# Low-Comotovation: System Design Document

Michael Ghaben

## Contents

## 0.1 Versioning & Authorship

Version 0.1
Low-Comotovation ©
Software Design Specification: Low-Comotovation
Status: Preliminary Release: Software Design Review

## 0.2 References

During the development of this document, IEEE 1016 was utilized.

## 0.3 Purpose

This document will specify the architecture and design of the Low-Comotovation train system. It shall discuss the structural and design and considerations of the train system and the accompanying subsystems of the train system. It shall also detail design considerations in vital subsystems.

## 0.4 Stakeholders & Concerns

The stakeholders of this document are anticipated to be the following:

- Future Design Teams: Future design teams are anticiapted to utilize this document to guide their usage of the track controller system
- Pittsburgh Rail Company: The rail company utlizing the Software Design Specification (SDS) to guide the development of physical systems associated with the software

Future design teams associated with the continued development beneift from increased documentation of the original system by allowing for more efficient software design procedures in future revision by potentially unrelated developers.
The benefits to the Pittsburgh Rail Company from a detailed software design specification are twofold. First, a detailed SDD provides developers of railway hardware the information required to produced a paired system. Second, a documented SDD allows the Pittsburgh Rail Company to evaluate the designs ability to meet specifications for vitality.

# 1 Introduction

To ensure safe, predictable, and reliable operation of the system, there are three primary considerations:

1. *Vitality:* Vitality of a system within this document referse to a safety-critical system.

2. *Testability:* Any system implemented must be easily tested to ensure reliability

3. *Modularity:* Any system designed must reuse code wherever possible

# 2 System Design Use Cases

In this section, we detail the use cases of each subsystem. The use case of each subsystem is accompanied by brief descriptions of the use cases.

## 2.1 Track Model

In this subsection, the use cases of the train model are provided.

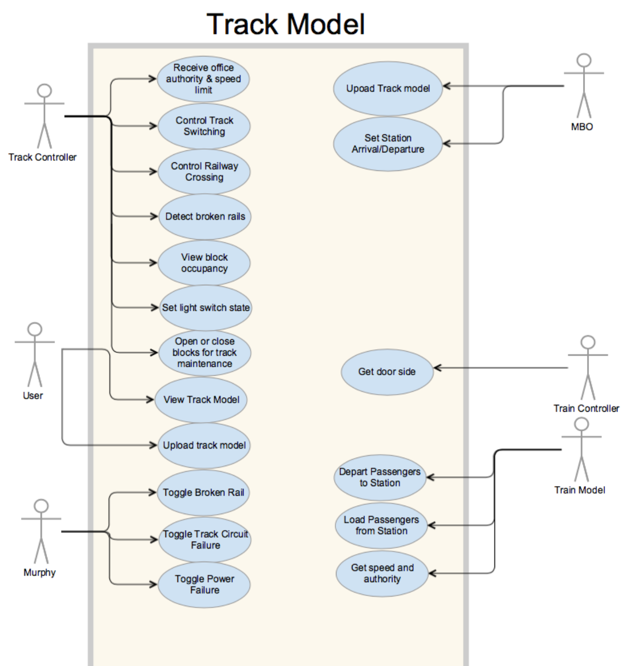


Figure 1: Track model use case diagram

## 2.2 Track Controller

In this subsection, the use cases of the track controller are provided.

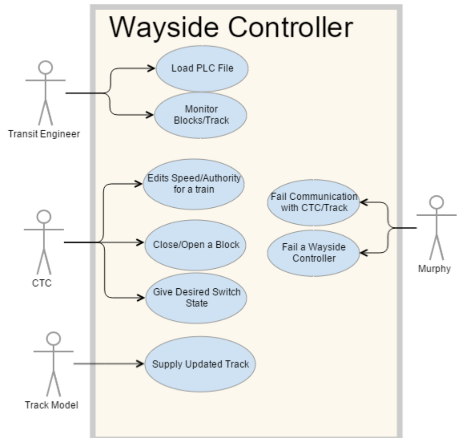


Figure 2: Track controller use case diagram

## 2.3 Train Model

In this subsection, the use cases of the train model are provided.

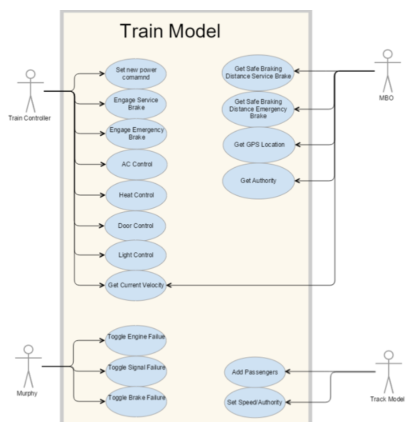


Figure 3: Train model use case diagram

Table 1: Set New Power Command

| | |
|---|---|
| Actors | Train Controller |
| Description | Train Controller will set a new power command based on the current velocity of the train and the new setpoint speed set by the driver. This power command will be used to determine the force applied to the train and thus compute the new current velocity. |
| Data | Power Command issued to the train |
| Stimulus | When a setpoint speed is provided to the train controller, a Power command is computed using the current velocity and sent to train |
| Response | New current velocity is returned to actor at the end of the computation. |
| Comments | |

Table 2: Engage Service Brake

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will engage or disengage the service brake in order to slow down or stop the train for any given reason. Once engaged the power command will be set to zero and the train will begin to decelerate |
| Data | Service Brake command |
| Stimulus | Service brake will be engaged under the following conditions:<br>1) Service brake button is manually pressed by the driver via the train controller<br>2) Failure occurs in the train that requires the train to stop, this will engage the service brakes unless failure is caused by service brakes<br>3) Train is set to slow down and service brakes are applied to reduce speed |
| Response | Service brake status is set to engaged and train begins to decelerate at service brake deceleration rate. |
| Comments | The service brake can either posses the status of on, off , or failure. |

Table 3: Engage Emergency Brake

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will engage or disengage the emergency brake in order to slow down or stop the train for any emergencies that may occur. Once engaged the power command will be set to zero and the train will begin to decelerate |
| Data | Emergency Brake command |
| Stimulus | Emergency brake will be engaged under the following conditions:<br>1) Emergency brake button is manually pressed by the driver or passenger via the train controller<br>2) Failure occurs in the service brakes and the emergency brakes are required to stop the train |
| Response | Emergency brake status is set to engaged and train begins to decelerate at emergency brake deceleration rate. |
| Comments | The Emergency brake can either posses the status of on or off. For this model we are assuming that the emergency brakes never fail |

Table 4: Air Conditioning (AC) Control

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will activate or deactivate the Air conditioning unit onboard the train to decrease the current temperature of the train. |
| Data | Air conditioning command |
| Stimulus | The air conditioning will be turned on or off by the train controller. This will either be performed manually by the driver using a button or automatically by the train controller based on current temperature and thermostat setting. |
| Response | AC control set to on will result in a gradual decrease of the current train internal temperature. |
| Comments | The AC can either posses the status of on, off, or failure. |

Table 5: Heater Control

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will activate or deactivate the heating unit onboard the train to increase the current temperature of the train. |
| Data | Heater command |
| Stimulus | The heating unit will be turned on or off by the train controller. This will either be performed manually by the driver using a button or automatically by the train controller based on current temperature and thermostat setting. |
| Response | Heater control set to on will result in a gradual increase of the current train internal temperature. |
| Comments | The heater can either posses the status of on, off, or failure. |

Table 6: Door Control

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will open and close the doors on the left and right side individually using individual commands for each side. |
| Data | Left door command, Right door command |
| Stimulus | The left or right doors will be opened or closed by the train controller. This will either be performed manually by the driver using a button or automatically by the train controller upon arrival and departure at each station. |
| Response | If the right door command is passed, all doors on the right side are opened. If the left door command is passed, all doors on the left side are opened. |
| Comments | The Left and Right doors can either posses the status of open, closed, or failure. |

Table 7: Light Control

| | |
|---|---|
| Actors | Train Controller |
| Description | Train controller will turn the interior lights onboard the train on and off based on time of day and location of train (e.g. within tunnel or not) |
| Data | Interior Light command |
| Stimulus | The lights will be toggled on and off by the train controller. This will either be performed manually by the driver using a button or automatically by the train controller based on time of day and upon entering and exiting a tunnel |
| Response | If the light command is passed, all lights onboard the train are turned on. |
| Comments | The interior lights can either posses the status of on, off, or failure. |

Table 8: Get Current Velocity

| | |
|---|---|
| Actors | Train Controller, MBO |
| Description | A call will be made to request the current velocity of the train and this will be passed back to the actor which required it. The train controllor will request the current velocity in order to compute the power command to send to the train model. The MBO will request the current velocity in order to compute the variation between the suggested speed and the actual speed of the train. |
| Data | Current Velocity value |
| Stimulus | A request will be sent to the train model to obtain the current velocity of the train at that given moment |
| Response | The current velocity of the train will be returned to the caller in MPH. |
| Comments | |

Table 9: Toggle Engine Failure

| | |
|---|---|
| Actors | Murphy |
| Description | Murphy is able to toggle the engine failure status in order to distrupt the train's engine. Once engaged the train will be required to stop until the issue is resolved. |
| Data | Engine Failure command |
| Stimulus | A command will be sent to the train model from the Murphy console to toggle the failure status of the train's engine. |
| Response | The engine failure status will be toggled as a response to the command. When an engine failure occurs the service brakes are also engaged to bring the train to a stop until issues are resolved. |
| Comments | The engine failure status will toggle between failure, and non-failure. |

Table 10: Toggle Signal Failure

| | |
|---|---|
| Actors | Murphy |
| Description | Murphy is able to toggle the signal failure status in order to distrupt the train's signaling and communication abilities. Once engaged the train will be required to stop until the issue is resolved. |
| Data | Signal Failure command |
| Stimulus | A command will be sent to the train model from the Murphy console to toggle the failure status of the train's signaling system. |
| Response | The signal failure status will be toggled as a response to the command. When a signal failure occurs the service brakes are also engaged to bring the train to a stop until issues are resolved. |
| Comments | The signal failure status will toggle between failure, and non-failure. |

Table 11: Toggle Brake Failure

| | |
|---|---|
| Actors | Murphy |
| Description | Murphy is able to toggle the brake failure status in order to distrupt the train's service brake. Once engaged the train will be required to stop until the issue is resolved. |
| Data | Brake Failure command |
| Stimulus | A command will be sent to the train model from the Murphy console to toggle the failure status of the train's service brake |
| Response | The brake failure status will be toggled as a response to the command. When a service brake failure occurs the emergency brakes are also engaged to bring the train to a stop until issues are resolved. |
| Comments | The brake failure status will toggle between failure, and non-failure. |

Table 12: Get Safe Braking Distance (Service Brake)

| | |
|---|---|
| Actors | MBO |
| Description | In order to better determine the train's footprint the MBO will call to obtain the safe braking distance of the Train. This will be the distance required to bring the train to a complete stop using the service brake deceleration rate. This distance will vary based on the number of passengers on board the train and the current velocity of the train. |
| Data | Safe Braking Distance for Service Brake |
| Stimulus | Command will be requested from the MBO to get the current safe braking distance using the service brakes which would be computed based on the current velocity and mass of the train. |
| Response | The safe braking distance using the service brakes will be returned to the MBO. |
| Comments | |

Table 13: Get Safe Braking Distance (Emergency Brake)

| | |
|---|---|
| Actors | MBO |
| Description | In order to better determine the train's footprint the MBO will call to obtain the safe braking distance of the Train. This will be the distance required to bring the train to a complete stop using the emergency brake deceleration rate. This distance will vary based on the number of passengers on board the train and the current velocity of the train. |
| Data | Safe Braking Distance for Emergency Brake |
| Stimulus | Command will be requested from the MBO to get the current safe braking distance using the emergency brakes which would be computed based on the current velocity and mass of the train. |
| Response | The safe braking distance using the emergency brakes will be returned to the MBO. |
| Comments | |

Table 14: Get GPS Location

| | |
|---|---|
| Actors | MBO |
| Description | The MBO will elect to receive the current GPS location to determine the train's current location to the nearest meter. This will be determined by calculating the distance traveled by the train and compute the distance into the current block to return to the MBO |
| Data | Current Block, Distance Into block |
| Stimulus | Command will be requested from the MBO to get the current GPS location from the train |
| Response | GPS location will be returned providing the current block the train is in as well as the distance into that current block to the nearest meter. |
| Comments | |

Table 15: Get Authority

| | |
|---|---|
| Actors | MBO |
| Description | The MBO will request to receive the current Authority of the given train. This will be used in conjuction with the suggested authority to determine the variation between suggested authority and actual authority for the train. |
| Data | Current Authority |
| Stimulus | Command will be requested from the MBO to get the current Authority from the train |
| Response | Current authority will be returned for that given train |
| Comments | |

Table 16: Add Passengers

| | |
|---|---|
| Actors | Track Model |
| Description | The track model will randomly generate a number of passengers to wait at a station then upon arrival to a station a random number of passengers will board based on space avaliable on the train. This number will be sent to the train model to modify passenger count and mass of train based on capacity. |
| Data | Number of passengers boarding |
| Stimulus | Command will be requested from the MBO to get the current Authority from the train |
| Response | Based on space on board, a random number of passengers between 0 and amount of space will be passed to the train model |
| Comments | |

Table 17: Set Speed/ Authority

| | |
|---|---|
| Actors | Track Model |
| Description | The track model will pass the speed and authority to the train model. This speed and authority will then be passed to the train controller with no variation. |
| Data | Speed, Authority |
| Stimulus | Command will be sent to train model with speed and authority |
| Response | Speed and authority will be passed to train controller. |
| Comments | |

Table 18: Set Current Block

| | |
|---|---|
| Actors | Track Model |
| Description | The track model will pass the current block the train is on as the train enters each new block area. This current block object will provide the train with the block's grade as well as its length, to be used by the train's GPS |
| Data | Current Block |
| Stimulus | Command will be sent to train model with current block |
| Response | Block length will be extracted for train GPS, and Block grade will be extracted for train movement calculations |
| Comments | |

## 2.4 Train Controller

In this subsection, the use cases of the train controller are provided.

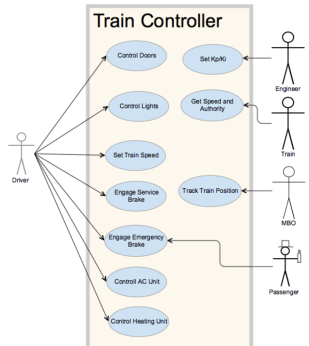


Figure 4: Train controller use case diagram

## 2.5 Moving Block Overlay

In this subsection, the use cases of the Moving Block Overlay (MBO) are provided.

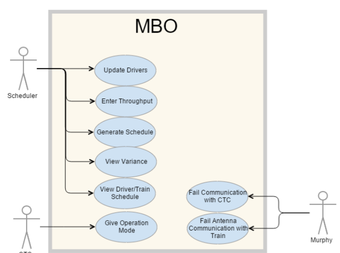


Figure 5: Train controller use case diagram

Table 19: Update

| | |
|---|---|
| Actors | Scheduler |
| Description | The Scheduler is able to update the list of drivers. This will change whether or not a driver is able to be scheduled. |
| Data | filename |
| Stimulus | Click drivers button |
| Response | Loops through a CSV file to add all the drivers to the list of drivers. When adding a driver, a driver object will be created with the entered properties. This object will then be added to the Driver Schedule where it can be accessed as part of the list. |
| Comments | There will be a default file so that it can be saved between sessions. |

Table 20: Enter Throughput

| | |
|---|---|
| Actors | Scheduler |
| Description | The Scheduler enters the number of trains they would like to be on the track at a certain point in time. |
| Data | number of trains |
| Stimulus | Click submit button |
| Response | The number of trains is entered by the scheduler. This is used to generate both the train and driver schedules for both MBO and FB modes. |
| Comments | |

Table 21: View Train/Driver Schedule

| | |
|---|---|
| Actors | Scheduler, CTC |
| Description | Scheduler can see a list of all trains, as well as their station arrival times. Scheduler can see a list of all current drivers, as well as their corresponding break times and current train. |
| Data | train ID, arrival times, driver name, ID, break times |
| Stimulus | Updates triggered by clock |
| Response | Two tables will be displayed, one for the train schedule, and one for the driver schedule. The train schedule will list IDs as the rows and station names as the columns. Each cell will contain the time that train will arrive at that station. The driver schedule will what train they are on at what times. It will also show whenever they start and stop work and when they are on breaks. |
| Comments | The table that is displayed will automatically update itself when triggered by the clock. |

Table 22: View Variance

| | |
|---|---|
| Actors | Scheduler |
| Description | Scheduler can see a list of all trains, as well as their corresponding speed and current position. The suggested speed and authority will be displayed as well as the variance between the two. |
| Data | train ID, speed, suggested/actual position/authority, variance |
| Stimulus | Updates triggered by clock |
| Response | In Fixed Block mode the current block will have to be kept track of based on past block occupancy. In MBO mode the position can be gotten through GPS. |
| Comments | In Fixed Block mode the position is denoted as the current block. In MBO mode the position is denoted as the current block and the distance into that block. |

Table 23: Generate Schedules

| | |
|---|---|
| Actors | Scheduler |
| Description | When required a schedule will be generated based on the input data. This will then be displayed for the scheduler/CTC. It is used to dispatch trains and calculate a path for a train. |
| Data | number of trains, track data |
| Stimulus | On launch, change in number of drivers, clock triggered |
| Response | A schedule will be generated for trains and drivers. It will have to take into account the mode of operation (MBO or FB), speed limits, track occupancy, drivers break times, and other variables. |
| Comments | Can only happen in automatic mode - schedule will be either fixed block or MBO depending on dispatcher's selection of mode. |

Table 24: Give Operation Mode

| | |
|---|---|
| Actors | CTC |
| Description | The CTC sends the mode of operation whenever it is changed. |
| Data | mode |
| Stimulus | CTC changes the mode. |
| Response | The mode is updated in the MovingBlockOverlay class. Any shutdown procedures to switch between modes are performed. |
| Comments | The default mode will be manual. |

Table 25: Fail Communication with CTC

| | |
|---|---|
| Actors | Murphy |
| Description | Murphy breaks communication between CTC and MBO. |
| Data | communication failure |
| Stimulus | CTC clicks Fail Communication with MBO button. |
| Response | Since scheduling will be unavailable without communication with the MBO, the CTC will be forced into manual mode and let the dispatcher know with a message. |
| Comments | |

Table 26: Fail Communication with Train

| | |
|---|---|
| Actors | Murphy |
| Description | Murphy breaks communication between Train and wayside. |
| Data | communication failure |
| Stimulus | Click Fail Communication with Train button. |
| Response | The MBO can no longer receive the GPS position of individual trains and is therefore unable to safely operate in MBO mode. So a transition must be made to either Fixed Block mode or to manual mode. |
| Comments | |

### 2.6 Centralized Train Control

In this subsection, the use cases of the Centralized Train Control (CTC) are provided.

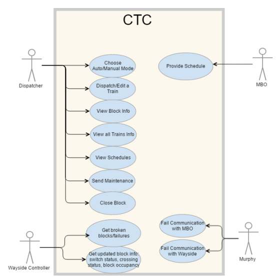


Figure 6: Centralized Train Control use case

## 3 Class Diagrams

In this section, we detail the use class diagrams for each subsystem..

## 3.1 Track Model

## 3.2 Track Controller

## 3.3 Train Model

## 3.4 Train Controlled

## 3.5 Moving Block Overlay

## 3.6 Centralized Train Controller

# 4 Sequence Diagrams

In this section, we detail the sequence diagrams each subsystem.

## 4.1 Track Model


Figure 7: Add Passengers Use Case Diagram

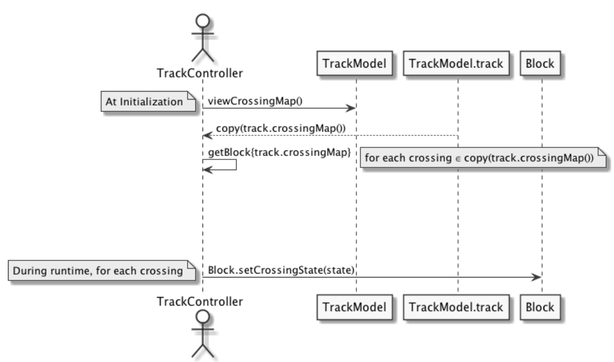


Figure 8: Toggle Crossing State Use Case Diagram

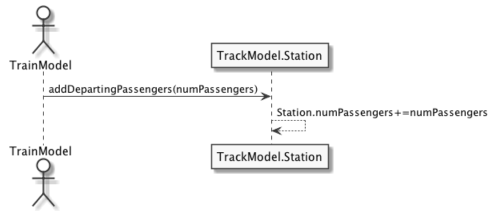


Figure 9: TDeparting Passengers to Station Use Case Diagram

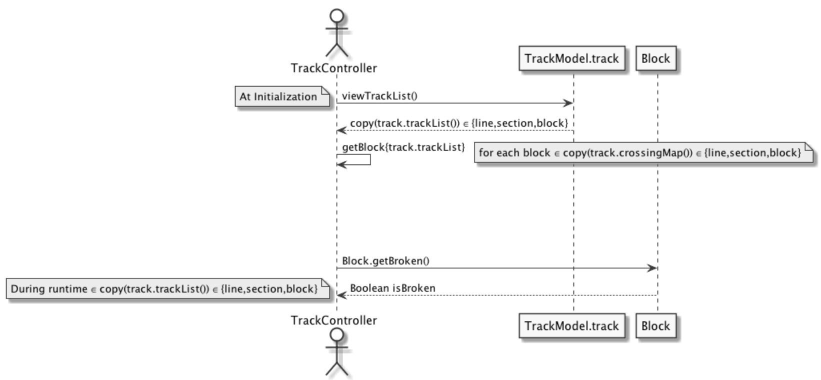


Figure 10: Detect Broken Rail Use Case Diagram

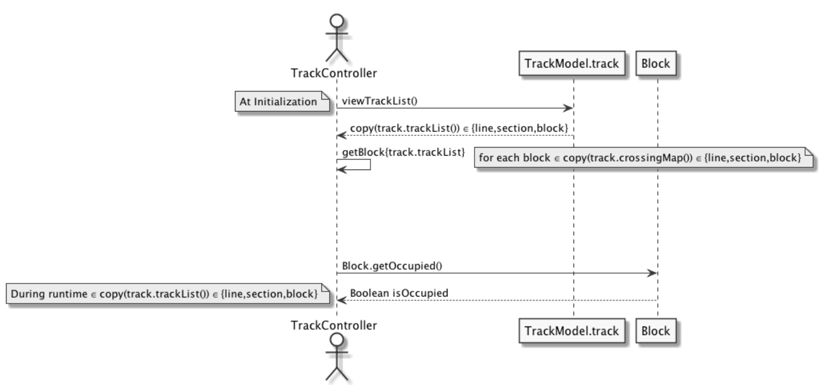


Figure 11: Detect Block Occupancy use Case Diagram

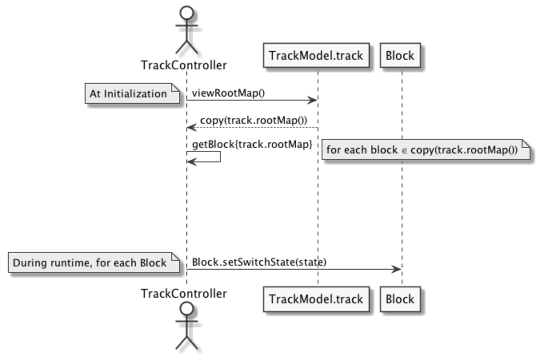


Figure 12: Toggle Switching Use Case Diagram

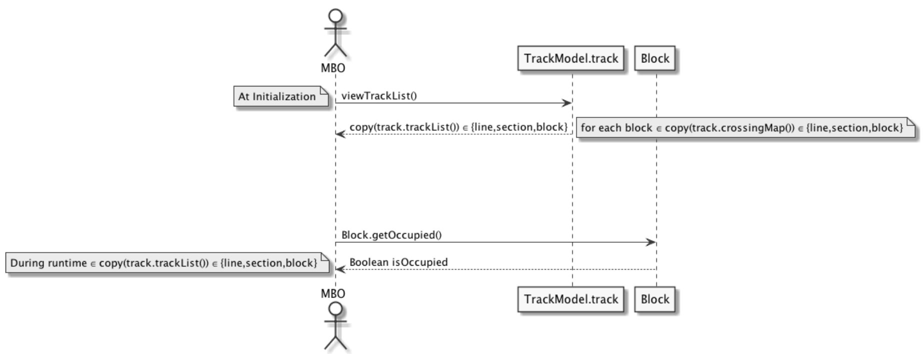


Figure 13: View Block Occupancy Use Case Diagram

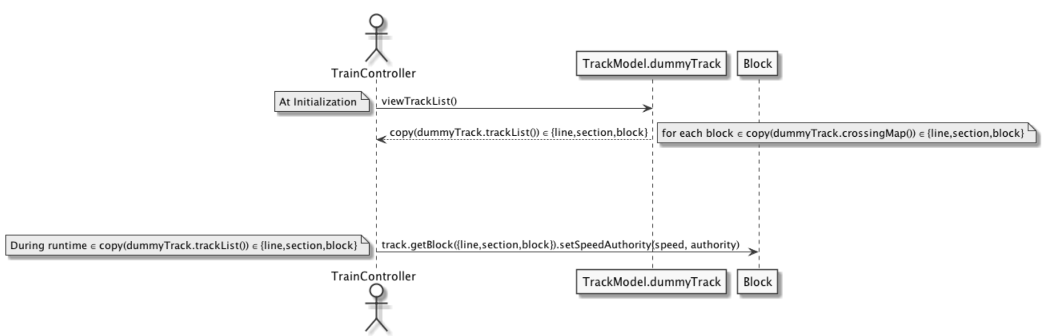


Figure 14: Set Speed and Authority use Case Diagram

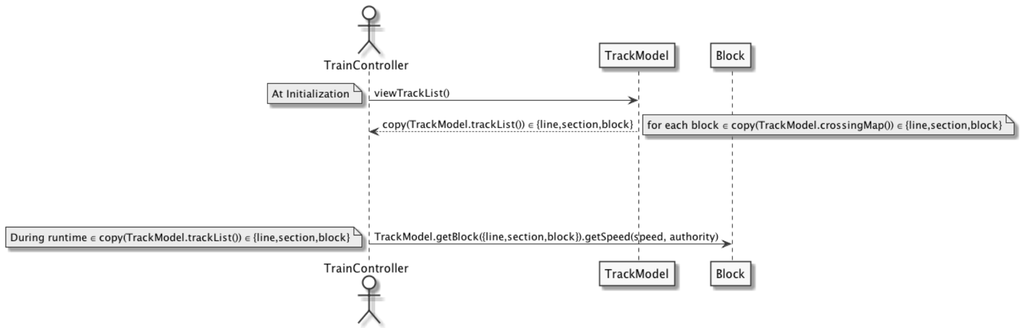


Figure 15: Get Speed Use Case Diagram

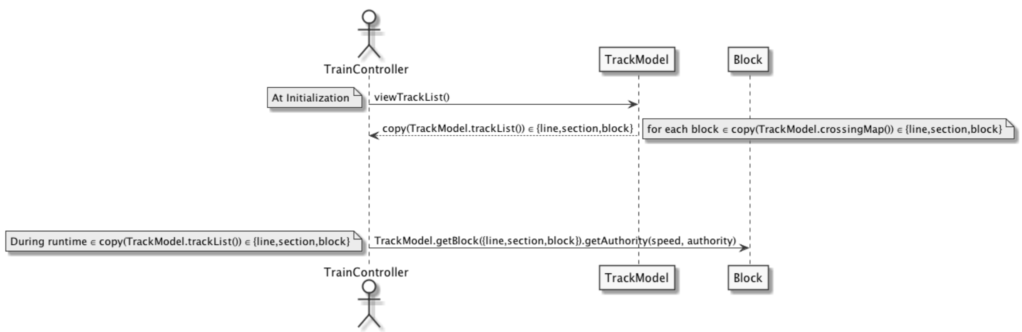


Figure 16: Get Authority Use Case Diagram

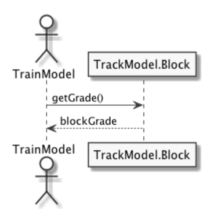


Figure 17: Get Grade Use Case Diagram

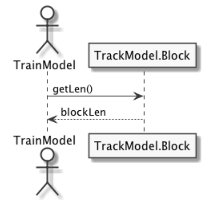


Figure 18: Get Length Use Case Diagram

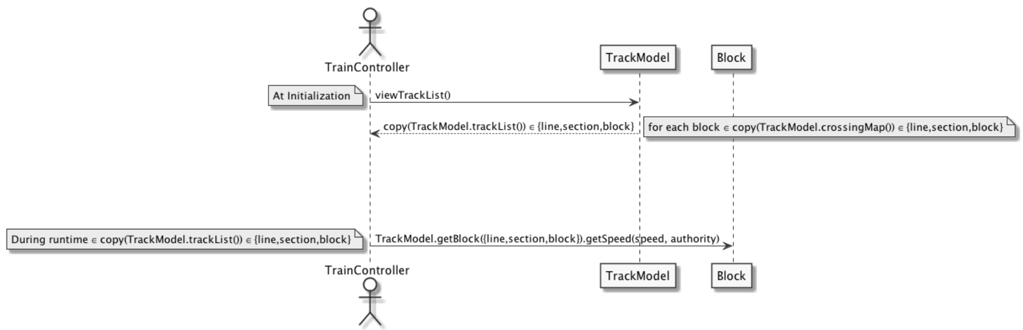


Figure 19: Get Speed Use Case Diagram

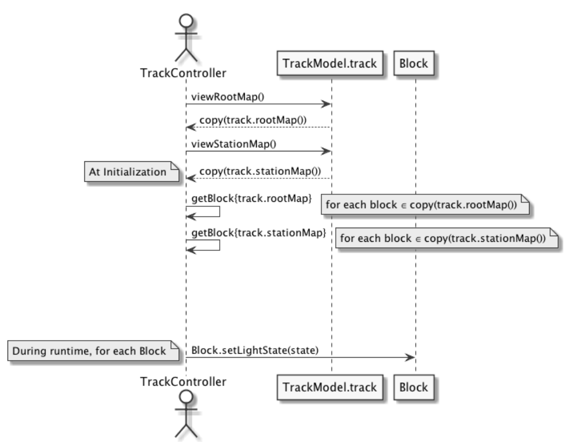


Figure 20: Toggle Lights Use Case Diagram

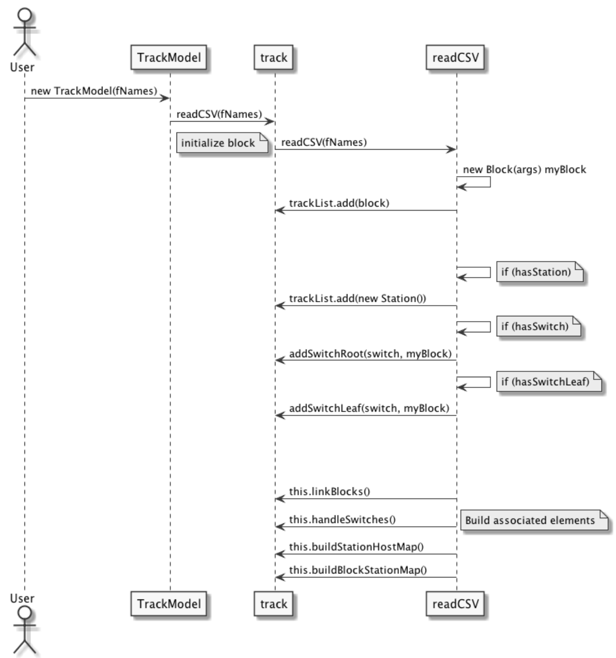


Figure 21: Read File Use Case Diagram

## 4.2 Track Controller

## 4.3 Train Model

In this seciton, we provide the sequence diagrams of the train model.

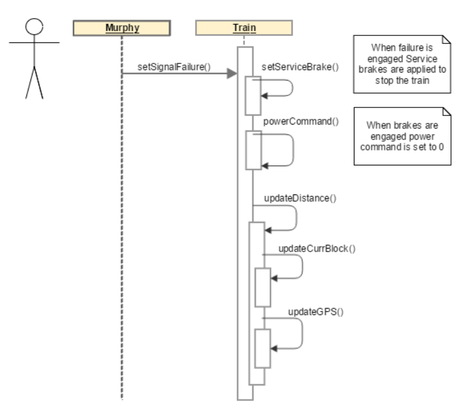


Figure 22: Toggle Signal Failure Use Case Diagram

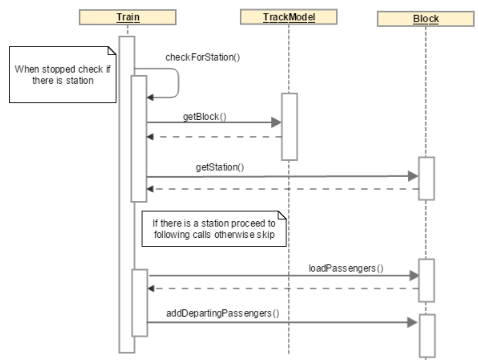


Figure 23: Add Passengers Use Case Diagram

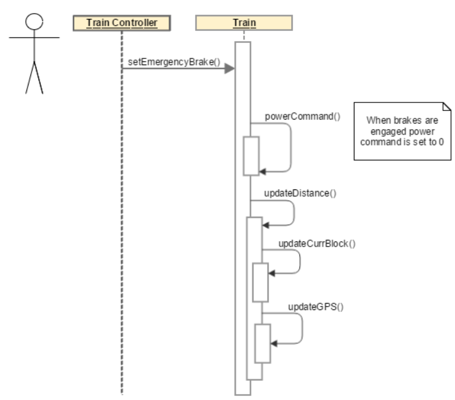


Figure 24: Engage Emergency Brake Use Case Diagram

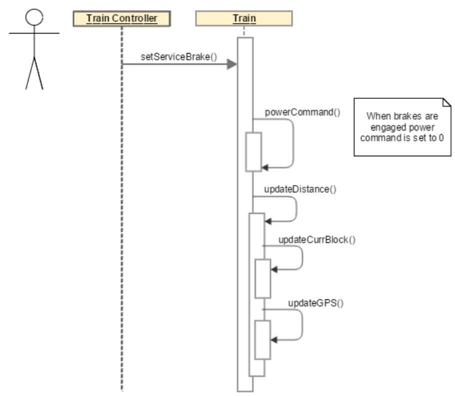


Figure 25: Engage Service Brake Use Case Diagram

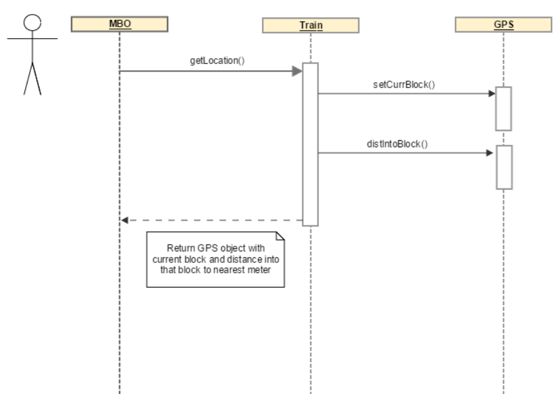


Figure 26: Get Location Use Case Diagram

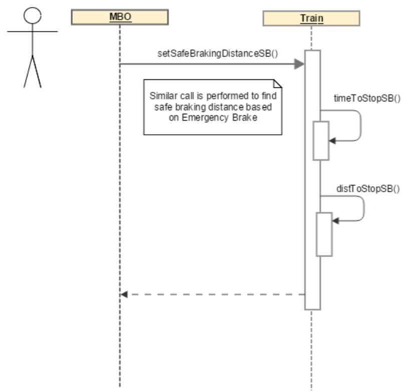


Figure 27: Calculate Safe Braking Distance Use Case Diagram

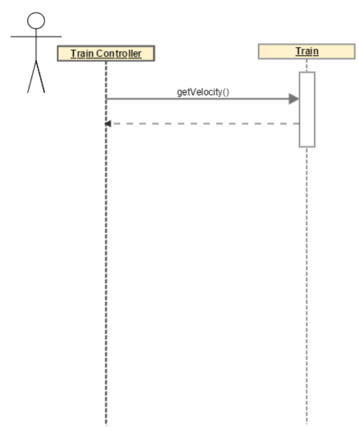


Figure 28: Get Velocity Use Case Diagram

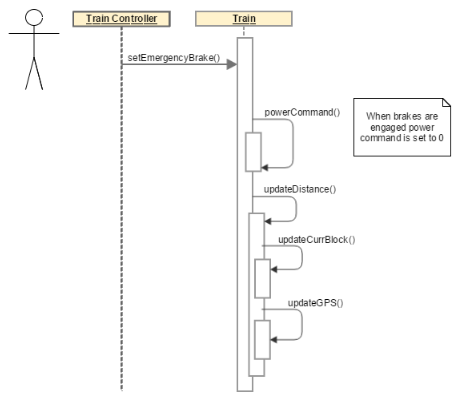


Figure 29: Increase Temperature Use Case Diagram

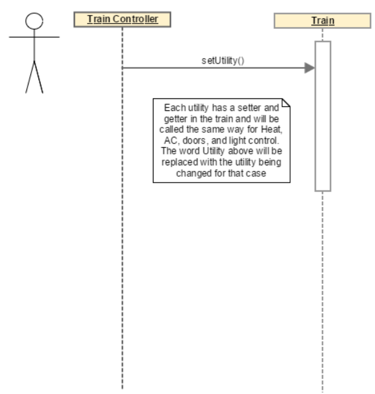


Figure 30: Modify Utilities Use Case Diagram

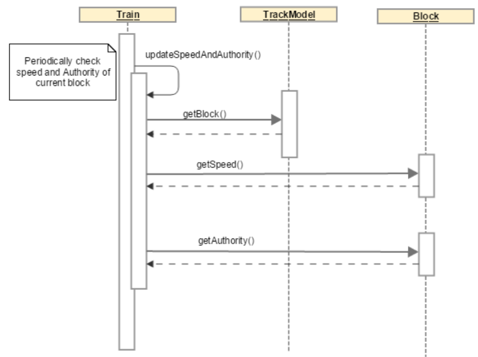


Figure 31: Set Speed and Authority Use Case Diagram

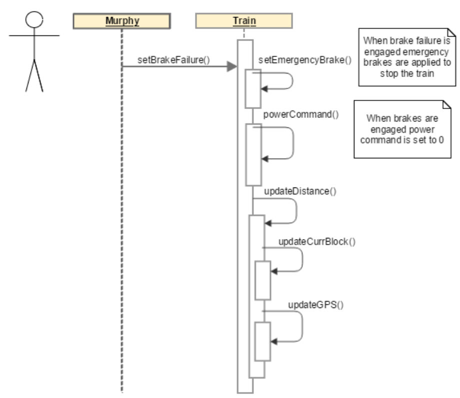


Figure 32: Toggle Brake Failure Use Case Diagram

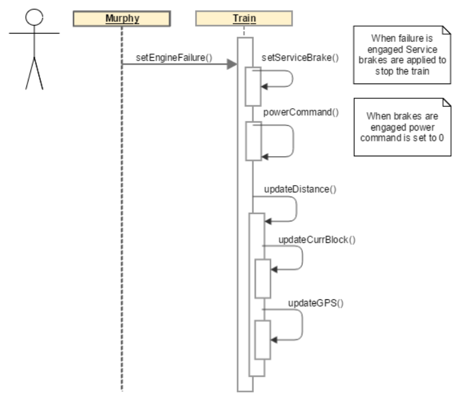


Figure 33: Toggle Engine Failure Use Case Diagram

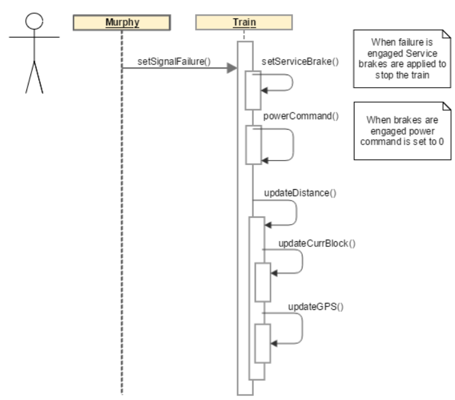


Figure 34: Toggle Signal Failure Use Case Diagram

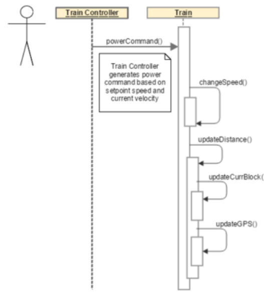


Figure 35: Toggle Signal Failure Use Case Diagram

## 4.4 Train Controller

## 4.5 Moving Block Overlay

## 4.6 Centralized Train Controller