Train Location Project Rough Order of Magnitude (ROM) Estimate

Stephen Jalbert

Rashad Madyun

Corey Sanders

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Description | Author |
| 1.0 | 9/30/2015 | Initial Version | Stephen Jalbert  Rashad Madyun  Corey Sanders |

# Introduction

The computer engineering department owns a model train system is intended to mirror a typical train environment. The purpose of the train track is to be a teaching tool for instructing students on creating safety critical software. In a real-word train environment, trains often carry very valuable assets: people, oil, merchandise, etc. It is important for railway system to be able to track the location of each train in order to prevent collisions and to monitor the state of trains in the event of attack. It is desired for the department model train system to be able to track the location in for each train for this reason. Like subway trains, the department model train system is completely indoors, so a Global Position System (GPS) is not possible.

The train navigation system proposed by our team is composed of a User Interface (UI), a Motion Detection Unit, a Train Navigation Library, and a Train Position Database. The UI displays each train's position on the track, reports any alerts that that are issued by the system, and collects any input needed by the system from the Train operator. The Motion Detection Unit collects information from the train and track necessary to calculate the train's position. This includes collect sensor information about the forces acting on the train. The Train Navigation Library calculates the train’s position from Motion Detection Unit measurements. The Train Position Database stores the history of train position estimates as well as supplementary information to correct for errors in the Train Navigation Library's estimates in train position. A secondary purpose of the database is to use its information to map the geometry of the train railway system.

The figure below shows the relationship between components:

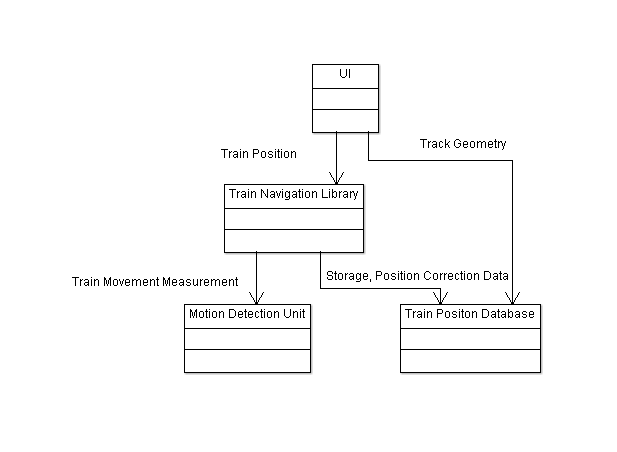


Figure 1 Navigation System Entity STATE DIAGRAM

# Function Point Estimate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | UI | Motion Detection Unit | Train Navigation Library | Position Database | Weighting Factor | Total Per Domain Value |
| Inputs | 2 | 2 | 6 | 4 | 3.43 | 34.3 |
| Outputs | 3 | 5 | 2 | 1 | 4.27 | 42.7 |
| External Inquiries |  | 3 |  | 1 | 3.6 | 10.8 |
| Internal files for storing | 1 |  | 1 |  | 7 | 14 |
| External files for storing |  |  | 1 | 2 | 5 | 5 |
| Totals | **26.67** | **39.01** | **41.12** | **31.59** |  | **106.8** |
| GSC | 39 |  |  |  |  |  |
|  |  |  |  |  |  |  |
| Total Function Points | **111.072** |  |  |  |  |  |

Figure 2 Function Point Summary

# Lines of Code Estimate

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Component | Function Point | Language | Average | Median | Low | High |
| UI | 26.67 | Java | 1413.51 | 1413.51 | 373.38 | 3573.78 |
| Motion Detection Unit | 39.01 | C++ | 1950.5 | 2067.53 | 975.25 | 3120.8 |
| Train Navigation Library | 41.12 | Java | 2179.36 | 2179.36 | 575.68 | 5510.08 |
| Position Database | 31.59 | Java | 1674.27 | 1674.27 | 442.26 | 4233.06 |
| Total |  |  | **7217.64** | **7334.67** | **2366.57** | **16437.72** |

Figure 3 Estimated Lines of Code Projected

The Average (7217.64) and Median (7334.67) lines of code estimates are believed to the most relevant estimates to the size of the project. From the study[1] that was used to derive these line of code estimates, there was a high fluctuation between the lowest estimate of function points per Line of Code and the highest estimate of functions points per Line of Code. Previous experiences of the team have shown that the Average and Median estimates tend to be fairly accurate.

# Function Point Analysis

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | Low | Medium | High |
| Inputs | 3 | 4 | 6 |
| Outputs | 4 | 5 | 7 |
| External Inquiries | 3 | 4 | 6 |
| Internal Logic Files | 7 | 10 | 15 |
| External Logic Files | 5 | 7 | 10 |

Figure 4 Complexity Table for Software Components

|  |  |  |  |
| --- | --- | --- | --- |
|  | Complexity | Weighting Factor | Average per type |
| **Inputs** |  |  |  |
| **UI** |  |  |  |
| Position of the train | Low | 3 | 3.428571429 |
| Time of when train is at specified position | Low | 3 |  |
| Motion Detection Unit |  |  |  |
| Address of the motion detection unit hardware | Low | 3 |  |
| Initial system time | Low | 3 |  |
| **Train Navigation Library** |  |  |  |
| Measured acceleration of the train | High | 6 |  |
| Measured angular velocity of the train | High | 6 |  |
| Time measurements are collected | Low | 3 |  |
| ID of the last detected RFID tag | Low | 3 |  |
| Position of the last detected RFID tag | Low | 3 |  |
| Time of when the last RFID tag was detected | Low | 3 |  |
| **Position Database** |  |  |  |
| Address of the database | Low | 3 |  |
| Position of the train | Low | 3 |  |
| Time of when train is at specified position | Low | 3 |  |
| ID of the last detected RFID tag | Low | 3 |  |
|  |  |  |  |
| **Outputs** |  |  | 4.272727273 |
| **UI** |  |  |  |
| Position of the train | Low | 4 |  |
| map of the track | Medium | 5 |  |
| Time of when train is at specified position | Low | 4 |  |
| **Motion Detection Unit** |  |  |  |
| Measured acceleration of the train | Medium | 5 |  |
| Measured angular velocity of the train | Medium | 5 |  |
| Time measurements are collected | Low | 4 |  |
| ID of the last detected RFID tag | Low | 4 |  |
| Time of when the last RFID tag was detected | Low | 4 |  |
| **Train Navigation Library** |  |  |  |
| Position of the train | Low | 4 |  |
| Time of when train is at specified position | Low | 4 |  |
| **Position Database** |  |  |  |
| RFID tag position | Low | 4 |  |
|  |  |  |  |
| **Internal Logic File** |  |  | 7 |
| **UI** |  |  |  |
| Configuration file to communicate with other components. | Low | 7 |  |
| **Train Navigation Library** |  |  |  |
| Configuration file to communicate with other components. | Low | 7 |  |
|  |  |  |  |
| **External Inquiries** |  |  | 3.75 |
| **Motion Detection Unit** |  |  |  |
| Contact with IMU for acceleration measurements | Medium | 4 |  |
| Contact with IMU for angular velocity measurements | Medium | 4 |  |
| Contact with RFID reader for notification of read RFID tags | Low | 3 |  |
| **Train Navigation Library** |  |  |  |
| **Position Database** |  |  |  |
| Lookup of RFID tag position | Medium | 4 |  |
|  |  |  |  |
| **External Files for storing** |  |  | 5 |
| **Train Navigation Library** |  |  |  |
| Raw sensor measurements | Low | 5 |  |
| **Position Database** |  |  |  |
| Train position | Low | 5 |  |
| Information about the last detected RFID tag from train | Low | 5 |  |

Figure 5 Detailed Break Down of Function Point Estimate

|  |  |  |  |
| --- | --- | --- | --- |
| General System Characteristic | | Brief Description | Value (0-5) |
| GSC 1 | Data communications | How many communication facilities are there to aid in the transfer or exchange of information with the application or system? | 3 |
| GSC 2 | Distributed data processing | How are distributed data and processing functions handled? | 3 |
| GSC 3 | Performance | Was response time or throughput required by the user? | 4 |
| GSC 4 | Heavily used configuration | How heavily used is the current hardware platform where the application will be executed? | 1 |
| GSC 5 | Transaction rate | How frequently are transactions executed daily, weekly, monthly, etc.? | 5 |
| GSC 6 | On-Line data entry | What percentage of the information is entered On-Line? | 0 |
| GSC 7 | End-user efficiency | Was the application designed for end-user efficiency? | 1 |
| GSC 8 | On-Line update | How many ILF’s are updated by On-Line transaction? | 1 |
| GSC 9 | Complex processing | Does the application have extensive logical or mathematical processing? | 4 |
| GSC 10 | Reusability | Was the application developed to meet one or many user’s needs? | 2 |
| GSC 11 | Installation ease | How difficult is conversion and installation? | 2 |
| GSC 12 | Operational ease | How effective and/or automated are start-up, back-up, and recovery procedures? | 5 |
| GSC 13 | Multiple sites | Was the application specifically designed, developed, and supported to be installed at multiple sites for multiple organizations? | 3 |
| GSC 14 | Facilitate change | Was the application specifically designed, developed, and supported to facilitate change? | 5 |
| GSC Total |  |  | 39 |

Figure 6 General System Characteristics Table

# References

1. <http://www.qsm.com/resources/function-point-languages-table>
2. <http://www.softwaremetrics.com/fpafund.htm>