Software Requirements Specification

for

Port Authority North Shore Extension

Version 1.0 approved

Prepared by Sarah Bunke, Nicholas Schnur, Meyling Taing, Keith Payne, Brandon Bock, Catherine Nalesnik

On Track Trainwreck

9/20/13

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Product Scope **Error! Bookmark not defined.**

1.3 Definitions, acronyms, and abbreviations **Error! Bookmark not defined.**

1.4 References **Error! Bookmark not defined.**

1.5 Overview **Error! Bookmark not defined.**

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 General Constraints **Error! Bookmark not defined.**

2.5 Assumptions and Dependencies **Error! Bookmark not defined.**

3. Specific Requirements Error! Bookmark not defined.

3.1 External interface requirements **Error! Bookmark not defined.**

3.2 Functional requirements **Error! Bookmark not defined.**

3.3 Non-functional Requirements **Error! Bookmark not defined.**

4. Appendix Error! Bookmark not defined.

4.1 User Interfaces **Error! Bookmark not defined.**

5. Index Error! Bookmark not defined.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This SRS document describes details and requirements for release 1.0 of the Centralized Traffic Control Center and Signaling System for the North Shore Extension Transit System. The documents primary audience is Port Authority of Allegheny County which is the client. This SRS was created based upon the specifications and requirements set forth by the client. The secondary audience is the members of the On Track Trainwreck team that are developing the system. The developers will use the document as design guidelines to implement and also to verify that the system functions correctly.

## Product Scope

The scope is to develop a fully operational prototype of a Centralized Traffic Control Center and Signaling System for the North Shore Extension Light Rail Transit system will be developed for the Port Authority of Allegheny County. The primary goal of the software is to create a safe and reliable system for transportation. This product will include a train and track model to simulate and control the transit system. It will also include a control center and communications between the different parts of the system.

## Definitions, acronyms, and abbreviations

### Centralized Traffic Control (CTC) – The main office in which the dispatcher monitors and controls the transit system. This interacts with the train and track controllers.

### Moving Block Overlay (MBO) – Controls speed limit and authority for the train in addition to scheduling the transit operators and tracking passenger movement.

### Wayside Controller/Track Controller – Controls signals, switches, detects trains, and detects track faults.

### Authority – The distance the train is allowed to travel.

### Setpoint – The target velocity for the train to travel

## References

### IEEE Standard 830-1993 - Recommended practice for software requirements specification

## Overview

In section 2 the CTC software is detailed. The six modules that comprise the system are described in 2.1. Section 2.2 gives a brief overview of the product funtions and what the overall system will be trying to accomplish. In section 2.3 the seven users of the system are discussed and how they affect the simulation. Constraints and assumptions of the system can be found in sections 2.4 and 2.5.

Section 3 includes all of the functional and non-functional requirements of the system. In section 3.1 the required communications to the separate modules are described. The external interfaces are detailed in section 2.3 for how the modules will interact with users. Section 3.2 lists all of the functional requirements which are listed by module. Since each module has its own set of requirements this section is divided into six parts. The non-functional requirements can be found in section 3.3. In section 3.4 the design constraints are discussed

Section 4 has the appendix which contains user interfaces.

Section 5 is an index for important terms used in the SRS.

# Overall Description

## Product Perspective

The North Shore Extension is an addition to the current transportation system for Port Authority of Allegheny County. The product includes six modules: Train Model, Track Model, Train Controller, Track Controller, Moving Block Overlay (MBO), and the Centralized Traffic Control (CTC). The software is designed to have safety systems in place to ensure that the CTC system runs reliably and transports passengers without harm.

## Product Functions

### To simulate the flow of passengers and trains through the system

### To simulate the scheduling of trains and personnel (or employees)

### Ensuring the safety of the passengers and personnel

### To have an intuitive user interface

### Model the track factors

### Model the train and forces acting upon it.

### Set authority and setpoint speed

## User Classes and Characteristics

### Track Creator

#### Will be able to create the track using a track layout file

### Dispatcher

#### Can close tracks down for maintenance

#### Can route trains

### Conductor

#### Can use the emergency brake

#### Can issue the setpoint speed command

#### Can issue the brake command

#### Can control the train lights

#### Can control the train doors

#### Can change the train temperature

### Passenger

#### Has a weight of 150 pounds

#### The total passenger weight has an effect on the power needed to move the train

#### Can pull the emergency brake

### Scheduler

#### Creates schedule for train operators

#### Accounts for breaks

#### Length of Shift

#### Trains have to return to the yard for shift changes

#### Shifts are 8.5 hours long

#### Breaks shall be 30 minutes and will be provided after 4 hours of driving

### Maintenance

#### When part of the system breaks, maintenance will be alerted to fix it

#### When the fix is accomplished, they change the status to fixed

### Track Engineer

#### Operates the lights, the crossing gates, and the switches manually

#### Loads the PLC program and sets it to auto run mode

## General Constraints

The system will run on a Windows 7 Operating System in a Java Runtime Environment (JRE). The Train Controller, MBO, and Track Controller are vital controllers and are implemented in a safety critical manner.

## Assumptions and Dependencies

We assume the computer has Java 1.7 Runtime Environment as well as supports MySQL. We assume the system will not lose power during the simulation.

# Specific Requirements

## External interface requirements

### The Centralized Traffic Control Office (CTC)

#### The CTC will be the location of the main interface to control the system. It shall monitor the entire transit system.

#### It shall be controlled by a human dispatcher.

#### The user interface shall be a window which includes a display of a map of the system, the status of the individual trains currently on the track, and buttons that allow the user to change different options in the system.

#### This map display shall consist of:

The track layout

##### All of the stations, including the CTC Office

##### Switches and which branch is currently active for each switch

##### Current position of trains currently on the track

##### Any broken parts of the system

#### This display shall have resolution of one meter.

#### The status of each train shall be displayed in a table which includes the following

##### Name of train

##### Location

##### Whether the train is running or stopped

##### Speed and direction

##### Authority

#### From the buttons, the dispatcher shall be able to

##### Schedule trains

##### Route trains

##### Set authority and speed for trains

##### Close track sections for maintenance

##### Add or remove tracks

#### When the user clicks on one of the buttons, a new screen will appear with more specific options to choose from. The user can submit new changes or cancel them to go back to the main screen

### Track Model

#### The Track Model shall only communicate with the Track Controller.

#### For each block the Track Model shall communicate to the Track Controller the speed limit, size, grade, elevation and allowable direction of travel.

#### The Track Model shall communicate to the Track Controller whether a block is a branch, a railway crossing or a station.

#### The Track Model shall communicate to the Track Controller which train is on which block.

#### The Track Model shall communicate to the Track Controller the following failure modes: broken rail, track circuit failure, or power failure.

### Track Controller

#### The track controller shall receive information from the CTC office that includes information about train authority and speed limits.

#### The track controller shall communicate to the CTC office the current state of the track including railway crossings, signals, and trains.

#### The track controller UI shall display the status of all of the individual PLCs. The display will include the following, PLC Identification, rail status, train presence, and intersection signals if activated.

### Train Model

#### The train model simulates the train

#### The train model can be controlled by the conductor, passengers, and maintenance users.

#### The user interface shall include a table that shows all relevant information for the users to control the train.

#### The table will have the following values:

##### Train ID

Current speed

Setpoint speed

Speed Command

Brake command

Lights

Doors

Temperature

Failure

#### The table values will be editable from five different interfaces that are relative to the user classes that will need to access the train model.

#### A dropdown box is used to choose the train that the user would like to update.

#### Slider controls along with confirmation buttons will be used to control the speed and temperature.

#### Buttons will be used to toggle values such as the lights, doors, and failures.

#### The passenger interface has a textbox that announces the next stop on the trains route.

### Train Controller

#### The train controller UI shall display information including:

##### The speed limit

##### The speed set-point

##### Authority

##### Command set-point

##### Position data from the GPS

##### Door status (open/closed)

##### Light status (open/closed)

##### Next station

##### Faults in the train

#### The train controller UI shall allow the user to:

##### Allow the engineer to adjust the speed.

##### Open and close doors (or auto).

##### Turn lights on and off (or auto).

#### The train controller shall receive authority and speed limit input from the Moving Block Overlay via track signals.

### Moving Block Overlay (MBO)

#### The MBO shall receive the following information from the Train Controller:

##### The train’s current position/location

##### Passenger count

#### The MBO shall receive the following information from the Track Controller:

##### Fixed block speed

##### Fixed block authority

##### Broken rail information

##### Track elevation and length

#### The MBO shall transmit the following information to the CTC:

##### The train’s safe Moving block speed set point

##### The train’s safe Moving block authority

#### The MBO shall transmit the following information to the Train Controller:

##### The train’s safe Moving block speed set point

##### The train’s safe Moving block authority

## Functional requirements

### CTC

#### The CTC shall be aware of the status of the entire train system as well as the status of the individual trains.

#### The CTC shall send train route and schedule information from the dispatcher's input to the train controller.

#### The CTC shall receive proper setpoint and authority for individual trains from the Moving Block Overlay, and send that information to the train controller.

#### The CTC shall be able to close track sections for maintenance, and reopen them when they are done with maintenance.

#### The CTC shall be able to add or remove tracks.

### Track Model

#### The Track Model will accept an Excel file as input to create the track and will store each block and its attributes in a database.

#### The Track Model will implement a train detection circuit to alert the Track Controller where the trains are.

#### The train signals and the switch machines will be modeled.

#### A track heater will be modeled.

#### The Track Model will detect broken rails, track circuit failures, and power failures and alert the track controller.

#### The Track Model will model branches, stations and railway crossings.

### Track Controller

#### The track controller shall detect the presence of broken rails

#### The track controller shall control the switching of the track as well as railway crossings. For railway crossings both the lights and cross bar will be controlled. These lights shall be activated one block before the train reaches the crossing and will remain active until the train leaves the block with the intersection.

#### The track controller shall detect the presence of trains on the rails.

#### The track controller must communicate to the train when to turn the lights on.

#### The track controller shall be a programmable unit that runs a PLC program written by the user. The program shall be separate from the implementation of the other track controllers.

### Train Model

#### The train shall have an engine that provides power and acceleration

#### The train movement shall be based on Newton’s laws

#### The train shall have brakes for deceleration and emergencies

#### The train model shall receive input from the MBO regarding the authority.

#### The train model shall accept input from the train controller regarding the setpoint speed command, brake command, speed limit, acceleration limit, deceleration limit, route information, temperature control, door open, door close, transponder input, track circuit input, and light controller for tunnel.

#### The train shall display information about the next stop to the passengers

#### The train shall have failure modes for Engine failure, signal pickup failure, and brake failure

### Train Controller

#### The train controller shall take input from the Engineer on power or speed.

#### The train controller shall take input from the Moving Block Overlay on the speed setpoint.

#### The train controller shall ensure that the train does not exceed the speed limit.

#### The train controller shall ensure that the train does not exceed authority.

#### The train controller shall use the track signal as input and decode the information to determine speed limit and authority.

#### The train controller shall take as input the command setpoint from a Transit Operator.

#### The train controller shall open and close doors at appropriate times.

#### The train controller shall turn lights on and off at appropriate times, as communicated from the track controller.

#### The train controller shall announce stations and stops at the appropriate times.

#### The train controller shall monitor the train for faults and act upon those faults in a safe manner.

#### The train controller shall take input from the GPS on position and transmit this data to the Moving Block Controller.

#### The train controller shall take input from the Engineer on power or speed.

### Moving Block Overlay

#### The MBO shall calculate the safe stopping distance of each train.

##### The MBO shall use this distance to determine the safe authority of each train.

#### The MBO shall display the variance between MBO and Fixed Block speed and authority.

#### The MBO shall control the mode of operation between MBO and Fixed Block.

#### The MBO shall track and display passenger movement.

#### The MBO shall contain a scheduler that creates schedules for train operators accounting for breaks and length of shift.

##### Trains shall return to the yard for shift changes and breaks.

##### Shifts shall be 8.5 hours long.

##### Breaks shall be 30 minutes long.

##### Breaks shall be provided after 4 hours of driving.

##### Conductors must work for at least 7 hours.

## Non-functional Requirements

The simulation will be run using the Java Runtime Environment and will be capable of running in real-time, 10 times normal speed, and 100 times normal speed. The system will execute on a Windows 7 Operating System. The Train Controller, Track Controller, and Moving Block Controller will be safety critical architectures and communication will be vital.

### Performance requirements

#### Track Model

##### The Track Model must communicate the presence of a train on a block.

##### The Track Model must communicate if a rail breaks, if power fails or if the train detection circuit is broken so that the Track Controller can pass the failures to the appropriate module.

##### The Track Model must communicate the speed limits to the Track Controller so that the trains travel at a safe speed.

#### Track Controller

##### The track control shall be a vital piece of software.

##### The track controller must communicate the status of the track in time for the CTC office to take appropriate actions.

#### Train Model

##### When the train receives input to change the speed, the train must accelerate or decelerate efficiently.

#### Train Controller

##### The train controller shall be a vital piece of software.

##### Some elements of the train controller, however, are not vital and may be separated from vital operations as to improve reliability. These non-vital operations may include operating doors, the annunciation system, lights, and route information.

#### Moving Block Overlay

##### The Moving Block controller must have a safety critical architecture.

##### Moving Block communications shall be vital.

## Design constraints

### Track Model

#### The Track Model will accept an Excel file as input to create the track.

#### The Track Model will store the track in a database using MySQL

### Track Controller

#### The track controller shall be written in a safety critical manner.

### Train Model

#### The train model shall have a maximum power output.

#### The train model shall have a maximum breaking capacity.

### Train Controller

#### The train controller shall be a vital piece of software.

#### The train controller input shall be taken via an encoded signal over the track. This signal must be decoded as to provide information such as the speed limit and authority.

#### The timing of the system must be scalable as to accommodate simulation of higher than real-time speeds.

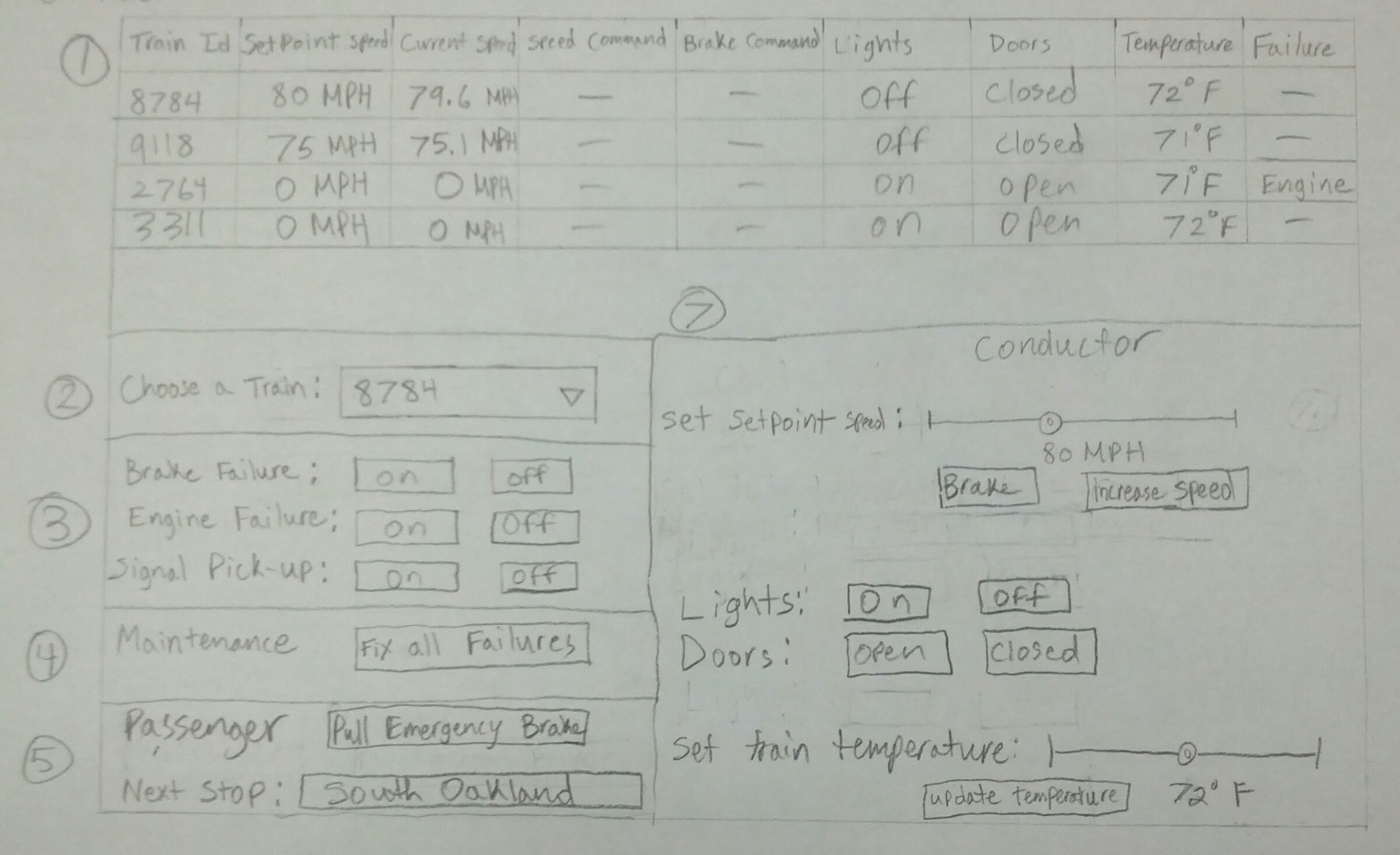
### Moving Block Overlay

#### The Moving Block controller shall be written in a safety critical manner.

# Appendix

## User Interfaces

### Train Model GUI



# Index

Authority, i, 1, 2, 5, 6

brake command, 3, 6

CTC, 1, 2, 4, 6, 7, 8, 11

doors, 3, 6, 10, 12

emergency brake, 3, 6

Java, 3, 4, 11

lights, 3, 7, 9, 10, 12

Location, 4

maintenance, 2, 3, 5, 8

maximum, 12

MBO, 1, 2, 3, 6, 7, 10, 11

Moving Block Controller, 10, 11

North Shore Extension, i, 1, 2

On Track Trainwreck, i, 1

safety, 2, 3, 11, 12, 13

Schedule, 5

Setpoint, 1

setpoint speed command, 3, 6

Shift, 3

simulate, 1, 2

simulation, 4, 11, 13

SRS, 1

system, 1, 2, 3, 4, 8, 11, 12, 13

temperature, 3, 6

track controller, 6, 8, 9, 10, 11, 12

Track Controller, 1, 2, 3, 5, 7, 8, 11, 12

Track Model, 2, 5, 8, 11, 12

train, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

train controller, 13

train model, 6, 12

Trains, 11

vital, 3, 11, 12, 13

Wayside Controller, 1

yard, 3, 11