-persistent, tally (observational data), count-based, output, and frequency-based. You cannot write data to an output file for time-persistent and frequency statistics if you specify a collection period other than the length of the entire replication.

10.5. Modeling Approach – Updating the Model

1. Create two new variables
   a. *varOreLoadCount:* used for counting the number of loads dumped into the crusher.
   b. *varWasteLoadCount:* used for counting the number of loads dumped at the waste dump.
2. Add an Assign Module:
   a. Count the loads of ore and waste dumped at the crusher and waste dump.
3. Record the number of loads and calculate the total tonnage based on the assumption that 150 ton trucks are used.
4. Add Statistics Module to record the ore count, waste count, and tonnages.
5. Update the travel times for the processes with normal distributions defined in Table 2.

11. A snapshot of the completed Arena model

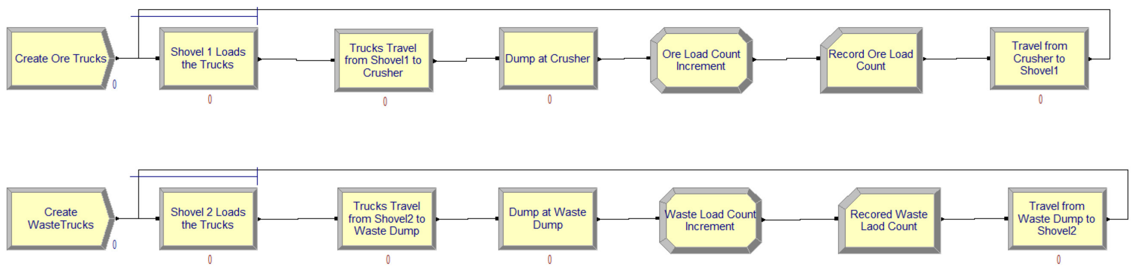


Fig. 28. Snapshot of Model 1.

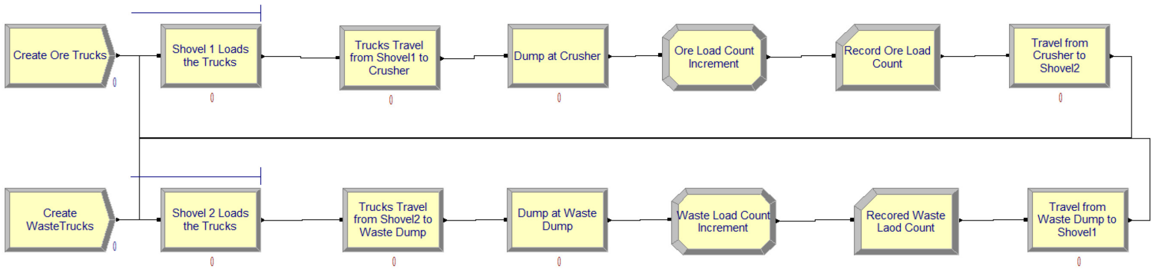


Fig. 29. Snapshot of Model 2.

12. Step by step updating the models

12.1. Create two new variables for counting ore and waste loads

In the Basic Process add two new Variables:

- varOreLoadCount:
- varWasteLoadCount:

Variable - Basic Process

	Name	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics
1	varOreLoadCount			Real	System		1 rows	☐
2	varWasteLoadCount			Real	System		1 rows	☐

Double-click here to add a new row.

Fig. 30. Defining user variables.

12.2. Define Statistics Data Module – Use Statistic Module in Advanced Process

Add the following stats to the Statistic Spreadsheet Module in Advanced Process.

a) *sttOreLoadCount:* reports the varOreLoadCount.
b) *sttWasteLoadCount:* reports the varWasteLoadCount.

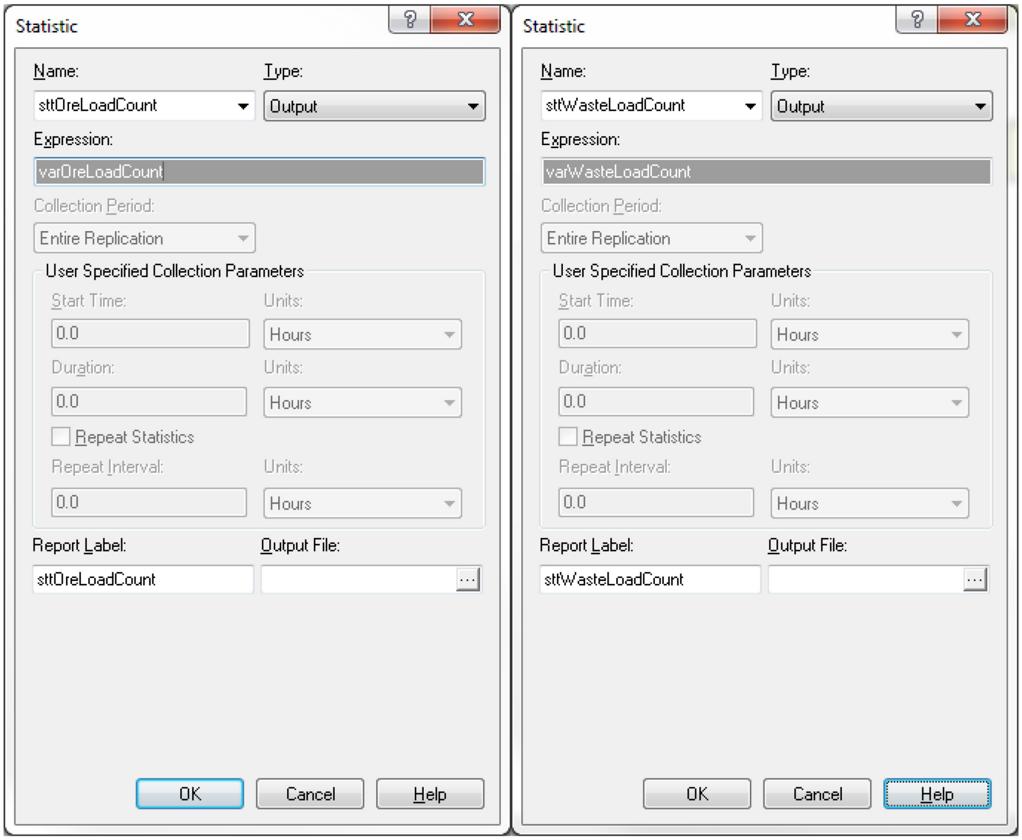


Fig. 31. Ore and Waste Load Counts.

Name	sttOreLoadCount
Type	Output
Expression	varOreLoadCount
Report Label	sttOreLoadCount

Name	sttWasteLoadCount
Type	Output
Expression	varWasteLoadCount
Report Label	sttWasteLoadCount

c) *sttOreTonnage:* used for reporting the total tonnage of ore loads at the end of the simulation.

Name	sttOreTonnage
Type	Output
Expression	varOreLoadCount*150
Report Label	sttOreTonnage

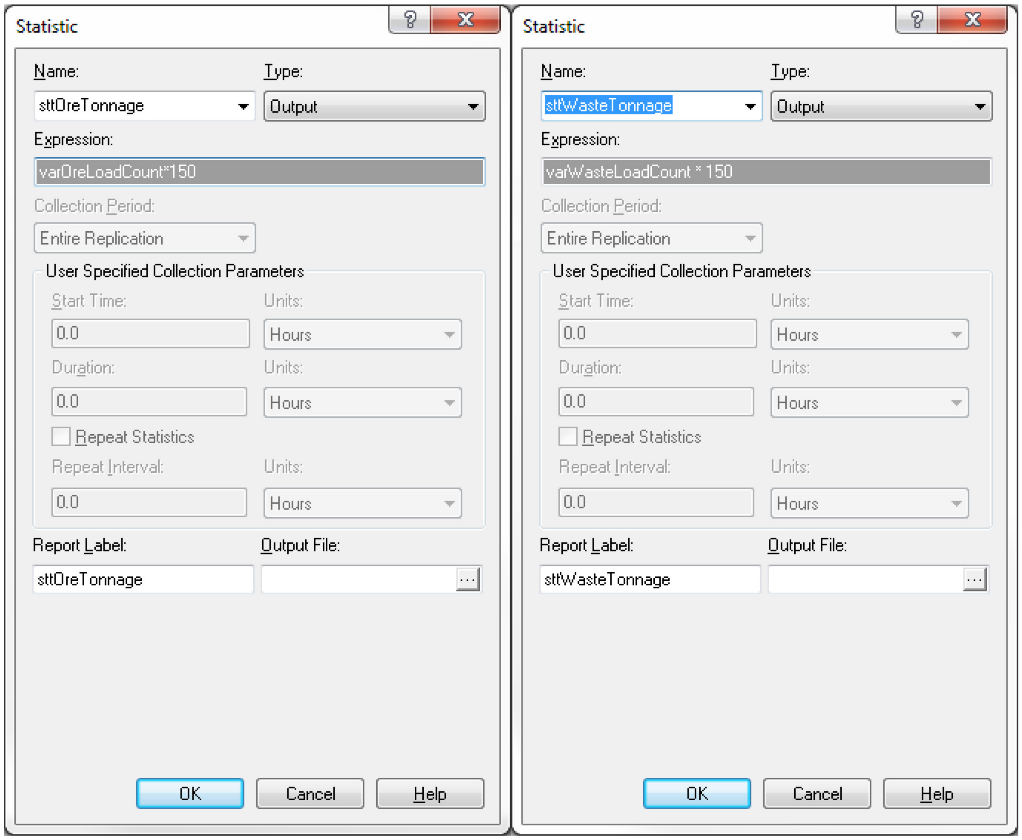


Fig. 32. Ore and Waste Tonnage Statistics

d) *sttWasteTonnage:* used for reporting the number of waste loads at the end of the simulation.

Name	sttWasteTonnage
Type	Output
Expression	varWasteLoadCount * 150
Report Label	sttWasteTonnage

e) *sttTotalTonnage:* used for reporting the number of waste loads at the end of the simulation.

Name	sttTotalTonnage
Type	Output
Expression	(varOreLoadCount + varWasteLoadCount) * 150
Report Label	sttTotalTonnage

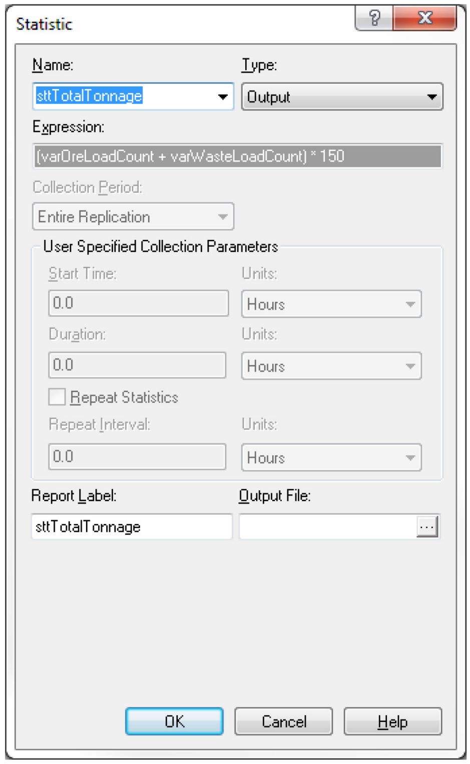


Fig. 33. Total tonnage statistics.

Statistic - Advanced Process

	Name	Type	Expression	Report Label	Output File
1	sttOreTonnage	Output	varOreLoadCount*150	sttOreTonnage	
2	sttWasteTonnage	Output	varWasteLoadCount * 150	sttWasteTonnage	
3	sttTotalTonnage	Output	(varOreLoadCount + varWasteLoadCount) * 150	sttTotalTonnage	
4	sttOreLoadCount	Output	varOreLoadCount	sttOreLoadCount	
5	sttWasteLoadCou	Output	varWasteLoadCount	sttWasteLoadCount	

Double-click here to add a new row.

Fig. 34. Statistics Data Module after the update.

12.3. **Add two Assign Modules for counting the loads –** Use Basic Process

Use assign module to count the number of ore and waste loads.

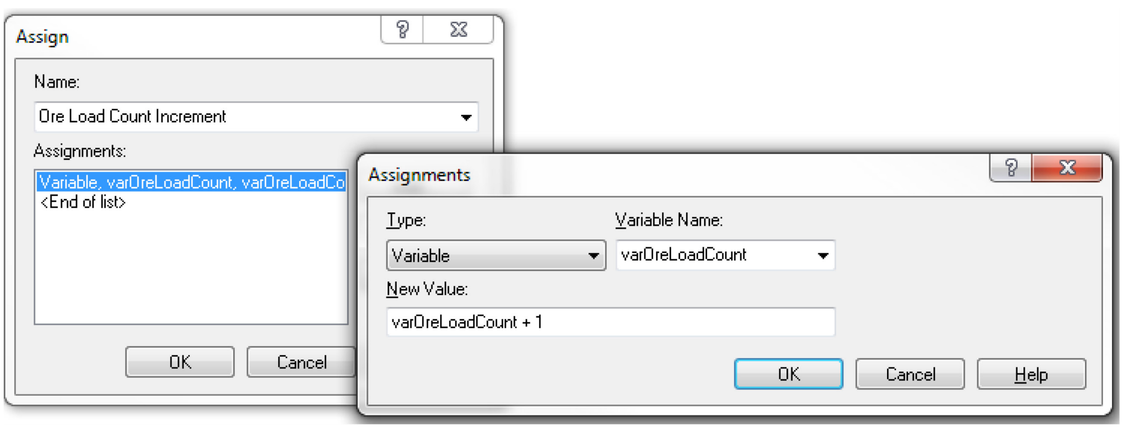


Fig. 35. Ore Load Count Assign Module

Name		Ore Load Count Increment
Assignments	Type	Variable
	Variable Name	varOreLoadCount
	New Value	varOreLoadCount + 1

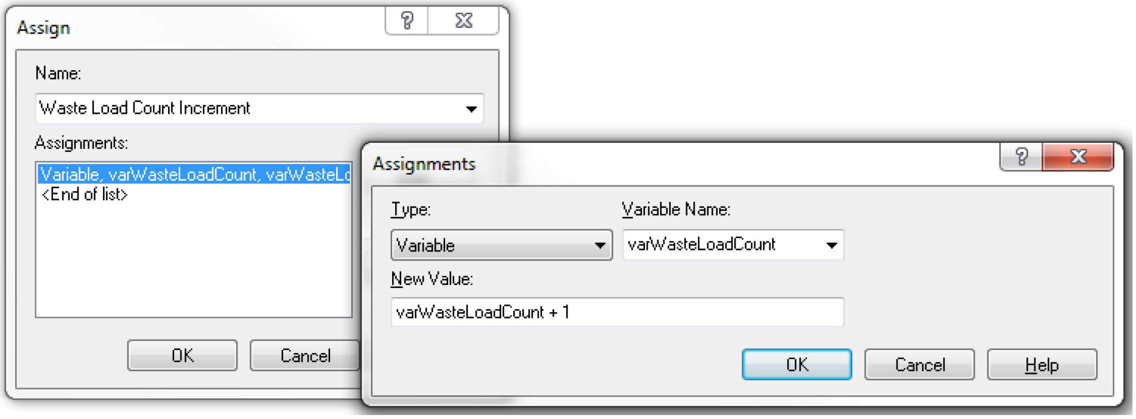
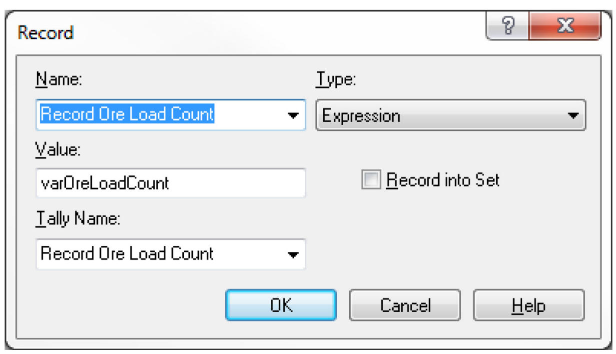


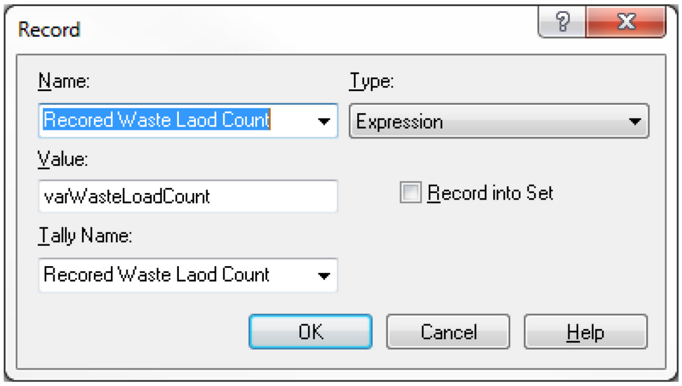
Fig. 36. Waste Load Count.

Name		Waste Load Count Increment
Assignments	Type	Variable
	Variable Name	varWasteLoadCount
	New Value	varWasteLoadCount + 1

12.4. **Add two Record Modules for counting the loads –** Use Record from Basic Process



Name		Record Ore Load Count
Record	Type	Expression
	Value	varOreLoadCount
	Tally Name	Record Ore Load Count



Name		Record Waste Load Count
Record	Type	Expression
	Value	varWasteLoadCount
	Tally Name	Record Waste Load Count

12.5. Run Models

Run the models with increasing the number of trucks and recording the key performance indicators of the systems with the two models. The number of trucks in the system could be changed by changing the Maximum Arrivals in the Create Module.

Table 3. Model 1b – Constant Average Travel Time – No Dispatch

Number of Trucks	# ore loads	# of waste loads	Ore (tons)	Waste (tons)	Ore shovel utilization %	waste shovel utilization %	Avg Queue Length
4×2	799	799	119,850	119,850	50	50	
5×2	998	998	149,700	149,700	62.5	62.5	0
6×2	1197	1197	179,550	179,550	75	75	0
7×2	1396	1396	209,400	209,400	87.5	87.5	0
8×2	1595	1595	239,250	239,250	100	100	0
9×2	1595	1595	239,250	239,250	100	100	1
10×2	1595	1595	239,250	239,250	100	100	2

Table 4. Model 2b - Constant Average Travel Time – With Dispatch

Number of Trucks	# ore loads	# of waste loads	Ore (tons)	Waste (tons)	Ore shovel utilization %	waste shovel utilization %	Avg Queue Length
4×2	983	983	147,450	147,450	61.56	61.56	0
5×2	1228	1228	184,200	184,200	76.94	76.94	0
6×2	1473	1473	220,950	220,950	92.31	92.31	0
13	1595	1595	239,250	239,250	100	99.99	0
7×2	1595	1595	239,250	239,250	100	100	0.67
8×2	1595	1595	239,250	239,250	100	100	1.677
9×2	1595	1595	239,250	239,250	100	1595	2.677

Table 5. Model 1c – Variable Average Travel Time – No Dispatch

Number of Trucks	# ore loads	# of waste loads	Ore (tons)	Waste (tons)	Ore shovel utilization %	waste shovel utilization %	Avg Queue Length
4×2	786	778	117,900	116,700	48.96	48.89	0.07
5×2	960	962	144,000	144,300	60.43	60.14	0.17
6×2	1143	1143	171,450	171,450	71.98	70.53	0.27
7×2	1305	1305	195,750	195,750	81.90	81.48	0.46
8×2	1457	1454	218,550	218,100	91.61	90.29	0.71
9×2	1563	1565	234,450	234,750	97.46	97.13	1.17
10×2	1595	1584	239,250	237,600	99.8	99.7	2.01

Table 6. Model 2c - Variable Average Travel Time – With Dispatch

Number of Trucks	# ore loads	# of waste loads	Ore (tons)	Waste (tons)	Ore shovel utilization %	waste shovel utilization %	Avg Queue Length
4×2	944	941	141,600	141,150	58.52	59.35	0.16
5×2	1155	1155	173,250	173,250	72.58	71.79	0.29
6×2	1354	1353	203,100	202,950	84.24	85.05	0.49
13	1437	1437	215,550	215,550	89.85	89.48	0.66
7×2	1508	1506	226,200	225,900	94.71	93.76	0.89
8×2	1569	1568	235,350	235,200	97.72	98.74	1.50
9×2	1583	1584	237,450	237,600	98.8	99.64	2.23

13. References

[1] Arnold, M. J. , & W, White J. (1983). Computer-Based Truck Dispatching. *World Mining, 36*(4), 53-57.

[2] Sturgul, John R. (2000). *Mine design : examples using simulation*. Littleton, CO: Society for Mining, Metallurgy, and Exploration.