**

Software Requirements Specification

Ryan Mohan

March 21, 2012

Track Model

Ryan Mohan

|  |  |  |
| --- | --- | --- |
| List Of Revisions | | |
| Date | Name | Description |
| 2/5/2012 | Ryan Mohan | Initial creation of SRS. V 1.0 |
| 3.21/12 | Ryan Mohan | Update for WP2 v 1.1 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Table of Contents

List Of Revisions 2

Table of Contents 3

1 Introduction 4

1.1 Product Overview 4

1.2 Purpose 4

1.3 Scope 4

1.4 Reference 4

1.5 Definitions and Abbreviations 4

2 Overall Description 4

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 User Characteristics 4

2.3.1 Dispatcher/Manager 4

2.4 Assumptions and Dependencies 4

3 Specific Requirements 4

3.1 External Interface Requirements 4

3.1.1 User Interfaces 4

3.1.2 The Track Module shall interface with the CTC office GUI. 4

3.1.3 The track Module shall interface with the Wayside Controller (Track Controller) 4

3.1.4 Software Interfaces 5

3.1.5 Communications Protocols 6

3.1.6 Memory Constraints 6

3.1.7 Operation 6

3.1.8 Product function 6

3.1.9 Assumption and Dependency 6

3.2 Software System Attributes 7

3.2.1 Security 7

3.3 Database Requirements 7

3.3.1 A database to hold Track Configurations 7

3.4 Other Requirements 7

4 Additional Materials 7

4.1.1 Track Layout & Vehicle Data vF.xlsx 7

4.1.2 Track Layout vF.docx 7

4.1.3 Schedules vF.docx 7

# Introduction

## Product Overview

The final module shall present the track in a configurable manor that easily interfaces with the other modules.

## Purpose

The purpose of this document is to define a set of requirements that will dictate all of the functions and features required by the Track module

## Scope

The scope of this document includes an overall description of the Train Module, a list of specific functions and features the system will have.

## Reference

1. IEEE-830 Software Requirements Specification
2. NSC-009 PAAC Bid Package

## Definitions and Abbreviations

**Authority** – how far in distance the train is permitted to travel.

**Block** – a section of a railway line of defined size prior to runtime

**CTC** – Centralized Traffic Control

**Super Green Light** – Two green Lights (max train speed permission)

Region- All the track between two switches

Wayside Controller - controls one switch and the surrounding regions (max of 3)

Signal – The signal light at each block

# Overall Description

## Product Perspective

The train model is a sub module of the final System (see system SRS)

## Product Functions

The product shall properly model the track in any given situation and display the function’s explained below.

## User Characteristics

### Dispatcher/Manager

This user shall be able to view the current track layout, open or close track blocks, configure block sizes, and access any information regarding the track.

## Assumptions and Dependencies

Module is dependent interfacing with the Train and Track Controller.

# Specific Requirements

## External Interface Requirements

### User Interfaces

Sub module- no UI

### The Track Module shall interface with the CTC office GUI.

### The track Module shall interface with the Wayside Controller (Track Controller)

#### The system shall provide the following functions to the following user types

##### Dispatcher

###### The dispatcher shall be able to modify the track.

Open Block

Close Block

The dispatcher shall be able to edit block sizes.

###### the dispatcher shall be able to add or remove Stations/ stops

###### The Dispatcher shall be able to view track conditions, grade and elevation

###### The Dispatcher shall be able to view all track signals and switches.

##### Track Manager

###### The track database shall allow the track manager to add track prior to runtime.

###### The GUI shall enable the user to open and close track blocks.

###### The GUI shall enable the user to close track sections for maintenance.

### Software Interfaces

#### The system shall have a model of the transit system track layout.

##### The track shall be divided into blocks.

###### The block size must be Shown

###### The block size must be configurable

##### Each block shall have an associated grade.

##### Each block shall have an associated elevation.

###### Each block shall have an Elevation associated with each end

Beginning Elevation (negative x or y plane)

Ending Elevation (positive x or y plane)

##### The user shall be able to configure the track model.

###### Prior to Run-Time

Through a database

##### Each block shall have an associated direction of travel.

###### A block may have both directions of travel associated with it.

##### The track model shall support branching via “switches”.

###### Each switch shall connect three pieces of track.

###### Each switch shall connect one “trunk” and two “branches”.

##### Each block shall have an associated speed limit.

##### The track model shall include the following signals designated by the indicated colors:

###### Stop is designated by a red signal

###### Slow is designated by a yellow signal

###### Proceed is designated by a green Signal

###### Full Speed ahead is designated by a Super green signal

Full speed ahead is constricted by the limits set forth by each block of track via the track controller

##### The track model shall detect train presence

###### The trains presence shall be reported to Track Controller

##### The track model shall include railway crossings.

###### Railway Crossings shall be configurable

Railway Crossings are in one of two states:

Open: Meaning non train traffic across is permited

Closed: Meaning gates are down and the train is clear to proceed (if track controller allows)

##### The track model shall include stations for loading and unloading passengers.

###### Prior and After each stop the track shall have a Transponder

The transponder shall communicate what station the train is leaving or entering.

##### Each block shall have the following three Failure Modes:

###### Broken Rail shall be a Failure Mode.

given a broken rail the track module shall report the block as closed to the Track Controller

###### Track Circuit failure shall be a Failure Mode.

The track shall report this to the track controller

###### Power failure shall be a Failure Mode.

Given a power failure the Track model shall report an error state to the Track Controller

##### The track model shall take the following inputs from the track controller:

###### Signal state

Stop

the track model shall change signal color to RED

Slow

the track model shall change signal color to YELLOW

Proceed

the track model shall change signal color to Green

Full Speed ahead

the track model shall change signal color to Super Green

###### Speed limit

The track model shall associate a speed limit with every block

###### Acceleration limit

The track model shall associate an acceleration limit with every block

###### Deceleration limit

The track model shall associate an deceleration limit with every block

###### Output originating from the CTC Office:

Speed

the track model shall change the speed limit for any given block

authority

the track model shall change authority for any given block

Route information system

The track model shall allow viewing of current route information

The track model shall allow for the editing of rout e information via the CTC GUI

##### The track model shall issue the following outputs

###### To the Track Controller

Speed limit

Authority

Block state

Train present

Broken rail

Track circuit failure

Power failure

Railway crossing state

Train approaching

Train present

No train present

signal state

Stop

Slow

Proceed

Full Speed ahead

### Communications Protocols

The train model shall effectively interface with the other system modules.

#### Communication shall be passed from module to module as follows:

##### Track model to CTC office (GUI)

##### Track controller to Track Model

##### Track Model to train controller

### Memory Constraints

No memory constraints are known at this time. The model will run in an efficient manner.

### Operation

#### The Track Module shall effectively and efficiently model any given track, while reporting all outputs to the correct modules

#### The Track Model shall run in a fail safe manor in accoradance with the system as a whole

### Product function

#### to model the track and track signals

### Assumption and Dependency

#### Assumptions

##### All inputs from other modules are correct and accurate

##### All modules properly interface

#### Dependencies

##### The Track Model is dependent on inputs from

###### The CTC GUI

Layout

###### Track Controller

Signal State

Acceleration / deceleration Limits

Speed Limit

## Software System Attributes

### Security

The track model shall only be able to interface with associated modules within the system and assumes the proper security clearances have been met to access the Track Model

## Database Requirements

### A database to hold Track Configurations

#### XML Database

## Other Requirements

The Track Module must be completed in time to be submitted with the System.

# Additional Materials

See The following materials on CourseWeb:

### [Track Layout & Vehicle Data vF.xlsx](https://courseweb.pitt.edu/webapps/blackboard/execute/content/file?cmd=view&content_id=_12491448_1&course_id=_39927_1" \t "_new)

### [Track Layout vF.docx](https://courseweb.pitt.edu/webapps/blackboard/execute/content/file?cmd=view&content_id=_12491450_1&course_id=_39927_1" \t "_new)

### [Schedules vF.docx](https://courseweb.pitt.edu/webapps/blackboard/execute/content/file?cmd=view&content_id=_12491471_1&course_id=_39927_1" \t "_new)

Figure 2 - CTC GUI / Track Layout Sketch