**

Centralized Traffic Control Graphical User Interface Architecture and Design Description

Version 1.0

March 22, 2012

PAAC Demonstration System

|  |  |  |
| --- | --- | --- |
| List Of Revisions | | |
| Date | Name | Description |
| 3/22/2012 | Jeremy Nelson | Initial design document |

# Table of Contents

List Of Revisions 2

Table of Contents 2

1 Introduction 3

1.1 Product Overview 3

1.2 Purpose 3

1.3 Scope 3

1.4 Reference 3

1.5 Definitions and Abbreviations 3

2 Purpose and Responsibilities 3

3 Architecture 4

4 Design 5

4.1 Main Screen 5

4.2 Track Display 5

4.3 Information Display 5

4.4 Command Display 6

4.5 Table Display Screen 6

4.6 Scheduling Screen 6

4.7 CTC Controller 6

4.8 Schedule Manager 6

4.9 Class Diagram 6

4.10 Example Sequence Diagrams 8

# Introduction

## Product Overview

The final system will be a software prototype of a Centralized Traffic Control (CTC) Center and Signaling System for a light-rail passenger transit system. The software will be used for a demonstration of a proposed North Shore Extension of the Port Authority of Allegheny County (PAAC).

## Purpose

The purpose of this document is to outline the software architecture and design of the CTC Graphical User Interface (GUI) component of the system.

## Scope

The scope of this document is to provide a medium-level description of the software architecture and design of the CTC GUI to be used as a basis for its construction and a reference for maintaining it.

## Reference

1. IEEE-830 Software Requirements Specification
2. NSC-009 PAAC Bid Package
3. Centralized Traffic Control Graphical User Interface Software Requirements Specification

## Definitions and Abbreviations

**Block** – a section of a railway line

**CTC** – Centralized Traffic Control

**GUI** – graphical user interface

# Purpose and Responsibilities

The CTC GUI serves to display the real-time state of the transit system and to allow the user to interact with the system. The state of the system primarily consists of the current signal status of all track blocks, but also includes detailed information about each track block and train in the system. User interactions include setting train speed limits and authorities, suggesting train routes, scheduling trains, and opening or closing track blocks. Additionally, the CTC GUI provides an interface to simulate failure scenarios to demonstrate the system response.

# C:\Users\Jeremy\Documents\SoftwareEngineering\CTC_Use_Cases.jpg

Figure 1: CTC Use Cases

# Architecture

The CTC GUI architecture follows the Model-View-Controller pattern. The CTC view entity represents the screens and controls displayed to the user, while the CTC controller entity serves as an interface between the graphical display and the transit system. The model entity is the transit system itself, which is outside the scope of the CTC GUI.

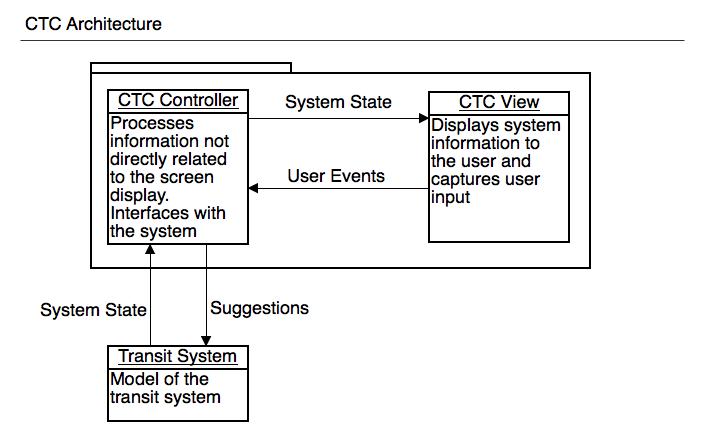


Figure 2: CTC Architecture

# Design

## Main Screen

The primary component of the CTC GUI is the Main Screen. This screen displays the track layout graphically, information about track components and trains, and buttons for user actions. Naturally, the Track Display, Information Display, and Command Display are split into separate entities referred to as User Controls in Visual Studio.

## Track Display

The Track Display shows the track layout in a semi-geographical fashion with colored lines to indicate individual track blocks. Other small graphics are used to represent trains, tunnels, railroad crossings, etc. Each graphic is also a separate user control. The Track Display contains a collection of the track components and sets their locations and sizes according to the layout and sets their state according to the state of the transit system. The individual components are then responsible for painting themselves appropriately. They use a drawing utility class to aid with this. When a component is clicked, it fires an event to the Track Display, which ultimately notifies the Main Screen to take the appropriate action. Generally, the component will blink to indicate it is selected.

## Information Display

The Information Display shows information about the selected component, if one is selected. It is responsible for knowing which information fields of the selected component to display to the user. The information is simply displayed as labels in the format “**Field:** Value” for each field.

## Command Display

The command display shows buttons to take action on the selected component, if one is selected. It is responsible for knowing what actions are available for the selected component. When a button is clicked, it fires an event to the Main Screen to take the appropriate action.

## Table Display Screen

This screen is accessed from a drop-down menu on the Main Screen. It displays information in a table format for all the track blocks or all the trains in the system. The user can switch between viewing track blocks and trains via another drop-down menu. The user can also click on cells to suggest values where appropriate, e.g. a track block authority. When an editable cell is selected, it fires an event to the CTC Controller to take the appropriate action.

## Scheduling Screen

This screen is accessed from a drop-down menu on the Main Screen. Similar to the Table Display Screen, it displays train schedule information in a table format. The user can click on cells to suggest a schedule change. When an editable cell is selected, it fires an event to the Schedule Manager to take the appropriate action.

## CTC Controller

This class is responsible for providing an interface between the GUI screens and the transit system. It is also responsible for passing the system state information to the screens to update the display. It does this by running a timer that expires every 200ms. When the timer expires, the CTC Controller queries the transit system (the track controllers) for the current state of the system. It processes the information and pushes it to the Main Screen and/or Table View Screen, which then pushes it the information to child controls as necessary. The CTC Controller also provides methods for querying system information on demand and passing commands to the transit system.

## Schedule Manager

This class is responsible for maintaining train scheduling information and pushing it to trains and track controllers appropriately. Trains are notified when to depart stations according to the schedule, and track controllers are notified of train routes to adjust switches accordingly.

## Class Diagram

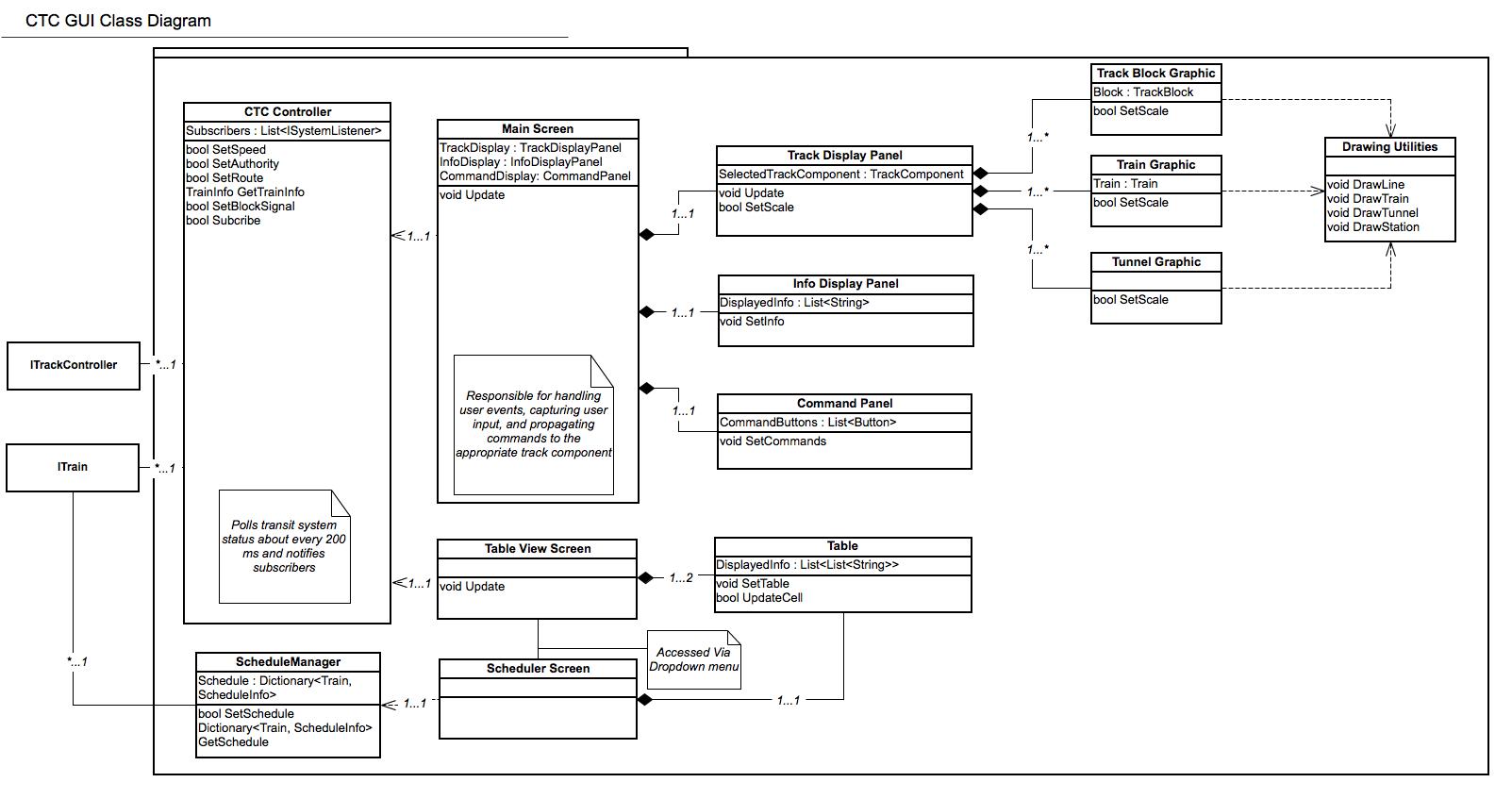


Figure 3: CTC GUI Class Diagram

## Example Sequence Diagrams

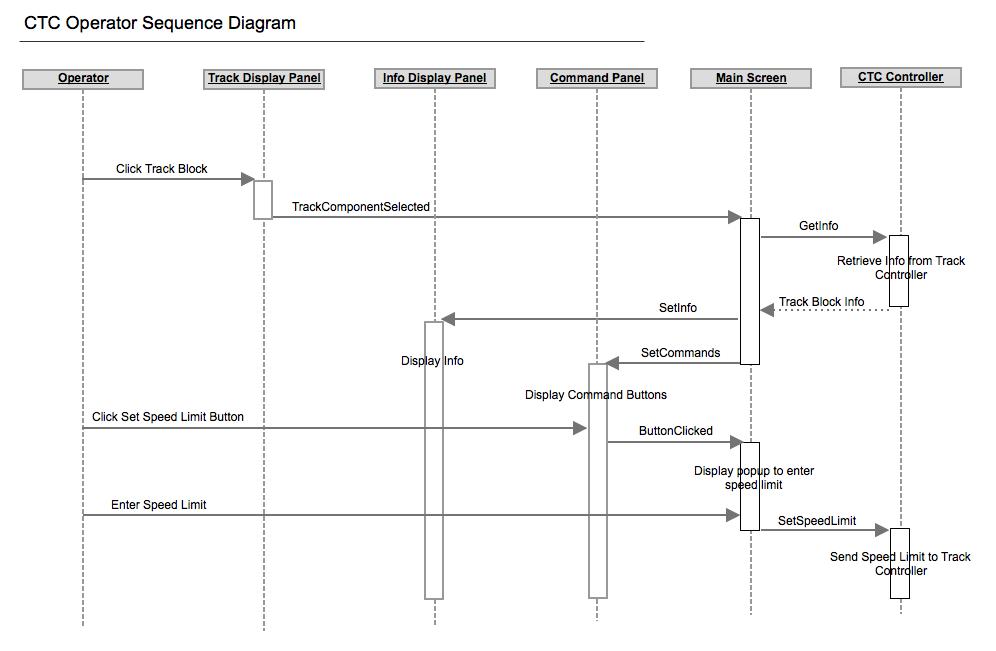


Figure 4: Set Speed Limit Sequence

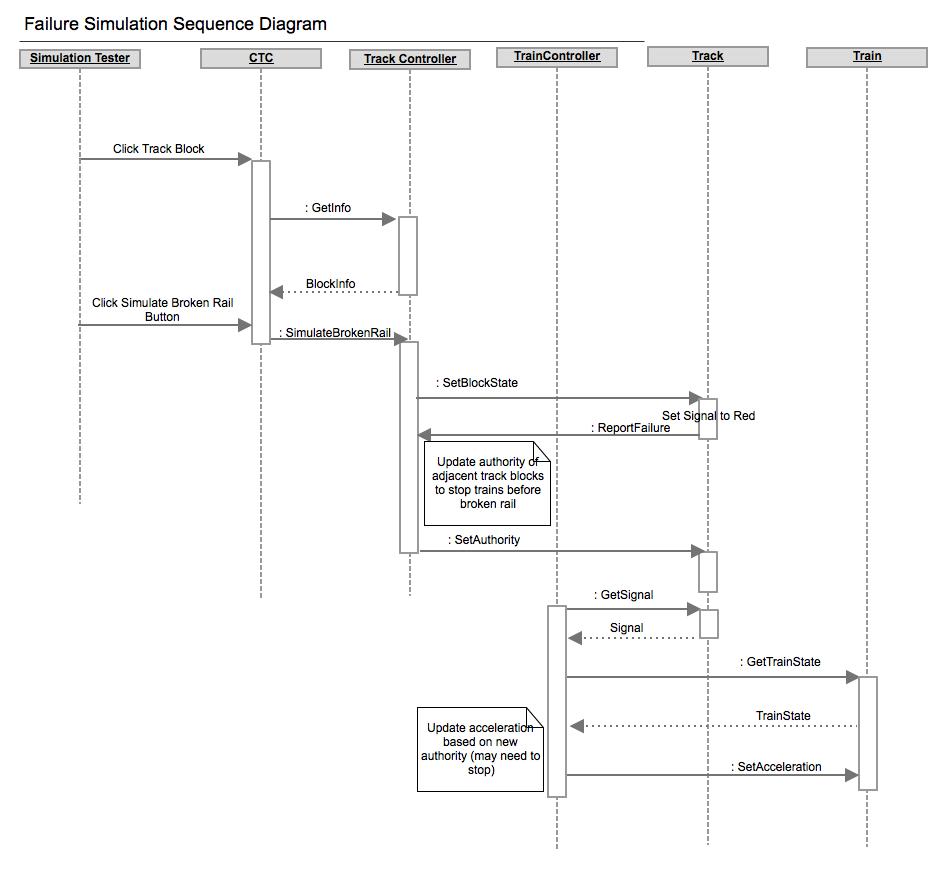


Figure 5: Simulate Broken Rail Sequence