Mine Road Network and Rimpull Curve Configuration Interface

Shiv Prakash Upadhyay and Hooman Askari-Nasab
Mining Optimization Laboratory (MOL)
University of Alberta, Edmonton, Canada

1. Introduction

Simulation models have proved to be an effective tool for understanding the behavior of the open pit mining systems and for “what if...?” analysis. Truck haulage, having a significant portion of the operating cost and a bottle neck in production for most operations, is required to be modeled precisely for an efficient simulation model of the production operations in open pit mines. Most of the existing simulation packages used for haulage simulation in mining industry are based on Macroscopic Simulation, which are incapable of capturing the interactions and platoon formations of trucks on the haul roads. A Microscopic modeling of truck movements is therefore necessary to capture these interactions and model the truck travel times with a relatively more precision, so as to have a better representative simulation model of the actual mining operation. Apart from taking into consideration the accelerations and decelerations of trucks based on gradient, turning angle and road condition, this approach takes into account the interactions of trucks on the haul roads, which may cause platoon formations, and affect significantly the total travel time of trucks.

To develop a simulation model, exercising control on the truck movements and capturing interactions, we need to model the haul road network and its characteristics within the simulation model and move the trucks through it based on truck characteristics and the haul road gradient, rolling resistance, turning angle and interactions with other trucks. Thus we will first describe how we can generate the haul road network within Arena using a MATLAB application with GUI. In the next section we will discuss a MATLAB application to generate the speeds of trucks on haul roads based on rimpull characteristics of the trucks and various haul road gradients. Finally we will model the trucks in Arena and discuss how to incorporate the desired characteristics into the simulation model.

2. Learning Objectives

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

- Use the RoadNetwork GUI to build haulage road network for Arena simulation
- Generate speeds of trucks based on rimpull curve characteristics and haul road gradients
- Model haulage system in Arena simulation capturing truck interactions and platoon formations on haul roads

Pourrahimian, Y., Ben-Awuah, E., and Askari-Nasab, H. (2015), Mining Optimization Laboratory (MOL), Report Six, © MOL, University of Alberta, Edmonton, Canada, Pages 250 , ISBN: 978-1-55195-356-4, pp. 241-250.

3. Concepts and Terminology (Rockwell Automation, 2010)

3.1. Transporters

Transporters are one type of device that moves entities through the system. They can be used to represent material-handling or transfer devices such as fork trucks or delivery vehicles. Transporters can also be used to model personnel whose movement is important to modeling a system, such as a nurse or a food server. When transporters are used, you provide information defining the transporter's speed and the travel distances between stations served by the transporter.

3.2. Free-path Transporters

Free-path transporters move freely between stations and are not influenced by other transporter traffic.

3.3. Guided Transporters

Guided transporters are restricted to run on fixed paths such as tracks or rails. Movement may be affected by traffic congestion from other vehicles.

3.4. Distance

The Distance module defines the distance between two stations in the distance set of a free-path transporter device. The beginning station, ending station, and distance are used to create the appropriate distance set, which is used during the simulation run by the transporter moving between the specified stations.

3.5. Network

The Network module defines a system map that a set of guided transporters will follow. A network encompasses the set of links specified in its Network Links repeat group. The parameters of a network link (e.g., length, intersections, directions), are defined in the Network Link module.

3.6. Network Link

The Network Link module defines the characteristics of a guided transporter path between an intersection pair Beginning Intersection Name and Ending Intersection Name. The Network module then references a set of network links to define a network that guides transporters follow for movement. Each link is composed of a Beginning Intersection ID, an Ending Intersection ID, and one or more Number of Zones -- each Length of Each Zone units long.

3.7. The Link Type

The link type — Unidirectional, Bidirectional, or Spur — dictates whether the simulation will allow transporters to move from Ending Intersection Name toward Beginning Intersection Name. When traveling on a unidirectional link, a guided transporter may only move from Beginning Intersection Name toward Ending Intersection Name. A bidirectional link allows vehicles to move either from Beginning Intersection Name toward Ending Intersection Name or from Ending Intersection Name toward Beginning Intersection Name. In the case of spurs, the Ending Intersection Name must be a "dead end" — not connected to the network by any links other than the spur. When the transporter arrives at the Ending Intersection Name of a spur, it does not give up control of the link zones between Ending Intersection Name and Beginning Intersection Name; instead, the transporter keeps control of the entire link to ensure that it can return to the main path intersection (Beginning Intersection Name).

3.8. Beginning and Ending Directions

The fields Beginning Direction and Ending Direction are used to define the direction of the link (in degrees) as it leaves the beginning intersection and as it enters the ending intersection; the direction entering the ending intersection defaults to the direction leaving the link's beginning intersection.

The value entered should be an integer value between 0 and 360 representing direction of travel in degrees (0 and 360 represent right or east). These directions are used in conjunction with the turning velocity of a vehicle to slow down a transporter as it turns a corner.

3.9. Number of Zones

The Number of Zones operand is used by SIMAN to determine how to move transporters through the links. If the number of zones is one, then SIMAN simply moves the transporter through the link as a single event. If the number of zones is greater than one, then SIMAN moves the transporter through the link zone by zone. The entity controlling the transporter seizes the first zone in the link before commencing movement. When it gets the first zone, it moves through the new zone, and depending upon the vehicle size and zone control policy, it releases trailing zones as they are no longer needed. When the transporter arrives at the end of the zone, it waits to seize the next zone on the link. When it gets this next zone, it commences movement through the new zone. This process repeats until the entity arrives at the end of the link.

3.10. Length of zones

The Length of Each Zone is the same for all zones in a link. The product of Number of Zones and Length of Each Zones is the total length of the link. The units used to measure Length of Zone should be consistent with all other measurements for the transporters using this link.

3.11. Velocity change factor

The Velocity Change Factor specifies a multiplier that is to be applied to the current velocity of any vehicle moving through the link only during travel through the link. The value entered is multiplied by the transporter unit’s current velocity to determine the travel velocity through the link.

4. MATLAB Application for Generating Haul Road Networks in Arena

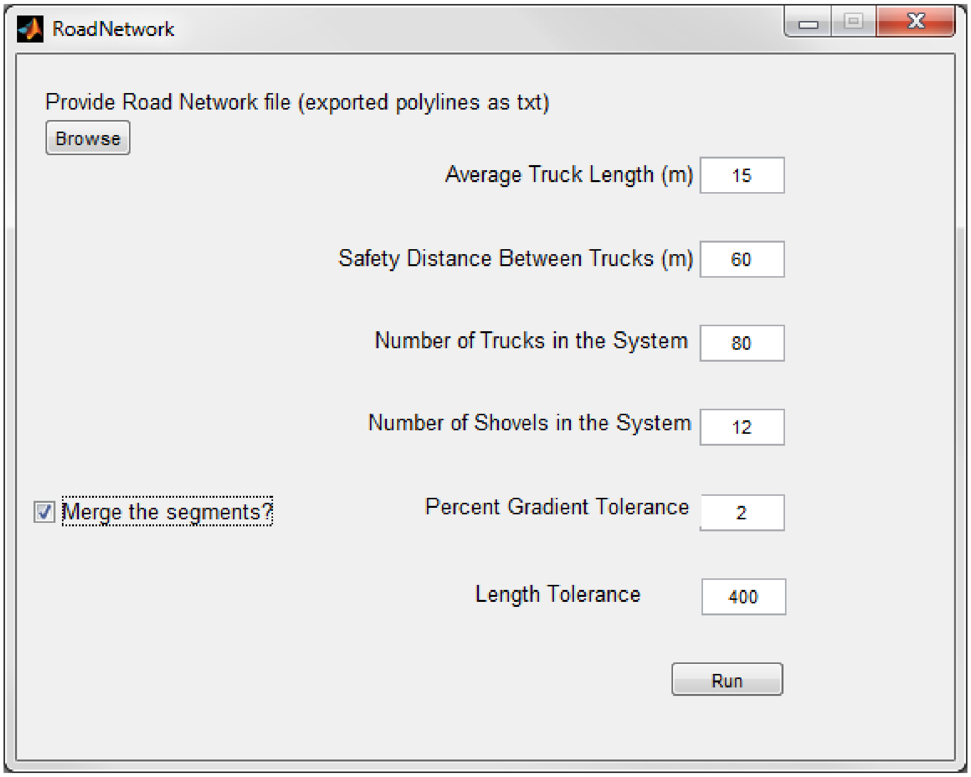
To generate the haul road network in Arena, a text file containing the road information is required. We can export the road information from within a dxf file representing the road network of the mine as polylines. Before exporting the polylines, it is necessary that:

- Polylines are connected.
- Polylines start and end at junctions or end points, i.e. any polyline must not extend beyond a junction.

The polyline data is then exported as ASCII file. The exported ASCII file must be formatted as Line Number, Line Type, Y, X, Z, Tag. It should be noted that this format is essential for the text file so that we may use it with MATLAB application, RoadNetwork.m GUI, to create road network in Arena. Line Type and Tag fields in the exported output can take any value as it is discarded during reading the file.

Create an excel file named “Config.xlsx”. Create two sheets “DumpLocations” and “Schedule” within the excel file. Provide all the dump locations, their coordinates as X, Y, Z and each dump locations capacities i.e. how many simultaneous dumps are possible at each dump location. Name the ranges as “DumpCoordinates” and “DumpCapacities”. Provide all the scheduled mining polygons and respective coordinates as X, Y, Z in the “Schedule” sheet. Name the coordinate range as “PolygonLocations”. Also provide the scheduled shovels to mine respective polygons. Name this range as “ShovelScheduled”.

Run the “RoadNetwork.m” GUI. Browse the ASCII file containing the road network information that we exported as polylines. Provide the other required data in the GUI and click on run to generate three sheets “Nodes”, “Links” and “FailureNetwork” in the Config.xlsx file. The data from these sheets will be read into Arena to generate the road network, which will be discussed in the last section.



Provide Road Network file (exported polylines as txt)

Average Truck Length (m) 15

Safety Distance Between Trucks (m) 60

Number of Trucks in the System 80

Number of Shovels in the System 12

☑ Merge the segments?

Percent Gradient Tolerance 2

Length Tolerance 400

Fig. 1. RoadNetwork GUI to generate road network data for Arena Simulation

Average truck length and safety distance between trucks is required to model the zone lengths for individual network links in Arena. Zone length provides the length of the part of a road segment that a truck (transporter unit) seizes when it is stationary. A truck unit moves through the network zone by zone by seizing the next zone and releasing the previous zone for the trailing transporters. The zone control rule thus can be defined as ‘Start’, i.e. a truck releases a zone, to be seized by a trailing truck, as soon as it is given the next required zone. This setting enables trucks not to overtake and maintain a safe following distance on the haul roads.

Check the “Merge the segments?” checkbox if the road network is very large. This will enable merging the road segments based on the length and gradient tolerance. If the difference between the gradients of two consecutive road segments is less than percent gradient tolerance or total length of the combined segment is less than length tolerance, the segments will be merged together to form a single segment.

5. MATLAB Application for Modeling Truck Speeds

In normal case, when there is no interaction between vehicles, they try to move freely on their normal driving speed. According to Bonates (1996), maximum obtainable speed by any truck can be determined by the rimpull curves generally provided by the manufacturers. He describes the rimpull as the force exerted on ground by the drive wheels to get the truck in motion. This force is generated by the torque that the engine develops and it is a function of the gear ratios.

To model the velocities of trucks in simulation, a table is created which provides a speed factor for each truck type on different total resistance haul roads. Total resistance of haul roads is the summation of haul road gradient resistance and rolling resistance. Based on the total resistance of each haul road segment, which we created in previous section, and referring to this table, we can determine the maximum possible speed of each truck type while traveling on that haul road segment.

To generate the speed factor table, a MATLAB application is created which reads in truck payloads and rimpull characteristics from a separate excel workbook and writes down the speed factors table in “Config.xlsx” file. To run this application:

- Create an excel file “RimpullRetardCurves.xlsx”.
- Create a sheet “TruckSpecification”.
- Write down the names of all truck types employed followed by their gross empty and full vehicle weights. Name the range as “Specifications”

Name of the truck type	Truck Type 1	Truck Type 2
Gross Empty Vehicle Weight	283495	278690
Gross Loaded Vehicle Weight	610082	623690

- Create two worksheets for each truck type. Name the worksheets as “Rimpull-” and “Retard-” followed by name of the truck type given in truck specifications.
- Provide the rimpull and retard curve data as Point, Rimpull/Retard Force (Kg. force), Speed (Km/h).
- Save the excel file and run the MATLAB application TruckSpeedsByRimpull.m”.

The MATLAB application will create a table in “Config.xlsx”, sheet “RimpullSpeeds”.

6. Haulage Modeling in Arena

A flowchart of the haulage simulation model, incorporating the interaction between trucks on haul roads and intersections, is given in Fig. 2. This chapter focuses only on the haulage simulation model, and a detailed modeling of the entire simulation model is not presented.

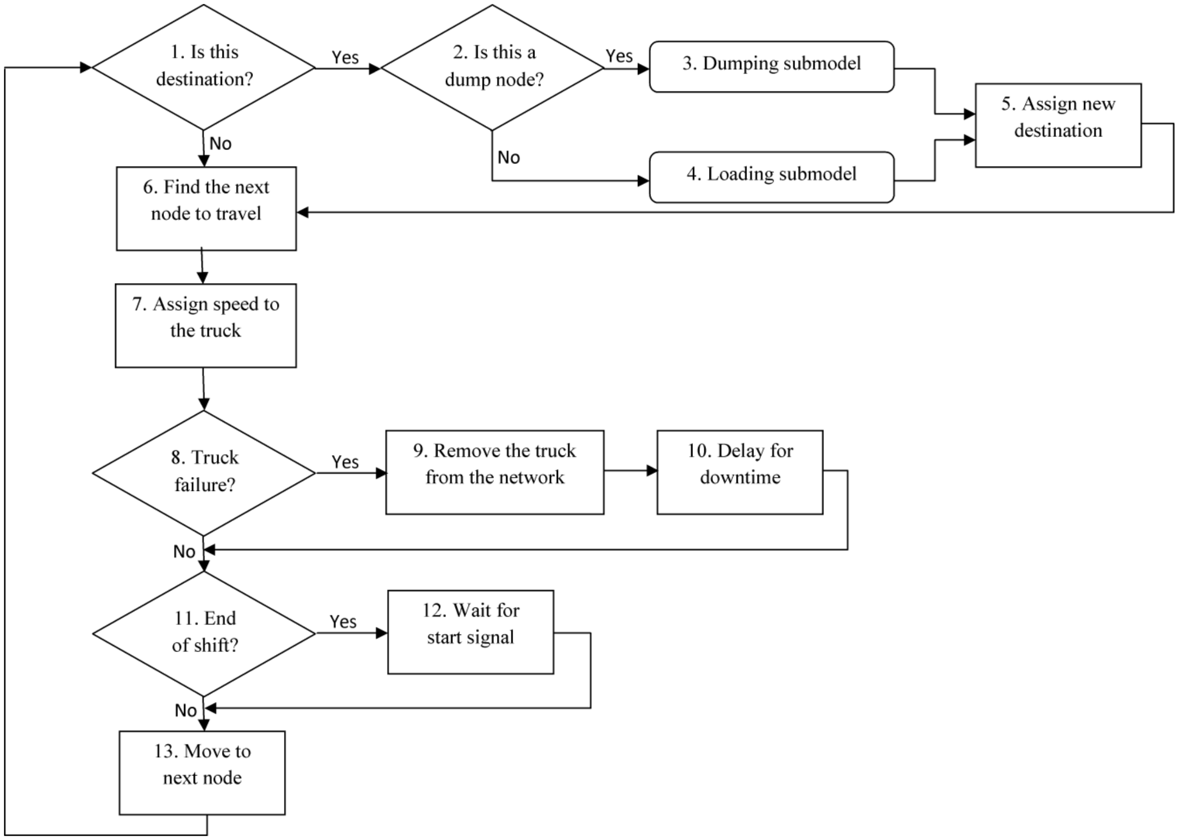


Fig. 2. Flow chart of the haulage simulation model

Prior to building the simulation model a VBA code is written which uses the data in “Nodes” sheet of “Config.xlsx” file to create stations and intersections. A Network, along with Network Links, is created between the intersections using the data in “Links” and “FailureNetwork” sheet. All the stations coupled with their corresponding intersections are written in a station set module in sequential order. This station set module is the module where entity enters back into the model after being transported to next node at step 13 in Fig. 2.

To model the truck interactions on haul roads, guided path transporters will be used which move from zone to zone by seizing next zone and releasing the occupied zone on the network link. This characteristic prohibits overtaking of trucks and models the truck interactions precisely. Moving the trucks from node to node on haul roads also provides opportunity to control the speeds of trucks based on the haul road gradient and rolling resistance.

6.1. Definitions of terminology

6.1.1. VBA Macro

- BuildModelFromExcel: A macro is written in the VBA editor of Arena to read the data in “Config.xlsx” file and build the sets, expressions, variables, transporters, network, network links and stations in the model to avoid manual work to enter each data, create haulage road network and formulate expressions in the model.

6.1.2. Entities

- entLoad: load entities associated, each associated with a transporter (truck).

6.1.3. Attributes

- atrTruckID: unique ID associated with each transporter (truck).
- atrTruckType: Truck type represented by the transporter
- atrTruckInitSpeed: Sampled speed of truck on a flat haul road (m/hr)
- atrVelocity: Velocity of the transporter on a Network Link (road segment)
- atrIsLoaded: 0 or 1 if travelling empty or loaded
- atrCurrentStation: Station name of the current node of the transporter.
- atrNextStation: Station name of the next node to travel
- atrNextLinkNum: Network Link sequence number of the next network link (road segment) to travel
- atrDumpStation: Station name of the dump station node
- atrDigStation: Station name of the node at which shovel loads the trucks

6.1.4. Variables

- v2DNetworkLinks: Variable containing information about each Network Link (road segment), i.e. start node, end node, Angle of segment from the east direction, zone lengths, Number of zones, Gradient, rolling resistance and total resistance of the road segment.

6.1.5. Station

- setStnNodes: A set of all the nodes in the haulage road network including the shovel reach node and dump nodes is created as stations associated with their corresponding intersections. It is created by the user macro “BuildModelFromExcel” written in VBA.

6.1.6. Network Link

- Network Links are created by the user macro written in VBA. Individual network links are named as “Network Link 1”, “Network Link 2” and so on. Each network link is created as unidirectional in nature between a pair of intersections, with a beginning direction, number of zones, zone length and the default velocity change factor; which are taken directly from the "Config.xlsx" file by the VBA macro "BuildModelFromExcel".

6.1.7. Network

- Network: It contains the names of all the links which were created in Network Link. Arena provides flexibility to create separate Networks using different Network Links.

6.1.8. Transporter

- Transporter: Created by the user macro written in VBA "BuildModelFromExcel". All the trucks in the system are modeled as transporters. Number of units equals number of trucks in the system. Network Name is "Network" which is the haulage road network used by trucks. A default velocity is given as 40,000 per hour. The basic distance unit is considered as meter, so the velocity is in meter per hour. Rest is left as default. Initial position of transporters is assigned sequentially on haulage road intersections.

6.1.9. Files

- filConfig: Input file "Config.xlsx" containing information about number of trucks, types, rimpull speed factors, flat haul velocities, haulage road intersections (nodes), links (segments) and haul road characteristics.

6.1.10. Advanced Sets

- setIntersections: Set containing names of all intersections in the model in sequential order
- setLinks: Set containing names of all the network links (haul road segments) in sequential order
- setTruckSpeed: Set of expressions as e2DTruckSpeed1, e2DTruckSpeed2 and so on for each truck type in sequential order
- setSpeedFactors: Set of expressions as e2DSpeedFactor1, e2DSpeedFactor2 and so on for each truck type

6.1.11. Expressions

- e2DTruckSpeed: Expressions e2DTruckSpeed1, e2DTruckSpeed2 and so on, refer to the recordsets in filConfig representing flat haul empty and loaded speeds of each truck type.
- e2DSpeedFactor: Expressions e2DSpeedFactor1, e2DSpeedFactor2 and so on, refer to the recordsets in filConfig representing empty and loaded speed factors for each truck type on various total resistances of haul roads.
- expVelocityTransporter: Expression to determine the speed of a truck on a haul road segment:

  *atrTruckInitSpeed * 1000 * EXPR (MEMBER (setSpeedFactors, atrTruckType), atrIsLoaded + 1, AINT(v2DNetworkLinks (atrNextLinkNum, 8)) + 11)*
- expNextStation: Expression to determine the next station to travel to reach the destination:

  MEMIDX(setIntersections,NEXTX(Network,INXNUM(setStnNodes(atrCurrentStation)),INXNUM(setStnNodes(atrDestStation))))

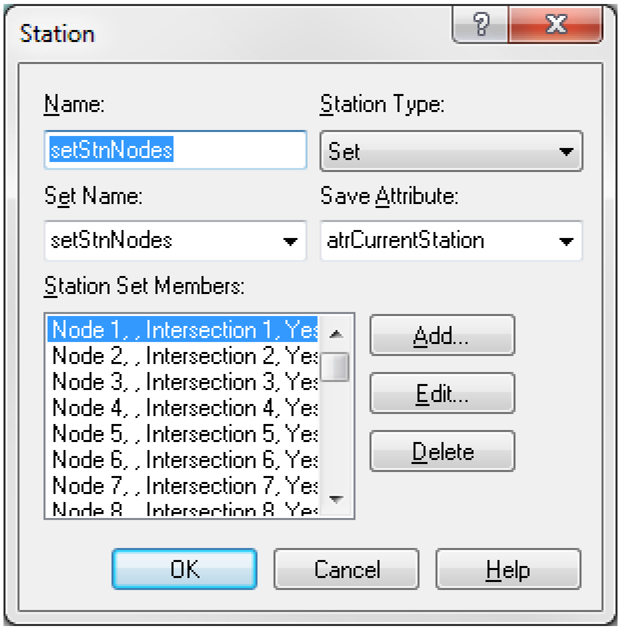
6.2. Pseudo code

- ***Check current station*** if it is not the destination station
- ***Assign*** the next station and velocity of the transporter on the next road segment
- ***Transport*** the truck through the next network link
- ***Station:*** Receive the transporter (truck) on the next station and update the current station

6.3. Step by step haulage simulation modeling

6.3.1. Station set module: setStnNodes

Create a station module and go to its properties and change the tag as “NodeStations”. When the “BuildModelFromExcel” macro will be run, all the nodes in the haul road network will be added here as stations forming part of the set “setStnNodes”.



Name: setStnNodes

Station Type: Set

Set Name: setStnNodes

Save Attribute: atrCurrentStation

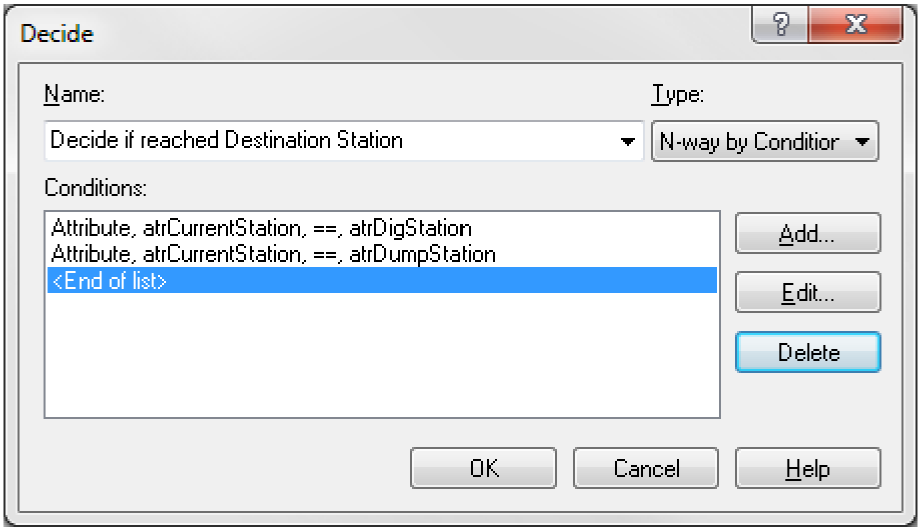
Station Set Members:

Node 1, , Intersection 1, Yes
Node 2, , Intersection 2, Yes
Node 3, , Intersection 3, Yes
Node 4, , Intersection 4, Yes
Node 5, , Intersection 5, Yes
Node 6, , Intersection 6, Yes
Node 7, , Intersection 7, Yes
Node 8, , Intersection 8, Yes

Fig. 3. Station module containing all the stations and associated intersections in the model

6.3.2. Condition module

Create a condition module to check if the current station of the transporter is its destination station. If it is the destination station of the transporter, the entity is sent into the loading or dumping sub-model which we are not discussing in this chapter. Otherwise, the entity is sent to the assign module.



Name: Decide if reached Destination Station

Type: N-way by Conditior

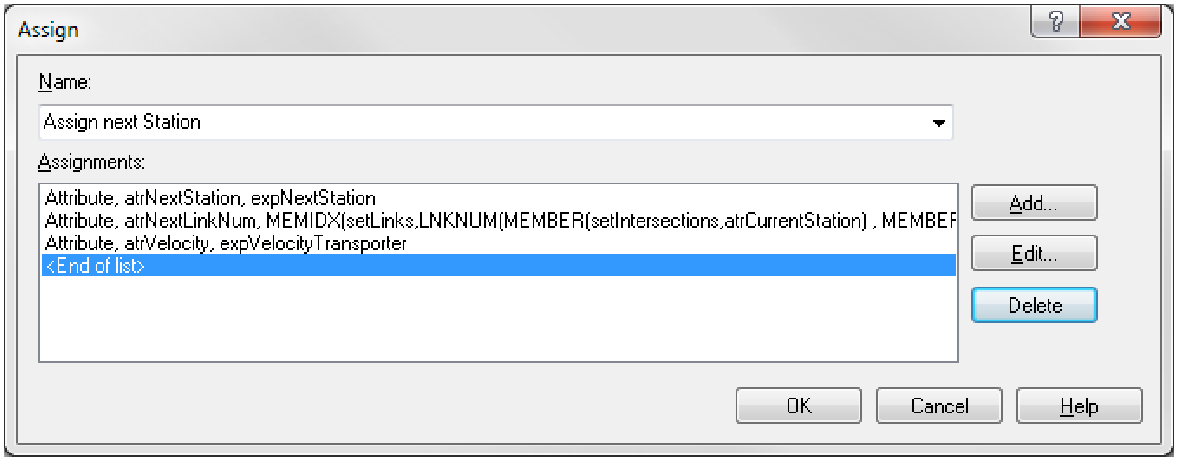
Conditions:

Attribute, atrCurrentStation, ==, atrDigStation
Attribute, atrCurrentStation, ==, atrDumpStation
<End of list>

Fig. 4. Condition module to check if the current station is the dig station or the dump station (destination)

6.3.3. Assign Module: Assign Next Station

The transporter is assigned the next node station on its path to its destination and the travelling speed.



Assign

Name:

Assign next Station

Assignments:

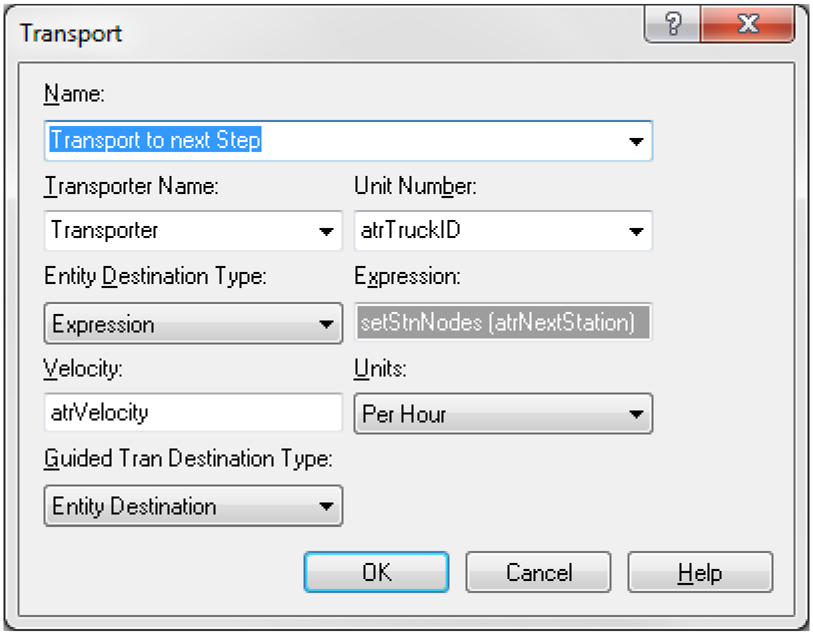
Attribute, atrNextStation, expNextStation
Attribute, atrNextLinkNum, MEMIDX(setLinks,LNKNUM(MEMBER(setIntersections,atrCurrentStation) , MEMBEF
Attribute, atrVelocity, expVelocityTransporter
<End of list>

Fig. 5. Assign Module to assign next station and speed of travel to the transporter

Name		**Assign next Station**
Assignments	Type	Attribute
	Attribute	atrNextStation
	New Value	expNextStation
Assignments	Type	Attribute
	Attribute	atrNextLinkNum
	New Value	MEMIDX(setLinks,LNKNUM(MEMBER(setIntersections,atrCurrentStation) , MEMBER(setIntersections,atrNextStation)))
Assignments	Type	Attribute
	Attribute	atrVelocity
	New Value	expVelocityTransporter

6.3.4. Transport

The transporter is sent to the next station node travelling through the next network link (haul road segment) using the transport module.



Name: Transport to next Step

Transporter Name: Transporter

Unit Number: atrTruckID

Entity Destination Type: Expression

Expression: setStnNodes (atrNextStation)

Velocity: atrVelocity

Units: Per Hour

Guided Tran Destination Type: Entity Destination

Fig. 6. Transport the load entity through the transporter to the next node station on the path to its destination.

Name	**Transport to next Step**
Transporter Name	Transporter
UnitNumber	atrTruckID
Entity Destination Type	Expression
Expression	setStnNodes(atrNextStation)
Velocity	atrVelocity
Units	Per Hour
Guided Tran Destination Type	Entity Destination

7. References

1. Bonates, E. (1996). Interactive truck haulage simulation program. *Mine Planning and Equipment Selection*, 51-57.

2. Rockwell Automation, I. (2010). Arena (Version 14.70.00004).