

# Software Requirements for the Train Module of the Port Authority of Allegheny County Transit Control System

Nathan Altay Hunter  
Software Engineer  
Bazinga! Industries

|  |  |  |
| --- | --- | --- |
| List Of Revisions | | |
| Date | Name | Description |
| 02/09/2012 | Nathan Altay Hunter | Initial version |

# Table of Contents

[Software Requirements for the Train Module of the Port Authority of Allegheny County Transit Control System 1](#_Toc316494798)

[List Of Revisions 2](#_Toc316494799)

[Table of Contents 2](#_Toc316494800)

[1 Introduction 4](#_Toc316494801)

[1.1 Product Overview 4](#_Toc316494802)

[1.2 Purpose 4](#_Toc316494803)

[1.3 Scope 4](#_Toc316494804)

[1.4 References 4](#_Toc316494805)

[1.4.1 IEEE-830 Software Requirements Specification 4](#_Toc316494806)

[1.4.2 NSC-009 PAAC Bid Package 4](#_Toc316494807)

[1.5 Definitions and Abbreviations 4](#_Toc316494808)

[2 Overall Description 4](#_Toc316494809)

[2.1 Product Perspective 4](#_Toc316494810)

[2.2 User Characteristics 4](#_Toc316494811)

[2.2.1 Train Controller 4](#_Toc316494812)

[2.3 Assumptions and Dependencies 4](#_Toc316494813)

[2.3.1 The train is dependent on the train controller to provide the necessary instructions 4](#_Toc316494814)

[2.3.2 The train is assumed to be a simple point mass which obeys Newton’s laws 4](#_Toc316494815)

[2.3.3 The acceleration limit provided for each train is assumed to be safe for the cargo and passengers it carries 4](#_Toc316494816)

[2.3.4 The mass of an average passenger is assumed to be the mass of each passenger 4](#_Toc316494817)

[2.3.5 The mass of an average car is assumed to be the mass of each car 4](#_Toc316494818)

[3 Specific Requirements 4](#_Toc316494819)

[3.1 External Interface Requirements 4](#_Toc316494820)

[3.1.1 Software Interfaces 4](#_Toc316494821)

[3.1.2 Communications Protocols 4](#_Toc316494822)

[3.1.3 Memory Constraints 4](#_Toc316494823)

[3.1.4 Product function 4](#_Toc316494824)

[3.2 Software Product Features 4](#_Toc316494825)

[3.2.1 Mass 4](#_Toc316494826)

[3.2.2 Speed 5](#_Toc316494827)

[3.2.3 Acceleration 5](#_Toc316494828)

[3.2.4 Length 5](#_Toc316494829)

[3.2.5 Height 5](#_Toc316494830)

[3.2.6 Width 5](#_Toc316494831)

[3.2.7 Doors 5](#_Toc316494832)

[3.2.8 Lights 5](#_Toc316494833)

[3.3 Software System Attributes 5](#_Toc316494834)

[3.3.1 Reliability 5](#_Toc316494835)

[3.3.2 Availability 5](#_Toc316494836)

[3.3.3 Security 5](#_Toc316494837)

[3.3.4 Portability 6](#_Toc316494838)

[3.3.5 Performance 6](#_Toc316494839)

[3.4 Database Requirements 6](#_Toc316494840)

[3.4.1 The train module will not require a database 6](#_Toc316494841)

[3.4.2 All user authentication information and train schedules shall be handled by the CTC 6](#_Toc316494842)

[3.5 Other Requirements 6](#_Toc316494843)

# Introduction

## Product Overview

This component will be used by the Centralized Traffic Control (CTC) system to simulate the behavior of a real train. It will be used by the North Shore Extension of the Port Authority of Allegheny County (PAAC) as part of the complete system.

## Purpose

This document is intended to provide the requirements the train component of the system will adhere to.

## Scope

The scope of this document includes the requirements expected to be fulfilled by the train module of the system.

## References

### IEEE-830 Software Requirements Specification

### NSC-009 PAAC Bid Package

## Definitions and Abbreviations

**CTC** – Centralized Traffic Control

**GUI** – Graphical User Interface

**PAAC** – Port Authority of Allegheny County

# Overall Description

## Product Perspective

The train module simulates a train for use by the system. The train is ultimately controlled by the train controller; it makes no decisions itself.

## User Characteristics

### Train Controller

This user’s role is to tell acceleration, door, light, and temperature to the train.

## Assumptions and Dependencies

### The train is dependent on the train controller to provide the necessary instructions

### The train is assumed to be a simple point mass which obeys Newton’s laws

### The acceleration limit provided for each train is assumed to be safe for the cargo and passengers it carries

### The mass of an average passenger is assumed to be the mass of each passenger

### The mass of an average car is assumed to be the mass of each car

# Specific Requirements

## External Interface Requirements

### Software Interfaces

#### The train will only interface directly with the train controller

### Communications Protocols

#### The train will receive all instructions directly from the train controller module

### Memory Constraints

#### Each instance of the train module shall consume less than 1 megabyte of memory at all times

### Product function

The train module is responsible for modeling the behavior of a train. It is the lowest level component of the system and generally performs the actions given it by the train controller.

## Software Product Features

### Mass

#### The train shall have a mass determined by the sum of the engine, cars, crew, and passengers

### Speed

#### The train shall have a speed at which it travels

#### The train’s speed shall not exceed any of the speed limits it is given

### Acceleration

#### The train shall have an acceleration which changes its speed

#### The train shall not exceed any acceleration or deceleration limits it is given

### Length

#### The train shall have a length equal to the length of a car multiplied by the number of cars it contains

### Height

#### The train shall have a height which must never change after its creation

### Width

#### The train shall have a width which must never change after its creation

### Doors

#### The train shall have two possible states for its doors: open and closed

#### Doors may not be opened or closed independently; they must all have the same state

#### Doors shall be closed whenever the train has a non-zero speed or acceleration

#### Doors shall be open whenever the train is at a station

### Lights

#### The train shall have two possible states for its lights: on and off

#### Lights may not be turned on or off independently; they must all have the same state

#### Lights shall be on whenever the train is in a tunnel

#### Lights shall be off whenever the train is not in a tunnel

## Software System Attributes

### Reliability

#### The train module shall reliably perform its expected functions

#### Any unexpected exceptions should be handled by stopping the train

### Availability

#### The train shall be available to take orders from the train controller at all times

### Security

#### The train shall require speed, acceleration, and deceleration limits to be provided when it is created

#### The train must obey all orders given by the train controller unless they exceed a preset limit

#### In the event that communication with the train controller is lost, the train shall decelerate until its speed reaches zero

#### In the event that the train has a break failure it shall not be permitted to accelerate

### Portability

#### The train module is implemented as a simple C# program and should run on any system which supports C#

#### The train module is only intended to be used as part of the complete system

### Performance

#### Each train shall consume insignificant system resources such that there are no perceivable changes to the system when fewer than 100 trains are in use

## Database Requirements

### The train module will not require a database

### All user authentication information and train schedules shall be handled by the CTC

## Other Requirements

The train module itself will not be required to provide any graphical user interface. It will be controlled directly by the train controller, which will be controlled by the CTC. Only the CTC will provide a graphical user interface.