Software Requirements Specification

for

Pittsburgh Train Automation System

Version 0.2

Prepared by Alexander Mcmoil, Cameron Kirsch, Kyle Kessler, Justin Pacella, Yuheng Lin, and Yun Dong

Aurora, Inc.

9/23/2023

Table of Contents

1. Introduction 3
1.1 Purpose 3
1.2 Product Scope 3
1.3 Document Conventions 3
1.4 References 3
1.5 Overview 4
2. Overall Description 4
2.1 Product Perspective 4
2.2 Product Functions 4
2.3 User Classes and Characteristics 5

2.4 Design and Implementation Constraints 7
2.4.1 Intermodular Communications Constraints 7
2.4.2 Hardware Constraints 8
2.4.3 Design Conventions 8
2.5 Assumptions and Dependencies 8
2.5.1 Operating System Compatibility ..... **Error! Bookmark not defined.**
2.5.2 Programming Language ..... **Error! Bookmark not defined.**
2.5.3 Module Interactions 8
2.5.4 Track Controller Hardware Implementation 8
2.5.5 Track and Train Models 9
2.5.6 Documentation and Training 9
2.5.7 Regulatory Compliance ..... **Error! Bookmark not defined.**
2.6 Apportioning of Requirements 9
3 Specific Requirements 9
3.1 Functional Requirements 9
3.1.1 CTC Office 9
3.1.2 Wayside Controller 10
3.1.2.1 Hardware 10
3.1.3 Track Model 11
3.1.4 Train Model 12
3.1.5 Train Controller 13
3.2 Non-functional Requirements ** 14
3.2.1 CTC Office 14
3.2.2 Wayside Controller 14
3.2.3 Track Model 14
3.2.4 Train Model 14
3.2.5 Train Controller 15
3.3 Other Requirements 15

Revision History

Name	Date	Reason For Changes	Version
Aurora Team	9/23/2023	Initial Version	0.1.0
Aurora Team	9/31/2023	Work Package 2 Revision	0.2.0

1. Introduction

1.1 Purpose

This document defines the functional and non-functional requirements that will guide the design, development, implementation, and testing of the Pittsburgh Train Automation System. It is a crucial reference for all stakeholders involved in the project.

1.2 Product Scope

The purpose of the Pittsburgh Train Automation System is to automate train dispatching and control and provide easy-to-use applications for relevant users. The system is split into five separate modules: CTC, Wayside Controller, Track Model, Train Model, and Train Controller. The product will be deployed on the green line and red line of the Pittsburgh Light Rail.

1.3 Document Conventions

1. “Aurora Team” refers to the six technical stakeholders in Aurora: Alexander Mcmoil, Cameron Kirsch, Kyle Kessler, Justin Pacella, Yuheng Lin, and Yun Dong.
2. “The product”, “Our product”, and “The system” refer to the Pittsburgh Train Automation System.
3. The terms “Track Controller” and “Wayside Controller” are equivalent.
4. SRS is an acronym for Software Requirements Specifications
5. “Murphy” is a menace to society who breaks rails, breaks trains, and generally causes faults in the train network.
6. PLC is an acronym for Programmable Logic Controller
7. Authority is a distance value given to each train. It tells each train exactly how far it must travel before stopping. Note that each train must be completely stopped when its authority is zero.

1.4 References

1. Coding Standard
2. Defect tracking Policy
3. Architecture and Interface Dictionary
4. IEEE 830 SRS template: Dalhousie University

1.5 Overview

The Introduction section serves as the gateway to the SRS document for the Pittsburgh Train Automation System project. Its primary purpose is to provide context and clarity to the project's scope and objectives, as well as to the references and conventions of this document. It is like an executive summary in that it can be easily understood by non-technical stakeholders.

The Overall Description section aims to provide more detailed descriptions of the functionality of the system, the assumptions the Aurora team made, and the anticipated user base.

The Specific Requirements section provides detailed lists of all the functional and non-functional requirements for the project along with a list of the constraints of the project.

2. Overall Description

2.1 Product Perspective

The Pittsburgh Train Automation System is intended as a replacement for the current system responsible for managing Pittsburgh's train lines. Our product consists of five modules, each with a specific set of responsibilities:

1. CTC Office
2. Wayside Controller
3. Track Model
4. Train Model
5. Train Controller

The order of these points is deliberate; each module may only directly communicate with its adjacencies with no exceptions. Information transferred between non-adjacent modules must be relayed through the modules that separate them. Figure 2.1 illustrates the allowed communications between the five modules. The directionality of the arrows indicates the allowed flow of information.

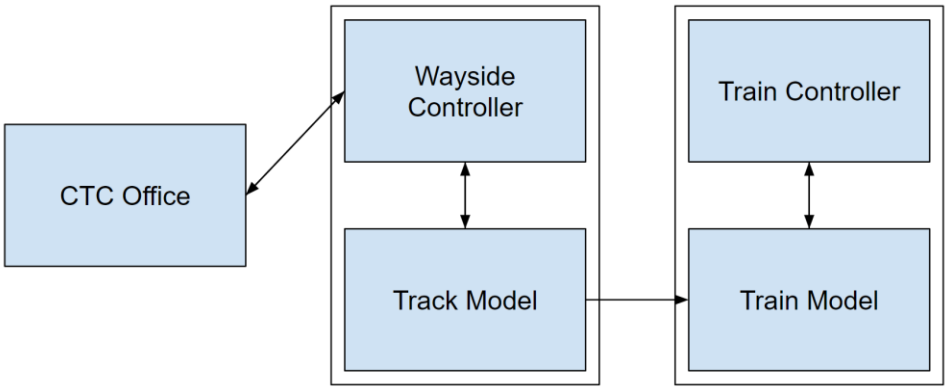

Figure 2.1.1

2.2 Product Functions

The primary function of our product is to fully automate all train system functions, taking safety features, manual mode options, and environmental factors into account. The major responsibilities of each module are as follows:

1. CTC Office

   a. Sends control information to trains
   b. Enables and disables track sections
   c. Load schedule to dispatch trains
   d. Select automatic/manual mode for full system
2. Wayside Controller
   a. Display states of switches, signals, and crossings
   b. Control switches, signals, and crossings
   c. Load PLC programs for track switch logic
   d. Allow manual switch control for manual mode
3. Track Model
   a. Display track map to user
   b. Display environmental variables
   c. Report occupied track sections to the CTC office
   d. Report states of switches, signals, and crossings to the wayside controller
   e. Simulate and report track faults
4. Train Model
   a. Display sensor/actuator states to user
   b. Emergency brake input
   c. Report onboard sensor information to train controller (including train faults)
   d. Control onboard actuators according to train controller
   e. Simulate drive physics
5. Train Controller
   a. Display speed and power information to user
   b. Perform power calculations from proportional gain and integral gain
   c. Select automatic/manual mode for trains
   d. Send control information to train model

The wayside controller, train controller, and train model are all instance modules. Each train dispatched by the CTC office has a model and controller. The number of wayside controllers will be decided by assigning a certain number of track blocks to each controller.

2.3 User Classes and Characteristics

Train Dispatcher

Directs and facilitates the movement of trains to ensure trains and crews move safely and that there are no delays.

Responsibilities

1. Dispatch trains, oversee the system, and issue permits and authorities.
2. Maintain familiarity with scheduled trains, destinations, departure, and arrival times.
3. They may remotely control track switches, traffic signals, and apply or remove blocking devices.
4. Report all accidents, distress signals, hazards, and other emergencies to authorities.
5. Operators are subject to CFR Part 219 Control of Alcohol & Drug Use and CFR Part 228 Covered Service Regulations.
6. They must become and remain General Code of Operating Rules (GCOR) qualified.

Applicant Required Information

1. Minimal education qualifications are a GED or a High School Diploma. Usually train dispatchers will undergo an apprenticeship under experienced train dispatchers.
2. Must maintain and have excellent reading, communication, and computer data entry skills.
3. Valid driver's license

Traffic Control Programmer

The traffic control programmer's role is to control track switches to route trains to their correct destinations. They may do so manually or by writing Programmable Logic Controller (PLC) scripts to manage track switches.

Responsibilities

1. Oversee railroad traffic
2. Control track switches to route trains
3. Control both block and interlocking track signals
4. Control railroad crossings
5. Upload PLC code to automate switch, signal, and crossing control

Applicant Required Information

1. Minimal education qualifications are a GED or a High School Diploma.
2. Traffic Control Programmers must also possess some form of programming education.

Murphy

The term "Murphy" refers to a societal menace or act of God which causes faults throughout the rail system. Individuals may deface, destroy, or cover signs, lights, and other hardware and property, create false blockages on railroads, trip false blockages, or attempt to derail a train by blocking tracks. Fences with barbed wire are installed in high population areas to prevent individuals from interfering with train operations and to deter individuals who may attempt to perform the above acts. Tracks, fences, and hardware may be blocked or destroyed by a tornado, flood, or fallen tree requiring repairs or inspections before operations may resume. The Aurora team does not, nor do any of its affiliates, condone the actions of Murphy.

Passenger

While train passengers are considered users of the Pittsburgh Light Rail, they are not considered employees or affiliates of Aurora. Their demographics vary and they require no formal education.

Responsibilities

1. Pull the emergency brake in an emergency

Train Driver

A train driver operates the passenger or freight train on a rail network. They generally have a high school diploma or equivalent.

Responsibilities

1. Control the speed of the train, enter, and exit platforms safely, brake, and control the train doors when needed. Train drivers shall obey all signaling, safety, and speed instructions.
2. Drivers are expected to work holidays and weekends, and a base roster is used to display workers shifts for the next week, what train they will drive will not always be displayed. Shifts times change by the day as the train schedule is not always the same.
3. Drivers are subject to CFR Part 219 Control of Alcohol & Drug Use and CFR Part 228 Covered Service Regulations.
4. We provide on-the-job training and a certification exam issued by the Federal Railroad Administration (FRA) must be passed before applying to a train route.
5. Train Drivers must recertify once every three years and perform CFR 218 Subpart F Operations and Year A or B training annually.

Applicant Required Information

a. Applicants should have at least a GED.
b. Valid driver's license
c. If an applicant wants to go to a tech school, they must be at least 21 years of age.

Train Engineer

A train engineer operates and controls the power of a train on the railway. To obtain this job, a high school diploma or equivalent is required.

Responsibilities

1. Monitors the speed, air pressure, battery, and other instruments to ensure the train runs smoothly.
2. Communicates with conductors, dispatchers, and other crew members to coordinate train movements and ensure the safety of passengers and crew.

Applicant Required Information

1. Applicants should have at least a GED.
2. Valid driver's license
3. If an applicant wants to go to a tech school, they must be at least 21 years of age

2.4 Design and Implementation Constraints

The successful design and implementation of the Pittsburgh Train Automation System project must navigate various constraints and considerations. These constraints, while posing challenges, are essential factors in shaping the project's scope and approach. All project stakeholders must be aware of these constraints to ensure a realistic and effective project plan.

2.4.1 Intermodular Communications Constraints

As mentioned in section 2.1, the design of the new train automation system shall strictly avoid the creation of a "God Module" or monolithic codebase. A God Module is a single module that controls or interacts with a disproportionately large portion of the system. This poses a significant design challenge for all project stakeholders. In addition to handling its own inputs and outputs, each module must relay information between its neighbors.

The train model (and train controller by extension) is unable to send information to the rest of the system. The implication of this is that it is impossible for the CTC office, wayside controller, and track model to know precise information about the trains. For instance, the track model must use block occupancy detection to approximate each train's location.

2.4.2 Hardware Constraints

The track switch control logic in the wayside controller must be performed by a Programmable Logic Controller (PLC). The PLC must be programmable by the Rail Traffic Controller using PLC code during system operation. Further, the PLC code input by the user must consist solely of Boolean logic.

Communication from the track model to the train model is limited by a beacon capable of transferring 128 bytes of data. The Aurora team must design an equally sized data package to transmit all necessary information to the train model whenever the train model arrives and departs from a station.

The train model module is restricted to a single type of train. The existing infrastructure already uses only one train model. Replacing it would only increase the cost and complicate the design process. The train model used is the Bombardier Flexity 2 train. The train consists of 5 modules and holds up to 222 passengers. The Flexity 2 has a maximum motor power of 120 watts and can reach a maximum speed of 70 kilometers per hour.

2.4.3 Operating Environment

The final deliverable for the Pittsburgh Train Automation System must be runnable on the Windows 10 operating system.

2.4.4 Design Conventions

The Aurora team will adhere to the coding standard agreed upon in Coding Standard.docx. Defects will be tracked using GitHub Issues. See Defect Tracking Policy.docx.

2.5 Assumptions and Dependencies

This section outlines the key assumptions made and dependencies identified in the development and operation of the Pittsburgh Train Automation System. Clear recognition of these factors is essential for the successful planning, implementation, and maintenance of the system. Understanding the underlying assumptions and dependencies helps ensure a robust and reliable system.

2.5.1 Module Interactions

1. **Assumption:** The six modules of the system (CTC control software, track controller software, track controller hardware, track model, train model and train controller software) will interact seamlessly.
2. **Dependency:** Successful integration and communication among the modules are crucial for the system's proper functioning. Dependencies include proper API's, protocols, and interfaces that enable data exchange between these modules.

2.5.2 Track Controller Hardware Implementation

1. **Assumption:** The track controller's hardware implementation, using an ESP-32 microprocessor, will provide redundancy to the software track controller and enhance system reliability.
2. **Dependency:** The hardware implementation must be properly configured and connected to the rest of the system via a Python script that manages serial communication. The reliability and performance of the hardware component are critical for the overall system's robustness.

2.5.3 Track and Train Models

1. **Assumption:** The track model and train model modules will accurately simulate the behavior of the train network.
2. **Dependency:** The accuracy and reliability of the models depend on the quality of the mathematical models and algorithms used in these modules. Any inaccuracies or discrepancies in the model may affect the system's performance.

2.5.4 Documentation and Training

1. **Assumption:** Adequate documentation and training materials will be provided for system users and maintainers.
2. **Dependency:** The availability of comprehensive documentation and effective training resources is essential for the system operators and maintenance personnel. Dependencies include timely creation and updates of user manuals, training materials, and system documentation.

2.6 Apportioning of Requirements

1. As of 9/27/2023, the Aurora team believes all the listed requirements are correct. As the team further develops the project, requirements, functional or otherwise, may be added, removed, or changed.

3 Specific Requirements

3.1 Functional Requirements

1. The System *shall* use SI units for all applicable data unless otherwise specified. This is essential to prevent any unit-conversion-related errors.

3.1.1 CTC Office

1) The CTC Office *shall* supply all necessary information to the system via the wayside controller. This is necessary to ensure the system works as intended. The mentioned necessary information includes but is not limited to:
    a. Suggested Speed
    b. Authority
    c. Train Routing Information
    d. Set Maintenace Mode on Blocks
2) The CTC Office *shall* display all necessary information to the Train Dispatcher. This is needed to inform the user of the system's state.
    a. Train Destination
    b. Train Schedule (to include train arrival/departure times)
    c. Whether the system is in automatic or manual mode
    d. Track Faults (shown as notifications)
    e. Occupancies of track blocks
    f. Passenger Throughput
    g. Disabled Blocks
3) The CTC Office *shall* allow the Train Dispatcher to perform the following:
    a. Upload a train schedule (as a .xlsx or .csv file)
    b. Change the destinations of individual trains
    c. Update the authorities of individual trains
    d. Toggle between automatic and manual mode
4) The CTC Office *shall* receive all the following information from the wayside controller.

   a. Occupancies of track blocks
   b. Enabled/disabled blocks
5) The CTC Office shall receive all the following information from the track model:
   a. Ticket sales
6) The CTC Office *shall* accurately dispatch trains according to the schedule file provided by the dispatcher.
7) The CTC Office *shall* quickly assess and respond to any track fault reports received from the wayside controller. To do so, the CTC Office will disable the relevant track blocks/sections and update the authority values of all potentially affected trains. This is essential to the safety of our crew and passengers.
8) The CTC Office *shall* allow its user to toggle between "Manual Mode" and "Automatic Mode".
   a. In Manual Mode, the CTC Office *shall:*
      i. Allow the Train Dispatcher to dispatch trains along a user-specified route.
      ii. Allow the Train Dispatcher to put a block into and out of maintenance mode, effectively closing and opening the block.
      iii. Allow the Train Dispatcher to set the state of the switch of a block if the block is in maintenance mode and the block has a switch.
   b. In Automatic Mode, the CTC Office *shall* perform all its functions with no user input aside from uploading a train schedule file.

3.1.2 Wayside Controller

1. The Wayside Controller *shall* control all track switches, signal lights, and railroad crossings according to the train routing information provided by the CTC office.
2. The Wayside Controller *shall* immediately notify the CTC office of any track-related faults received from the track model.
3. The Wayside Controller *shall* send the following information to the CTC office:
   a. Block occupancies
   b. Enabled/disabled blocks
4. The Wayside Controller *shall* receive the following information from the CTC office:
   a. Authority for each train
   b. Suggested speed for each train
   c. Blocks to enable/disable
   d. Arrival/departure times
5. The Wayside Controller *shall* relay the following information to each train model via the track model:
   a. Authority
   b. Suggested speed
6. The Wayside Controller *shall* allow the Traffic Control Programmer to toggle between automatic and manual mode. In manual mode, the user may freely toggle track switches, signal lights, and railroad crossings. In automatic mode, the user may upload a Programmable Logic Controller (PLC) program to automate switch control. All other functions will be automated.

3.1.2.1 Hardware

1. The Wayside Controller Hardware *shall* meet all functional requirements listed for its software counterpart.
2. The Wayside Controller Hardware *shall* display all relevant information to the user in real time strictly using hardware components such as
   a. OLEDs

   b. LCDs
   c. LEDs
   d. Rotary Decoders
   e. Pushbuttons
3. The design of the Wayside Controller Hardware *shall* be correctly implemented on a breadboard.

3.1.3 Track Model

1. The Track Model *shall* display a map showing all track-related variables for each block, environmental or otherwise, to the user. This includes but is not limited to:
   a. Block Lengths
   b. Block Grade
   c. Elevation
   d. Track temperature
   e. Statuses of Switches, Signals, and Crossings, if applicable
2. The Track Model *shall* immediately report any of the following faults to the wayside controller:
   a. Broken track faults by setting track block as occupied
   b. Track circuit failure by setting track block as occupied.
   c. Power Failures
3. The Track Model *shall* maintain an actively updated listing of all variables for each track block. This includes but is not limited to:
   a. Heaters enabled/disabled
   b. Track switch states
   c. Track signal states
   d. Railroad crossing states
4. The Track Model *shall* contain the variables below for each track block:
   a. Block length
   b. Block grade
   c. Temperature
   d. Speed limit
   e. Elevation
   f. Cumulative elevation
   g. Traversal time
   h. Infrastructure
   i. Acceleration/deceleration limits
   j. Constant speed time (the time interval in which the train is not accelerating)
   k. Total time to the next station for each train
   l. Dwell time (time sitting at the station between arrival and departure)
   m. Time to station with dwell time

5. The Track Model *shall* update each train's information at each station using beacons. Beacon transmissions are sent to trains upon their arrival at and departure from train stations. Beacon transmissions consist of:
   a. Current station ID if it exists on a station
   b. Length of each block to the next beacon
   c. Grade of each block to next beacon
   d. Speed limit of the track blocks to the next beacon.
   e. Whether that track blocks between this beacon and next beacon are underground

6. The Track Model *shall* on the interactive map inform the user if each block is empty, occupied, or faulty.
7. The Track Model *shall* relay all relevant information from the wayside controller to the train model. This information can be found in Architecture and Interface Dictionary.xlsx.
8. The Track Model *shall* enable track heaters when the track temperature drops below 33 degrees Fahrenheit. This is necessary to prevent ice from increasing trains' stopping distances or interfering with track switches.
9. The Track Model *shall* receive and act upon orders sent by the wayside controller to toggle track switches, signal lights, and railroad crossings.
10. The Track Model *shall* be capable of changing these hardware devices upon signal from the Wayside Controller
    a. Switch states for changing line segments
    b. Lights
    c. Block roads for the train model

3.1.4 Train Model

1. The Train Model *shall* accurately provide all its sensor information to the Train Controller in real time. This includes but is not limited to:
    a. Current speed
    b. Power consumption
    c. Train doors state
    d. Train lights state
    e. Cabin temperature
    f. Emergency brake state
    g. Service brake multiplier
2. The Train Model *shall* relay all data received from the Track Model beacons directly to the train controller to be parsed. Such information includes:
    a. Block Length
    b. Block Grade
    c. Speed Limit
    d. Station ID
    e. Station Side Exit
    f. Underground/Tunnel
3. The Train Model *shall* immediately notify the train controller of any the following train-related faults:
    a. Engine failures
    b. Brake failures
    c. Signal-pickup failures
4. The Train Model *shall* perform the following actions as commanded by the train controller:
    a. Adjust motor speed
    b. Activate AC/heating
    c. Open/close doors
    d. Turn interior/exterior lights on or off
    e. Engage/disengage service brake multiplier

    f. Engage/disengage emergency brake
5. The Train Model *shall* simulate train movement using software. The Aurora team will use Newton's Laws to calculate a differential equation of motion to be simulated. The equation must account for:
    a. Block grade
6. The Train Model *shall* maintain an actively updated list of all train variables such as:
    a. Mass
    b. Height
    c. Length
    d. Width
    e. Crew Count
    f. Passenger Count
7. The Train Model *shall* engage the emergency brake whenever it is triggered by a passenger.
8. The Train Model *shall* only disengage the emergency brake when either the Train Model has come to a complete stop, or the Train Controller declares it is safe to do so.
9. The Train Model *shall* display stop announcements to the passengers.

3.1.5 Train Controller

1. The Train Controller *shall* immediately assess and respond to any faults reported by the Train Model. It may respond by engaging either the service brake or the emergency brake.
2. The Train Controller *shall* display all relevant information to the driver. This includes but is not limited to:
    a. Current speed
    b. Service brake status
    c. Emergency brake status
    d. Door status
    e. Light status
    f. Manual/Automatic mode status
    g. Speed limit
    h. Suggested speed
    i. Commanded Speed
    j. Used Power
3. The Train Controller *shall* receive the following information from the train model
    a. Beacon package
        i. Station ID
        ii. Station Side
        i. Block Lengths
        ii. Block Grades
        iii. Speed Limits
        iv. Underground
    b. Suggested Speed
    c. Current Speed
    d. Faults
    e. Authority
    f. Interior Lighting Status
    g. Exterior Lighting Status
    h. Service Brake Engagement/Disengagement

i. Emergency Brake Engagement/Disengagement
j. Faults with Brake Engagement

4. The Train Controller *shall* receive the following information from the Train Driver
    a. Service brake engagement/disengagement
    b. Emergency brake engagement/disengagement
    c. Door open/close
    d. Lights on/off
    e. Manual mode
    f. Command Speed
5. The Train Controller *shall* receive the following information from the Train Engineer
    a. Proportional gain
    b. Integral gain
6. The Train Controller *shall* send the following commands to the Train Model:
    a. Command Speed
    b. Power limiting
    c. Emergency brake
    d. Service brake
    e. Station stops
    f. Decoded Beacon Package
        i. Block Lengths
        ii. Block Grades

3.2 Non-functional Requirements

1. The system *shall* display all GUI information using imperial units. Since all (most) of our users are from the United States, it will be easier for them to read imperial measurements.
2. Regulatory Compliance
    a. **Assumption:** The system will comply with relevant industry and safety regulations.
    b. **Dependency:** Ensuring compliance with regulations is essential. Any changes in regulatory requirements may necessitate adjustments to the system's design or functionality.

3.2.1 CTC Office

1. The CTC Office *shall* inform the dispatcher of any faults using a pop-up notification or some other means of drawing the user's attention. The functionality of the system does not depend on the CTC office UX displaying pop-ups, but not showing them may impact the response times for faults.

3.2.2 Wayside Controller

1. The Wayside Controller *shall* have safety critical architecture. Since faults in this system can cause injury or loss of human life, this system should put the safety of the users above all other functions.

3.2.3 Track Model

3.2.4 Train Model

1. The Train Model should display advertisements in the GUI.

3.2.5 Train Controller

1. The Train Controller *shall* have safety critical architecture. Since faults in this system can cause injury or loss of human life, this system should put the safety of the users above all other functions.
2. The Train Controller *shall* send station announcement to the train model.

3.3 Other Requirements

1. The Pittsburgh Train Automation System *shall* allow each of its users to adjust one system-wide simulation speed. The user adjusts the simulation speed using a small range of constant multipliers.