# SIMULATION OF TRAIN TRAFFIC AT TWO STATIONS WITH ARENA


**ABHILASH MITTAPALLI**

**M08745664**

# Table of Contents

# 1. Introduction

Indian Railways is an Indian state-owned enterprise, owned and operated by the Government of India through the Ministry of Railways. It is one of the world's largest railway networks comprising 115,000 km (71,000 mi) of track over a route of 65,808 km (40,891 mi) and 7,112 stations[1].

*1.1 Problem Statement*

Trains in India are prone to delay in arrivals due to numerous reasons, one of the major reasons for the delays is the time spent waiting for a platform at an arrival station. A platform is analogous to a terminal in an airport. For a train to arrive at a station it has to capture one of many available platforms. In some of the busiest train routes, these delays of assigning a platform could go as high as 60 minutes.

*1.2 Study and Scope*

For the scope of this project, we will simulate train traffic between Secunderabad (referred as Station1 from here on) and Vijayawada (referred as Station2 from here on), which are amongst the busiest stations in the country. About 277 trains run through these two stations in a typical day, making the mechanism of platform assignment a cumbersome process.

*1.3 Assumptions*

Simulation is based on following few assumptions about the system

- Only passenger trains are considered, no freight traffic is recorded
- Platform assignment is based solely on the arrival time of the train with a FIFO logic, other factors like type of train (Super-fast, fast) and adhoc prioritization by scheduler are not considered.
- Platforms are assumed to be available all the time, any kind of downtimes related to maintenance are not considered
- Input distributions recommended by Input Analyzer are used directly, no optimizations/segmentations are performed to improve the fitting of the data.

*1.4 Objective*

- Objective of the study is to simulate train traffic and analyze any alternatives that could improve the wait times (for platform assignment) for trains between Station1 and Station 2

*1.5 Overview*

- Section 2 discusses the modelling approach and train system to arena controls mapping
- Section 3 Focuses on Data collection, cleaning and fitting the data to distributions
- Section 4 Explains how the scenarios are modelled into an Arena Model
- Section 5 Explains running the model and analyzing the results
- Section 6 Explores the alternatives, suggests a recommendation and discusses its statistical significance
- Section 7 Gives a conclusion and suggests alternatives

## 2. Modelling Approach - Train Traffic System

This system represents train traffic between two stations namely station1 and station2. Trains can arrive and depart from any station. Based on this, we divide Trains into four different routes. Those which pass through only one of the stations, those which pass from station1 to station2 and vice-versa. A train needs to capture a platform in order to arrive at a station. Hence, we model each railway station as a *Process Module* Seizing a Platform *Resource*. Trains which are modelled as *Entities* arrive at these stations, based on a pre-computed probability, we change the entity type of these entities and route them to their corresponding stations using a *Decide* module. Creating different entity types not only helps us to route the trains but also to track the statistics at a much granular level. Any journey time between stations is modelled using a *Delay* Module

Trains arrive → Decide on the route → Travel to first Station in the route → Decide if trains travel to a second station based on their route→if yes, trains travel to second station →Trains leave system.

*Train Traffic System and Arena controls mapping summary*

Entities – Trains

Process – Railway Station

Resource – Platforms at railways stations

Journey time – Delay

Decide Module – Decides route of a train

Assign Module – Gives an entity type to the trains

## 3. Data Collection and Fitting for Input Distributions

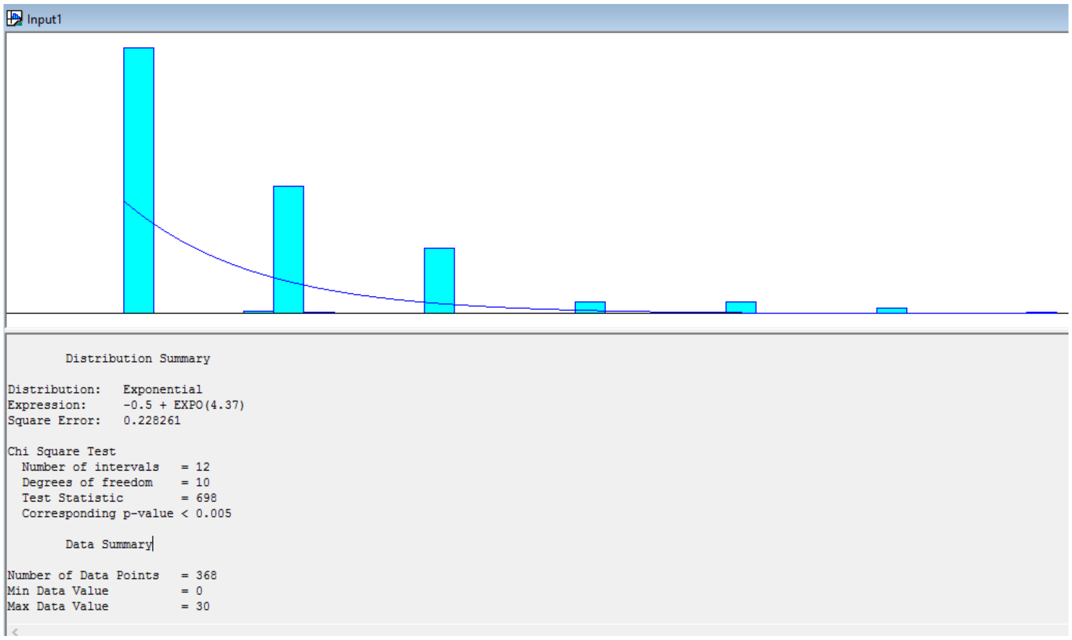
### *3.1 Data Collection and cleaning*

- Collected the data from data.gov.in (here). Data is available as a time schedule with arrival, departure times, train numbers and station names
- Computed new columns which are required for model building process. Few formulae used are as below
    i. Service Time at a station = Departure Time - Arrival Time + delays due to weather, mechanical or any other reasons
    ii. Inter Arrival Times = Arrival Time ($i^{th}$ train) –Arrival Time($i-1^{th}$ train)
    iii. Journey Times from station1 to station 2 = Arrival Time(Station2) – Departure Time(Station1)
- As discussed, we divided trains into four routes based on the direction and gave them a code to uniquely identify each route
    i. Route 1 : Trains from Station 1 to Station 2 (p=.22)
    ii. Route 2 : Trains from Station 2 to Station 1 (p=.28)
    iii. Route 3 : Trains passing through Station 1 alone (p=.33)
    iv. Route 4 : Trains passing through Station 2 alone (p=.17)
    v. Corresponding probability of occurrence of each type of train is identified. This is later used in 'Decide' module of Arena to divide the entities into different types

### *3.2 Fitting Data to Distributions*

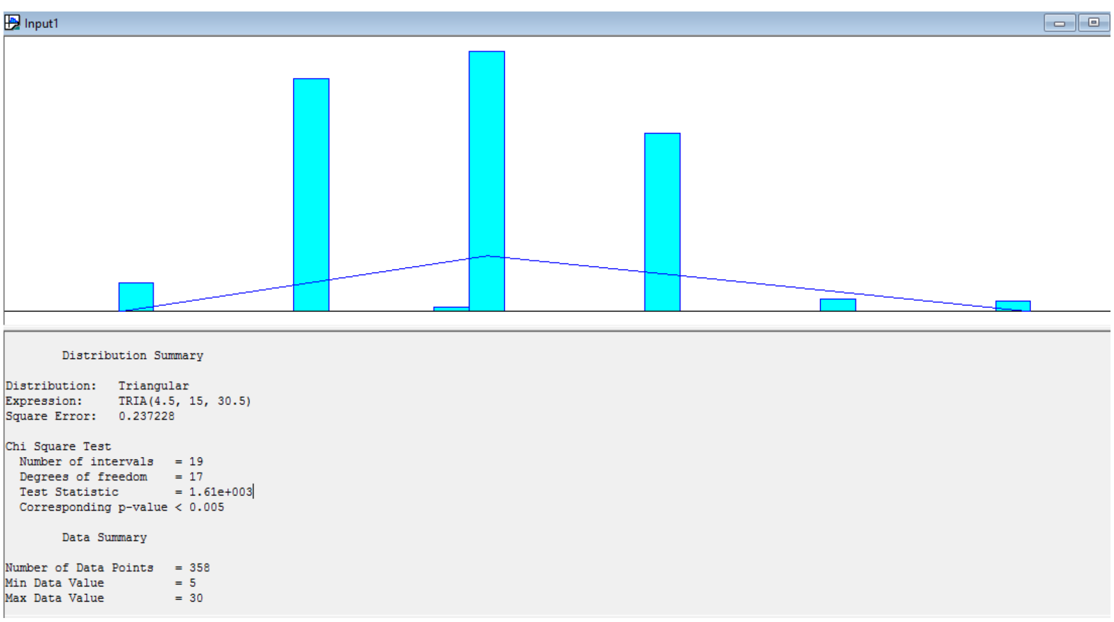
Used Input analyzer for fitting the input data to distributions, expressions obtained here can be entered directly in Arena modules. After browsing the observations as a txt file, used 'Fit All' option to get the approximate fitted distribution of the data. For the simulations, we pick the distribution which has the least square error and maximum p-value from the results.

- Inter Arrival times : -0.5+ Expo(4.37) minutes



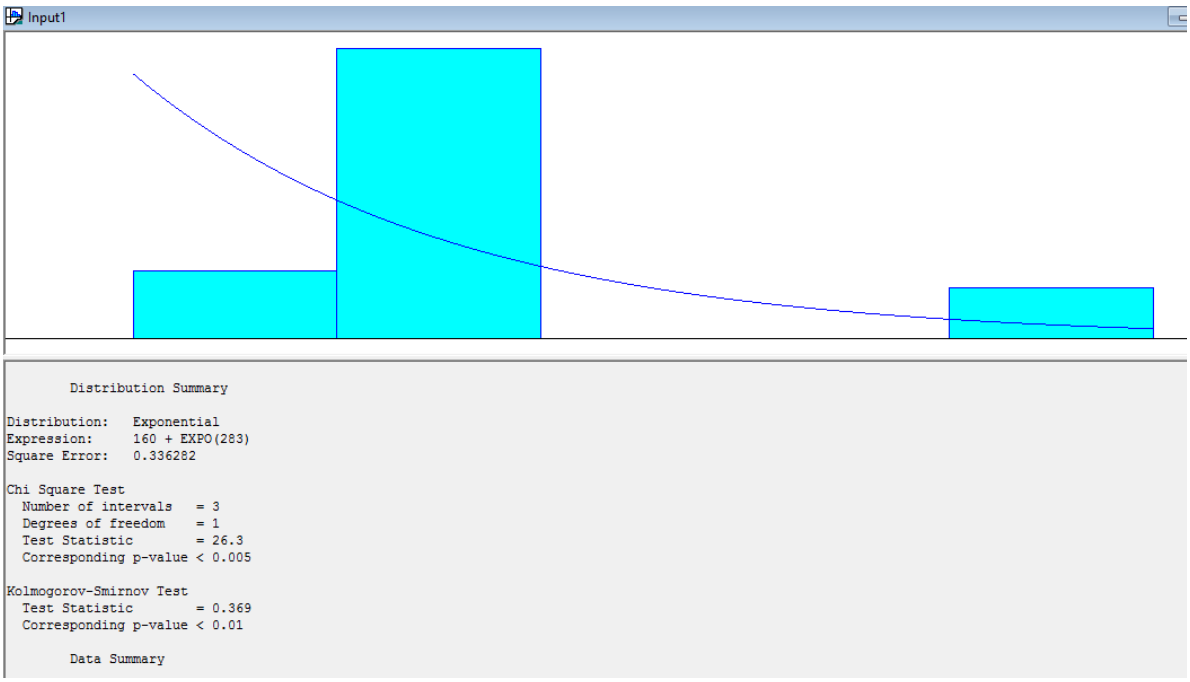

- Service Times

  - Station2 process TRIA(4.5,15,30.5) minutes



- Similarly for Station1 Process : 4.5+26*BETA(1.02,1.89) minutes

➢ Journey Times
- Station 1 to Station 2 : 160+ EXPO(283) minutes




- Similarly for Station2 to Station1: 330+795*BETA(.388,.379)

➢ Delay in Arrivals (trains have an additional delay in arrivals after the platform assignment, this can be due to lot of extraneous reasons like weather, accidents, mechanical errors etc. Trains hold the platform throughout this delay time). We do not have enough data to approximate and fit to a distribution. However, I was able to find the minimum, maximum and most frequent delay times at these stations, hence approximating it using a triangular distribution.
- Station 1 : TRIA(15,45,65) minutes
- Station 2 : TRIA(10,40,60) minutes

- Distributions Summary (minutes)

| | Trains |
|---|---|
| Inter Arrival Times | -0.5+EXPO(4.37) |

| | Station1 Process | Station2 Process |
|---|---|---|
| Service Times | 4.5 + 26 * BETA(1.02, 1.89)+TRIA(15,45,65) | TRIA(4.5, 15, 30.5)+TRIA(10,40,55) |

| | Station1 to Station2 | Station2 to Station1 |
|---|---|---|
| Journey Times | 160 + EXPO(283) | 330 + 795 * BETA(0.388, 0.379) |

# 4. Building the Model

## 4.1 Model at a glance

Based on our modelling approach, final model looks as below in figure 4.1. We would explore each module in detail in the next sections.

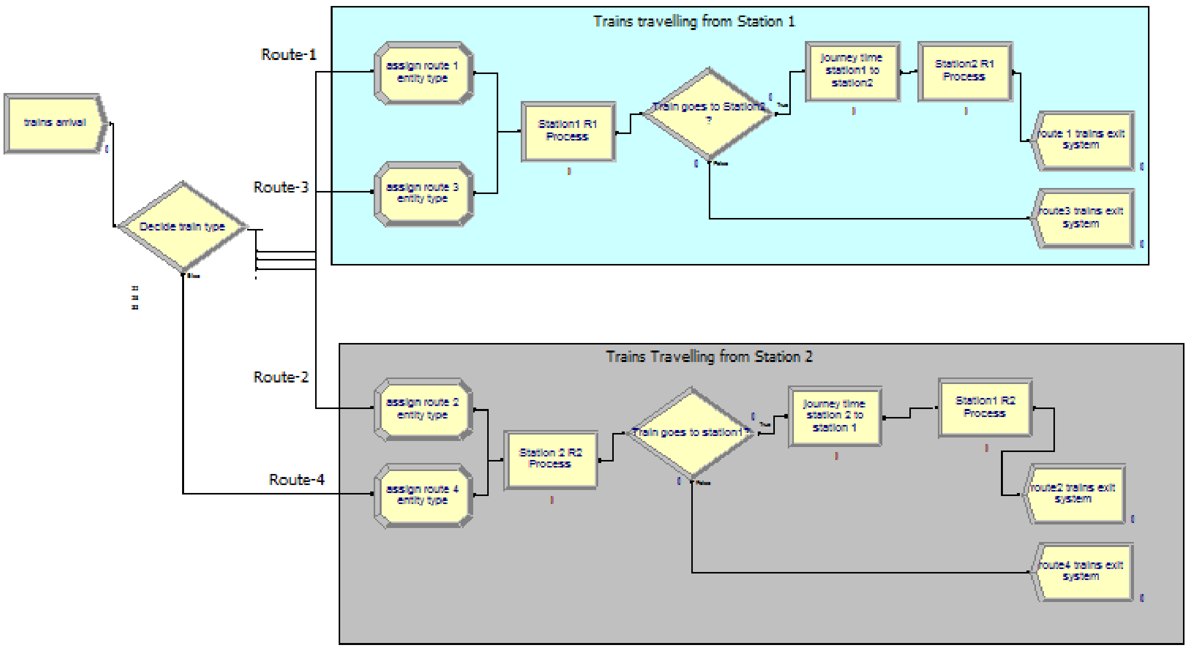
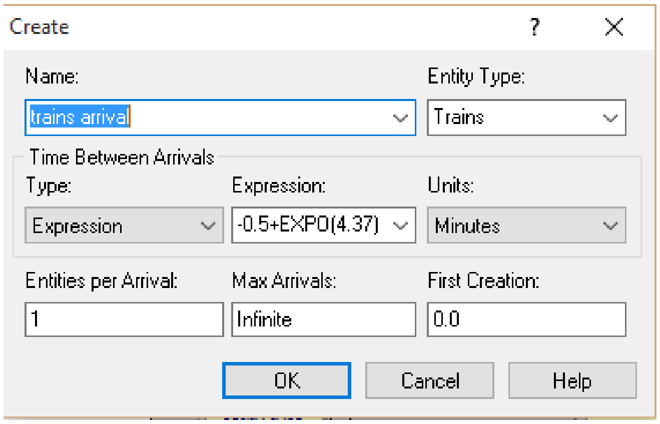



Figure: 4.1

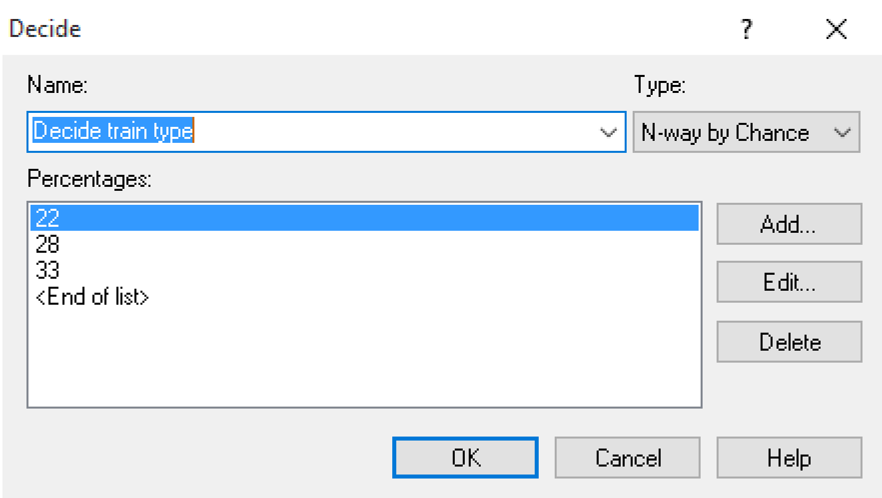
### *4.2 Create Arrivals and Direct to Stations*

Using the create module, we simulate the arrivals of trains into the system. Time between arrivals is random exponential with a mean 4.37 minutes as obtained in section-3 using Input analyzer. Entity type is trains. The remaining entries are default options



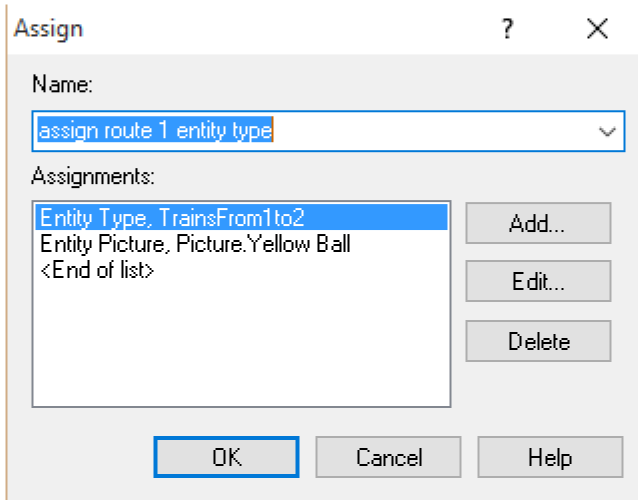

*4.3 Decide the Train type and route accordingly*

Trains can take any of the four routes as discussed. Based on the probabilities computed in section 2 we route trains to different stations. Type is 'N-way by chance' the option we use for routing based on probabilities.



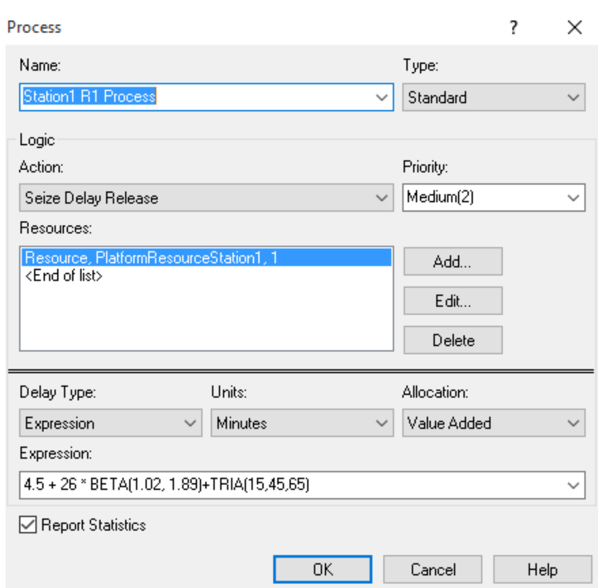

*4.4 Assign route and entity picture*

We assign entity type to trains and change the default entity picture using assign module. In the below example we assign route1 (Station1 to Station2) entity type which is TrainsFrom1to2 to the entities. We also change the entity picture for animation. Similar Process repeated for other three routes.




*4.5 Railway Station Process Modules*

Each Station identified with a number has its own service time, which is nothing but the halt time of trains at these stations plus any arrival delays. From section 3 we obtained service time distributions for each of the stations. Service time at Station 1 is obtained as 4.5 + 26 * BETA (1.02, 1.89) + TRIA (15, 45, 65) minutes. Where the triangular distribution component is the arrival delay (once the platform is assigned) and the other part is the halt time.




We repeat the process for other process module with corresponding service time. *Notice that we have two instances of Station1 and Station2 in the model. This is to simulate the bi-directional* *traffic in the system*. These two instances of the Stations seize the same platform resource, thus wouldn't create any difference in our metric measurement.

### *4.6 Resources*

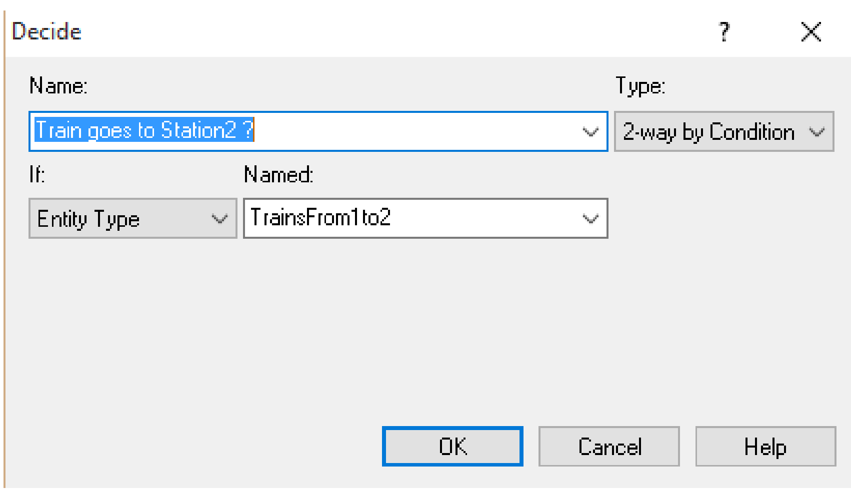
We have two types of resources in our model. Platform resources at station 1 and station2. While station1 has 11 platforms, station2 has 9 platforms. Capacity of resources can be fixed in the resources tab as shown below.

Resource - Basic Process

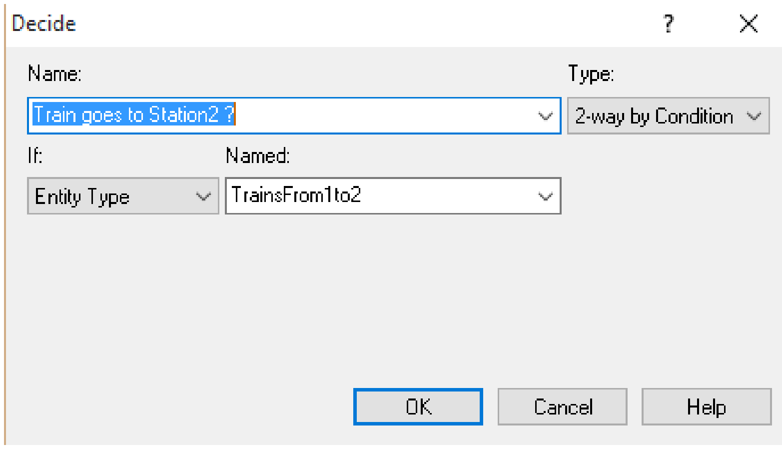
| | Name | Type | Capacity | Busy / Hour | Idle / Hour | Per Use | StateSet Name | Failures | Report Statistics |
|---|---|---|---|---|---|---|---|---|---|
| 1 | PlatformResourceStation1 | Fixed Capacity | 11 | 0.0 | 0.0 | 0.0 | | 0 rows | ☑ |
| 2 | PlatformResourceStation2 | Fixed Capacity | 9 | 0.0 | 0.0 | 0.0 | | 0 rows | ☑ |

### *4.7 Decide if trains go to second station*

This decide module routes the trains either to the exit of the system or to a second station in the train's route. We decide this based on the entity type using 2-way condition option in decide module. If Entity type is TrainsFrom1to2, trains are routed to Station2 else they leave system

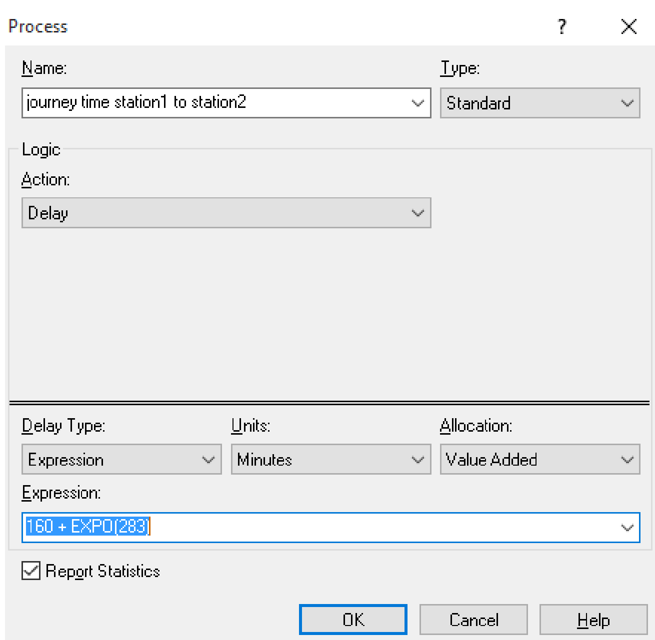



Similarly, trains with entity type 'TrainsFrom2To1' are routed to station1 and other trains exit system using another decide module



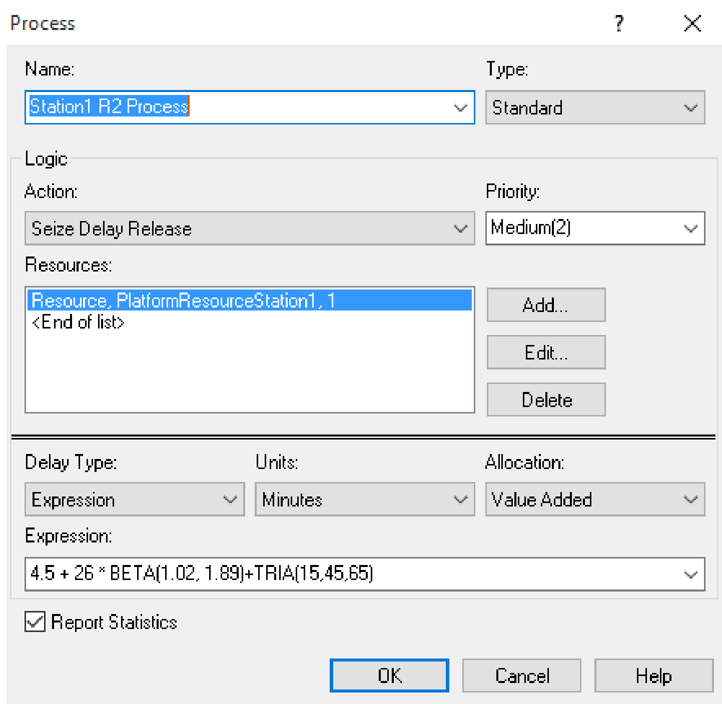

*4.8 Assign Journey Times for the trains*

Using delay module we assign journey time to trains. Based on the analysis in section 2 we arrive at an expression for journey. For trains travelling to station 2 journey time is given by 160 + EXPO (283) minutes. This expression can be directly used in delay module, we use delay module here, as there is no resource to seize during this time. Similarly, another delay module is used for modelling journey time from station2 to station1



*4.9 Process module for second Station*

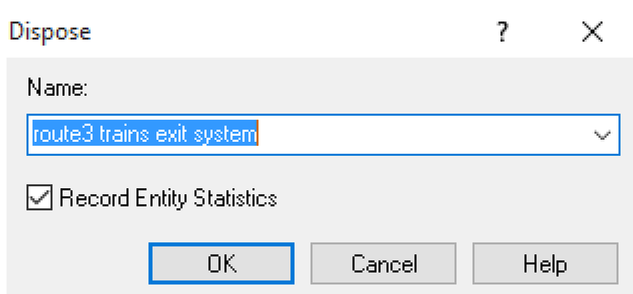
Similar to the first process module, we have another process module to model the second station in the route of a train. For entity type 'TrainsFrom2to1' second process module in the route looks as below. Service time and resource seized are the same as the process module created earlier, only difference being, this station is the second one in its route unlike the earlier module which is first in its route.




Another Process module is used to model station2 in route 'station1 to station2' which is very similar to the above created module expect with the expression for service time

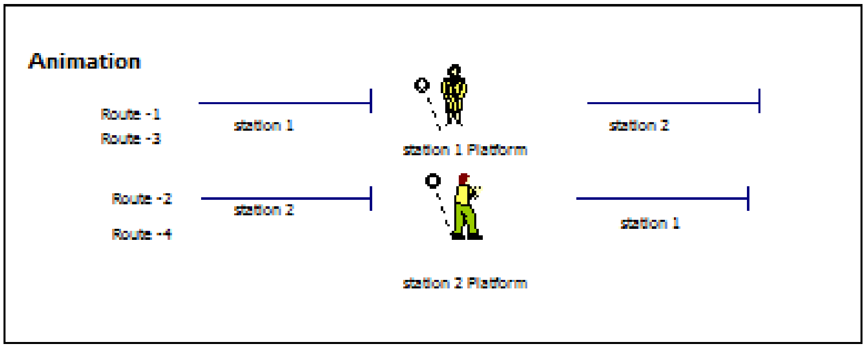
*4.10 Dispose module*

Trains that leave the system exit via the dispose modules. Record Statistics is checked to enable any metric collection on that entity. We create four such modules for each entity.



*4.11 Animations*

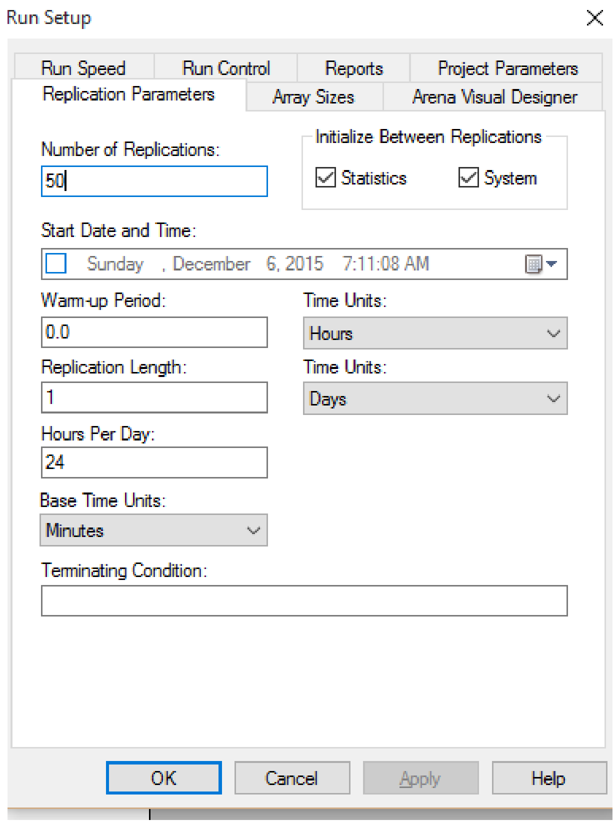
Incoming trains have an entity picture, based on the type of train we changed the entity type and picture earlier. We give resource animation for two platform resources at station1 and station2. We have different pictures to show idle and busy states of the platform. Entities in queue are shown above the queue line for each process module.



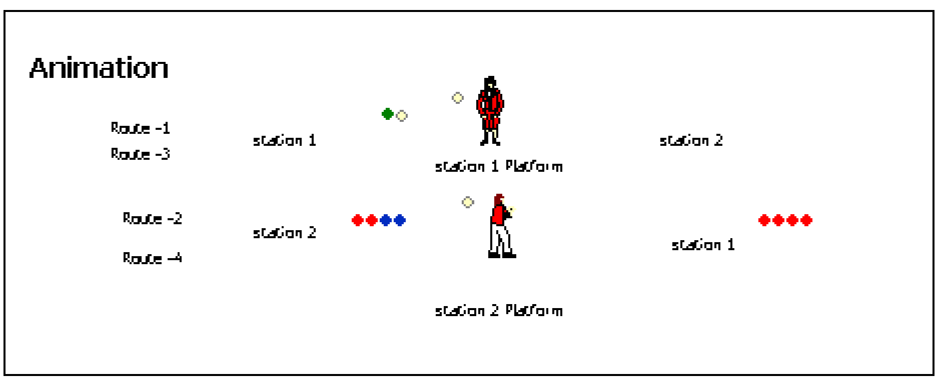

## 5. Running the Model and Results

*5.1 Running the model*

We configure the run parameters from Run→ setup as shown below. We run the simulation for one day (24 hours). Number of replications is set to 50 to obtain much reliable results




*Screenshot of the model animation at the end of 50 runs*




*5.2 Viewing Results*

Key metric we will be looking at is Entity.WaitTime, which is the average time spent by the Entities/Trains waiting in the Process queues. This time is nothing but the time spent by trains waiting for Platform assignment. Lesser the better, in terms of performance of the system. As we can see, for trains travelling from Station1 to Station2 average wait time is 23.98

minutes. And for the trains travelling in opposite direction i.e. from Station2 to Station1 average wait time before getting a platform is 25.03 minutes. In the next section we will explore alternatives that could possibly improve this performance.

**Entity**

**Time**

| VA Time | Average | Half Width | Minimum Average | Maximum Average | Minimum Value | Maximum Value |
|---|---|---|---|---|---|---|
| TrainsFrom1to2 | 462.77 | 7.52 | 421.66 | 528.80 | 233.43 | 1308.44 |
| TrainsFrom2to1 | 702.97 | 11.18 | 639.49 | 830.41 | 386.11 | 1256.48 |
| TrainsOnly1 | 55.1677 | 0.34 | 52.4765 | 58.0269 | 21.3140 | 90.9767 |
| TrainsOnly2 | 51.6288 | 0.44 | 48.4433 | 55.4148 | 19.2968 | 81.4926 |

| NVA Time | Average | Half Width | Minimum Average | Maximum Average | Minimum Value | Maximum Value |
|---|---|---|---|---|---|---|
| TrainsFrom1to2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TrainsFrom2to1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TrainsOnly1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| TrainsOnly2 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

| Wait Time | Average | Half Width | Minimum Average | Maximum Average | Minimum Value | Maximum Value |
|---|---|---|---|---|---|---|
| TrainsFrom1to2 | 23.9738 | 4.98 | 3.7970 | 98.0761 | 0.00 | 195.94 |
| TrainsFrom2to1 | 25.0353 | 4.37 | 4.0229 | 72.0000 | 0.00 | 159.83 |
| TrainsOnly1 | 10.8563 | 2.37 | 1.2457 | 41.4202 | 0.00 | 119.66 |
| TrainsOnly2 | 13.8619 | 3.16 | 2.5261 | 53.7875 | 0.00 | 174.35 |

## 6. Exploring Alternative Scenarios with Process analyzer

One obvious way to improve platform assignment time could be to increase number of platforms at the railway station. However, we cannot be sure on where to increase the resources, given that we have a choice of two stations.

Using Process Analyzer, we can simulate the increase in resources and observe the average wait time of the trains. As shown below, we compared two other alternatives where we increase platform resources by 2 units at these stations (one at a time). We can observe that increasing platforms in Station2 yields better wait times for trains travelling from station1 to station2.

Control Variables:

- Platform resource at station1
- Platform resource at station2

Response Variables:

- Wait time of Trains from station 1 to station 2 (lesser the response value better is the performance of system)

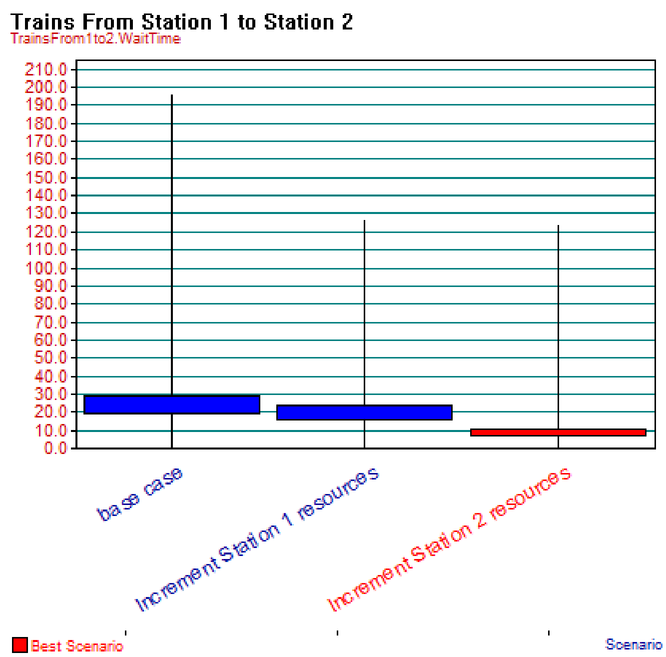
Scenarios

1. Scenario 1 - Original Model
2. Scenario 2 - Increase Platform resources at Station1
3. Scenario 3- Increase Platform resources at Station2

| | Scenario Properties | | | | Controls | | Response |
|---|---|---|---|---|---|---|---|
| | S | Name | Program File | Reps | PlatformResourceStation1 | PlatformResourceStation2 | TrainsFrom1to2.WaitTime |
| 1 | | base case | 96 : Mittapalli | 50 | 11.0000 | 9.0000 | 23.974 |
| 2 | | Increment Station1 resources | 96 : Mittapalli | 50 | 13.0000 | 9.0000 | 19.759 |
| 3 | | Increment Station2 resources | 96 : Mittapalli | 50 | 11.0000 | 11.0000 | 8.784 |

*Figure 6.1*

We can further explore statistical significance of the results with the help of plot option in PAN. As shown below in Box and Whisker plot the scenario in Red, which is increasing resources at Station 2 is statistically significant at 95% significance level compared to other two scenarios.




When we increase resources at station2, we can attain a total wait time with 95% confidence interval as (7.29, 10.28) minutes. Compared to the original base case (mean ~ 24 minutes) and scenario two (mean ~19.7 minutes), this interval is smaller. Hence, I would recommend an increase in platforms in Station2 for better wait times (before platform assignment) of trains travelling from station1 to station2.

## 7. Conclusion

Using Arena we were able to simulate train traffic between two stations. Running the model gave us an insight on the delays that trains incur while waiting for a platform assignment. Process analyzer provided us a way to compare various scenarios and evaluate a best scenario. Thus, from our simulation results, I would recommend an increase in platform capacity at station2 for reducing delays of trains running from Station1 to Station2.

## 8. References

1. https://en.wikipedia.org/wiki/Indian_Railways
2. https://data.gov.in/catalog/indian-railways-train-time-table-0#web_catalog_tabs_block_10
3. http://highered.mheducation.com/sites/0073401315/information_center_view0/index.html
4. http://inpro4u.com/