ADDIS ABABA UNIVERSITY INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Developing and Analyzing Automated Tests for Computer Based Interlocking System

The Case of AA LRT

A thesis submitted to:

Addis Ababa Institute of Technology

Addis Ababa University

In Partial Fulfillment of the requirements for the Degree

Master of Science in Electrical Engineering for Railway Systems

By

Bira Kawo

Advisor: Ato Jigsa Tesfaye

June, 2016

ADDIS ABABA UNIVERSITY INSTITUTE OF TECHNOLOGY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Development and Analysis Automated Test Computer Based Interlocking System the Case of AA LRT

By-Bira Kawo

Approval by Board Examiners

Chairman, School of Graduate Studies	Signature	Date
Ato Jigsa Tesfaye Advisor	Signature	Date
Ato Abebe Demissie Co- Advisor	Signature	Date
Internal Examiner	Signature	Date
External Examiner	Signature	Date

DECLARATION

I, BIRA KAWO, hereby declare that this paper entitled “Development and Analysis Automated Test Computer Based Interlocking System” represents my own work and has been written by me in its entirety. This dissertation has not already been submitted for any other degree or to any other university or tertiary institution for examination. To the best of my knowledge and belief, this paper contains no material previously published or written by another person, except where due reference has been made in the text.

Bira Kawo

Name

____

Signature

Date of Submission: ____________________________

This thesis been submitted for examination with my approval as a university advisor

Ato Jigsa Tesfaye

Advisor

____

Signature

Ato Abebe Demissie

Co- Advisor

____

Signature

Abstract

Railway system has safety, economic and environmentally critical system because its failure may cause serious consequences of loss human life, serious injuries, and large scale of environmental damages or considerable economical penalties. Therefore we have to guarantee that they are designed and put into operation properly. Thus, no error can be tolerated in the operation of the system. Interlocking system is used for ensuring the safety of trains. Interlocking systems control all wayside elements in railway traffic. These systems are accountable for safe train operations and must prevent collisions and derailments of the train. Testing the interlocking systems is a key focus. Before start the train operation must be test all equipment according to the interlocking law and safety regulation. In general there are two type of testing approach in railway system that is manual and automatic. Both have on their own advantage and disadvantage. Automation has two major benefits; it saves the verification time and reduces the risk of manual errors. In order to verify whether an interlocking system is consistent with the system requirement specification, it is necessary to carry out a series of testing.

In this thesis to verify the interlocking system use control tables, Boolean function, flowchart, a computer aided tool for the generation of test cases is studied and implemented. A number of test purpose models have been created to direct test case against the interlocking. These test purpose models were implemented in C sharp and Sql server. The entire test setup can be run on any PC and does not require the interlocking hardware. This method can be applies to a number of smaller interlocking systems and partly to the more complex of AALRT depot and station.

Simulations have GUI to interacts the user with the system and enable easy visualization of the movement of the train. The user can inject different possibility of testing mechanism. One testing mechanism was unavailable list of Tracks or Switch interlocking elements. Other methods are using randomly occupied of Tracks and changing of the switch position.

Result show that with 7 routing selected 21 conflicting and 64 derailment of the train are verified and validated. The verification is comparing the expected result with the actual simulations. This simulations tools are used for addition verification that support for AALRT.

Keywords: Computer Based Interlocking system, Control Table, Flow Chart, C Sharp, Sql server

Acknowledgments

I would like to thank my Advisor Ato Jigsa Tesfaye and my Co-Advisor Ato Abebe Demissie support, great interest and guidance in helping me complete my thesis. Thank you to the AAiT and ERC, for the help and use of their resources and for the knowledge that I've gained through experience. Final, thank you to everyone for their support in helping me accomplish this goal.

Table of Contents

Abstract i

Acknowledgments..ii

List of Figure...vi

List of Table vii

Abstractions viii

Chapter One 1

Introduction……………………………………………………………………………………1

1.1. Background..*3*

1.2. Motivation..*5*

1.3 Statement of Problem..*6*

1.4. Objectives..*6*

1.4.1 General Objective...6

1.4.2. Specific Objectives...6

1.5. Methodology...*7*

1.5.1. Literature Review...7

1.5.2. Data Collection..7

1.5.3 Tools Development..7

1.5.4 Simulation...7

1.6 Literature Review..*.7*

1.7 Thesis Limitation..*11*

1.8 Thesis Organization..*11*

Chapter Two..13

Interlocking System 13

2.1. Element of Interlocking System..*.14*

2.2 Principles for Safe Train Movement...*.21*

2.3 Fail Safe System..*.21*

2.4 Safety Standard..*22*

2.5 Modeling of Railway Interlocking...*23*

2.5.1 Formal Method..23

2.5.1.1 Sophistication-Based Formal Methods:..24

2.5.1.2 Semantic-Based Formal Methods:..24

2.5.1.3 Application-Based Formal Methods...24

2.5.1.4 Graphical-Based Formal Methods:....24
2.5.2 Life Cycle Method....25
2.5.3 Design Pattern....25
2.5.4 Software Modeling Technique....26
2.6 Evaluation of the Method....29
2.6.1 Interlocking System....29
2.6.2 Software Engineering Methods....31
2.6.2.1 Formal Method....31
2.6.2.2 Life Cycle Model....31
2.6.2.3 Design Pattern....32
2.7 Selected Methods to Develop the Tools....32
2.8 Railway Routing in Ethiopia....33
Chapter Three....36
3. Modeling of Interlocking....36
3.1 Chapter Overview....36
3.2 Modeling-View Controller Pattern....37
3.3 Agile Lifecycle Methods....38
3.4 Physical Layout....41
3.5 Control Tables....45
3.6 State Charts....45
3.7 Modeling of Interlocking using a Software....50
3.7.1. Introduction....50
3.7.2. Modeling of Interlocking element....51
Chapter Four....57
4. Result and Discussion....57
4.1. Analysis the Result....57
4.2. Test Cases & Results....58
4.2.1. Route One....58
Case 1: Route Request....58
Case 2: Set route....58
Case 3: During the train move testing conflict and non conflict route....
Case 4: Track Occupied Injection....60
Case 5: Track elements Unavailable....60

Case 6: Change the Switch Position........60
Case 7: Switch elements Unavailable........60
Case 8: Resetting of switch and track elements........61
4.3. Discussion........61
4.3.1. Flexibility........61
4.3.2. The Verification needs Signaling Knowledge........62
Chapter Five........63
Conclusion and Recommendation........63
5.1. Conclusion........63
5.2 Recommendation........64
Reference........65
Appendix A Definition........68
Appendix B Some Software Code........69

List of Figure

FIGURE1. 1 INTERLOCKING LAYER 8
FIGURE 2. 1 LED SIGNAL AS THE RUNNING TOKEN IN THE SWITCH AREA [3] 15
FIGURE 2. 2. LED SIGNAL INDICATES NO PASSING [3] 15
FIGURE 2. 3. INDICATES CLEAR ROUTE IN THE FRONT [3] 15
FIGURE 2. 4. INDICATES CLEAR ROUTE IN THE FRONT [3] 15
FIGURE 2. 5 BLOCK DIAGRAM OF AXLE COUNTING SYSTEM [3] 18
FIGURE 2. 6. WHEEL SENSOR [3] 19
FIGURE 2. 7. FUNCTION DIAGRAM OF ACS2000 AXLE COUNTING SYSTEM [3] 19
FIGURE 2. 8. SWITCH CONTROL TO LEFT AND TO RIGHT BRANCH [19] 20
FIGURE 2. 9. MODEL-VIEW-CONTROLLER ARCHITECTURAL PATTERN 26
FIGURE 3. 1 MODELING VIEW CONTROLLER PATTERN 38
FIGURE 3. 2. AGILE LIFE CYCLE 40
FIGURE 3. 3 KALITY DEPOT SCHEMATIC SCHEME [12] 42
FIGURE 3. 4 KALITY DEPOT SCHEMATIC SCHEME [12] 42
FIGURE 3. 5 PHYSICAL LAYOUT USING C SHARP 43
FIGURE 3. 6. ROUTE REQUEST SEQUENCE 46
FIGURE 3. 7 ROUTE AVAILABILITY 49
FIGURE 3. 8. THE TRACK FREE 51
FIGURE 3. 11 GREEN SIGNAL INDICATION 51
FIGURE 3. 12 YELLOW SIGNAL INDICATION 52
FIGURE 3. 13 RED SIGNAL INDICATION 52
FIGURE 3. 9. THE SAFETY CRITICAL OCCUPIED 51
FIGURE 3. 10 OCCUPIED OF THE SECTION 51
FIGURE 3. 14 AXEL COUNTER THAT WHEN THE SECTION FREE 52
FIGURE 3. 15 AXEL COUNTERS THAT WHEN THE SECTION OCCUPIED 52
FIGURE 3. 16 LOGIN VIEW WINDOW 53
FIGURE 3. 17 MAIN SIMULATION WINDOW 54
FIGURE 3. 18 EMPLOYEE REGISTRATION INTERFACE 55
FIGURE 3. 19 TRACKS ELEMENTS INTERFACE 56
FIGURE 4. 1 ROUTE ONE REQUEST 58
FIGURE 4. 2 ROUTE ONE TEST CASE 59

List of Table

TABLE 1. 1 HUMAN LOSES AND INJURIES DUE TO TRAIN ACCIDENTS AND DERAILMENTS FROM YEAR 2005-2009 5
TABLE 3. 1 ITERATION ONE 40
TABLE 3. 2 ITERATION TWO 41
TABLE 3. 3 ITERATION THREE 41
TABLE 3. 4 ITERATION FOUR 41
TABLE 3. 5 CONTROL TABLE OF TRACK 43
TABLE 3. 6 CONTROL TABLE OF SWITCH 44
TABLE 3. 7 CONTROL TABLE OF SIGNAL 44
TABLE 3. 8 BOOLEAN EQUATION 44
TABLE 4. 1 ROUTE ONE TEST CASE PARAMETERS 59
TABLE 4. 2 ROUTE ONE CONFLICT 60
TABLE 4. 3 TESTING CASES RESULT IN THE ROUTES 61

Abbreviations

AC	Axel Conter
ACl2	Advisory Circular 12
ASM	Abstract State Machine
ATP	Automatique Train Protection
ATS	Automatique Train Supervision
CBI	Computer Based Interlocking
CDE	Chemin de Fer Djibouti-Ethiopien, French versionforDjibouti-Ethiopia old Railway line
CENELEC	Comité européen de normalisation electrotechnique
CSP	Communication Sequential Processes
CTL	Computation Tree Logic
EMU	Electrical Multiple Unit
ERC	Ethiopian Railway Corporation
GDP	Gross Domestic Product
GUI	Graphic User Interface
HMI	Human Machine Interface
HSR	High Speed Rail/Railway
IEC	International Electro-Technical Commission
LRT	Light Tail Transit
MPM	Multiprocessor Module
MVC	Model View Controller
NRNE	National Railway Network of Ethiopia
PC	Personal Computer
PVS	Prototype Verification System
RCS	Rail Control Solution
NGO	Non-Government Organizations
RAMS	Reliability, Availability, Maintainability, Safety
RSP	Railway Speed-up Program/Plan
SIL	Safety Integrity Level
SSA	Sub-Sahara Africa
VDM	Vienna Diagram Method

Chapter One

Introduction

Railway is a very important means of transportation for passengers and freight, due to its typical characteristics. Few other transportation modes join dedicated infrastructure connecting point to point cities and places of interests, high operational speed, high reliability, cost effective operation, high energy efficiency and very high safety rate [1]. In Ethiopia, the need for moving passengers and merchandise increases quickly. A key factor in order to gain higher share in passenger transportation demand is needs a substantial improvement in the attractiveness of railway services. Key areas of improvement include providing a better service, decreasing or keeping low fares; increasing reliability and connectivity for both railways traffic and inter modal connections and safety. The increasing movement of people and products at the local, regional, national, and international levels has placed extreme demands on transportation systems, especially in the developed world [1]. This increasing and wide spread transportation demand quests for well-developed transport management system, operation principles, maintenance procedures and safe operations.

Intending to fulfill this enormous and growing demand there are four basic modes of transportation assisting the motorization process, Road, Water, Rail and Air. Of the four modes history tells us that interested question and wide-ranging studies have revolutionized the rail sector from its early beginning in the primitive means of hauling goods at a man step walk to maglev trains flying at speed of more than 500km/hr. This is really one of the most amazing evolutions justifying human’s history being made different by mobility. Nowadays, there are hundreds of thousands of kilometers of railway lines all over the world in operation being powered by Diesel and Electric motive power.

Even if railway transport has many advantages compare to other mode of transportation system, we must make sure that the system has safe operation. To ensure safe operations of train movement automatic train management system involves three main components:

- Automatic train supervision (ATS)
- Automatic train control system (ATC)
- Last but not the least that this paper looking Interlocking system (IXL).

The Automatic train supervision (ATS) involves monitoring the movement of individual trains in relation to schedule and route assignments. Train supervision includes certain information processing and recording activities not directly concerned with train safety and movement but necessary to the general scheme of operations. Train supervision functions are Schedule design and implementation preparing a plan of service in light of expected demand, available equipment, and environmental conditions and insuring a schedule to implement the plan, performance modification that is adjusting movement commands and revising the schedule in response to train, traffic and environmental condition. In general, ATS system used ensuring good traffic flow. Therefore ATS system is not safety critical because responsibility for safety remains with the interlocking system (IXL) and automatic train control system (ATC) systems.

The ATC system reduces the involvement of human in the operation of the trains. The main function improving safety, train services, reducing staff costs. It involves the following functions.

- The automatic train operation subsystem (ATO)
- The automatic train protection subsystem (ATP).

The ATO automatic train operation subsystem assists, and sometimes replace train drivers to operate trains optimally according to timetables, energy consumption, and precise stop of train in stations and so on. ATO system is not safety critical because local responsibility for safety remains with ATP subsystem. The ATP subsystem protects the movement of trains, the opening and closing of train and platform doors, the power supply in case of evacuation and so on. ATP ensures the safe driving of trains at all times. The implementation is trains carry a failsafe computer ensuring that the train does not exceed a safe speed. The computer needs to know a variety of data to calculate the safe speed includes: current speed/ distance, train length, train braking performance, maximum train speed, route data (gradient, maximum line speed). The general operation of ATP is as follows:

- Calculate the maximum permitted speed
- Display important information to the driver
- Monitor the actual train speed along the route
- If the speed exceed by a certain tolerance, give a warning sound to the driver
- If the driver reacts, e.g. by reducing the speed, the warning will cease

- If he fails to do so, the brake is applied automatically. The driver cannot release the brake until the speed reduces to the permitted level.

Therefore ATP subsystem is safety critical.
The interlocking system (IXL) controls the trackside devices (switch, signals, axel counter, and various key locks) according to the orders of the railway operator and the movement of trains. Typically, when the railway operator requests IXL to set and lock a route, the IXL, first checks that this route is compatible with other routes in operation (no risk of collision); second, commands switches along the route to the adequate position; third, lights the aspect of the signal allowing trains to enter the route when all the switch along the route are locked in the adequate position. When a train moves along a route, the IXL system releases the locked devices behind the train and lights the aspect to the signal at the entrance of the other route forbidding other trains to enter that route. Therefore IXL system is safety critical [2].

1.1. Background

Ethiopia, the seat of African Union, has a more than 100 year old meter gauge 781 km diesel railway owned jointly with the government of Djibouti and operated by CDE. This railway, which was established during the reign of Emperor Menelik II, in addition to its deterioration and malfunctioning due to age, almost abandoned due to its incapability of supporting the current demand of freight and passenger mobility [3]. In its times, it has served as a major means of passenger and freight transport to the Eastern part of the country and contributed to the Establishment and expansion of major economically active urban center along its line like Debrezeit, Nazret, Diredawa,Mieso, etc. The railway has a significant place in the cultural songs, proverbs, poetry, etc of the country. Especially for the city diredawa, railway is the soul that exists is the flesh of the villages, the streets; the squares which is revealed by their names. Therefore, the people and the land of Ethiopia are not new to the service of railway. However, due to the current growth of commodity of the country, they need a modern and reliable railway system that can accommodate and facilitate the growth.

In Ethiopian's new Era, following the establishment of ERC with sole aim to develop railway infrastructure and provide passenger and freight rail transportation services, construction of several railway lines is underway all over the country, being classified in to two phases and eight corridors, including the light train project in Addis Ababa [4]. Replacing the old Diesel Traction, the total railway line planned to be constructed by 2025 is EMU traction with more than 5000km costing about US$20 billion dollar [5]. Moreover, the LRT now in operation and the National Railway Network (NRN) 80% of its work accomplished.

In this railway project there are many track sharing area, switch and station layout. Therefore to get safe and reliable operation we must implement interlocking system. An interlocking system controls and monitors signaling elements in a railway layout. The interlocking establishes the safety of the rail system. As technology has progressed so has the type of interlocking systems and technologies as well as improved. There are four fundamental types of interlocking systems: mechanical, electro mechanical, relay and electronic system and electronic interlocking system.

In our railway project they are many stations, shared tracks and switch, therefore there must be applied interlocking system to get safe operation. In the LRT they use the technology of Computer Based Interlocking
System (CBI). CBI shall ensure the safety train running, and control the route, signals and switches under predetermined interlocking conditions and time sequences to make sure that the interlocking among the signal elements in the route such as track sections, switches and signals is safe. For the faulty operation of the device, there shall be valid protective capability.CBI is combined with the ATS system to control the train automatically. Through ATS network, the interlocking devices provide the signal status information to ATS equipments, and

receive the route control command from ATS system. CBI combined with ATS system shall achieve the local control and center control. CBI performs the interlocking logic processing of depot and the digital interface with mainline. Be responsible for collecting and driving the relevant wayside signal equipment. CBI interfaces with wayside signal equipments, switch machine, axle via Chinese vital relays, interface with depot ATS, and send the indication information to ATS. And provide the friendly maintenance and diagnose function [5].

1.2. Motivation

Railway transportation modes are highly safety-critical system, because small error could result in death or serious injuries of human being. For example the Table 1.1 shows the number of human loses and injuries due to trains accidents and derailments from year 2005 to 2009 [6].

Table 1. 1 Human Loses and injuries due to train accidents and derailments from year 2005-2009

Accident and Derailment	2005		2007		2009	
	killed	Injured	killed	Injured	killed Injured	
Level Crossing Accidents	122	370	85	130	35	26
Crossing Accidents	40	70	30	120	60	330
Derailments	230	1100	245	340	120	250
Linear Track Accidents	180	200	74	125	40	100
Other	110	237	70	200	35	120

Therefore Safety has an important issue in railway operation system. Most accident happened around in the interlocking system. Thus Interlocking system has an important safety responsibility in the railway infrastructure. Moreover it is absolutely essential that the systems will not fail in an unsafe manner. As a result, a lot of effort is put in the verification and validation of interlocking systems. An important part of the overall verification effort is whether the interlocking systems are safe meaning neither collisions nor derailing can happen.

Other issues are speed of verification and accurate result. To verify the interlocking equipment with manual it takes more time than computer based. In manual verification the human error has a big problem. With computer based verification we can reduce this error. Beside above, there is a benefit of cost minimization when we used computers to the verification, because we can reduce the verification time and manpower.

1.3 Statement of Problem

Railway interlocking is a system which prevents trains from collisions and derailment. Trains are always at risk for collision because, running at fixed track, train is not capable of steering away from another conflicting train as like road vehicle to avoid collision. Moreover, the train cannot decelerate rapidly, when the driver see track is already occupied. The responsibility of controlling of collision and derailment is for interlocking system. Due to the complexity of interlocking systems, it might be difficult and time consuming to manually prove the safety of trains at a given depot and station. Therefore, it has been decided to simulate interlocking systems in such a way that safety properties can be checked by an automated verification tools. To enable an automated verification of safety properties, one needs to change the complete behavior of an interlocking system and formulate safety properties for it. This thesis will only consider the following basic safety goals:

- ❖ Trains should not collide (If two route has conflict don't permit to move the same time).

Trains should not derail (once the trains permit to move in the rout, we have to make sure that the switch must be in the right position until the train reach at end of the route)

1.4. Objectives

1.4.1 General Objective

The overall objective of this research is to insure safe operation of train movement in the interlock system and how the interlocking system can be verified in relation to the safety properties.

1.4.2. Specific Objectives

The specific objectives of this thesis are to:

- ❖ Analyze of the physical layout of interlocking system for depot and also the regulation of train routing system and train routing table.
- ❖ Change the analysis result to state charts and change the routing table to Boolean formula
- ❖ Build up tools that automatically simulate in different cases that can be directly verification done
- ❖ Develop Graphic User Interface
- ❖ Develop for user to Add, Update and Delete the interlocking elements easily.
- ❖ Verifying each routing system using simulation

1.5. Methodology

1.5.1. Literature Review

In this study different related research works have been revised. This contain browsing internet, reading book, publications and journals related to interlocking system safety and related works. The thesis began with a pre-study focusing on gaining the data formats of the equipment and documents as well as standards and requirements of the routs. A literature study was carried out the related research that done previously and their contribution related to safety of railway operations under interlocking system.

1.5.2. Data Collection

Collecting the data on the interlocking system that is signaling, switch, and axel counter then the collecting data inputting into the physical layout, control table flow chart and simulation establish the objective of the work. The standard of safety was collecting in the interlocking system and the regulation of the routing system

1.5.3 Tools Development

The knowledge acquired from the pre study and literature study was put into practical use to develop maintainable and flexible tools that verifies the interlocking system in different cases. The development of the tool was the most time consuming part of the thesis.

1.5.4 Simulation

Simulation was done using the development tools according to the railway safety regulation and safety standard of the international railway system

1.6 Literature Review

There are many researches done in the verification, modeling and simulation of interlocking system. In this section we will see some of the works.

Hansen [7] presented a VDM (Vienna Development Method) model of a railway interlocking system, and validated it through simulation using ML. The work focuses on the principles and concepts of Danish systems rather than a particular interlocking system. He also pointed out that Interlocking systems from other countries may be different from the Interlocking described in [7]. The simulation he used are ML programming language it is very difficult to program. Due to the programming that uses machine language thus it has not easily flexible.

Fokkink and Hollingshea [8] provided a perspective that can classify the research work regarding verification of railway signaling systems. According to the railway signaling system is divided into three layers: infrastructure, interlocking and logistic layers. All layers must provide safety for railway operation. The infrastructure layer involves objects or equipment used in the yard. The work in this category is ties closely with manufacturer's products. The logistic layer involves human operation and train scheduling which aims at efficiency and deadlock free. It involves the operation of whole railway network thus the state explosion problem is often encountered. The interlocking layer provides the interface between the logistic and infrastructure layers. It prevents us from accidents caused by human errors or equipment failures.

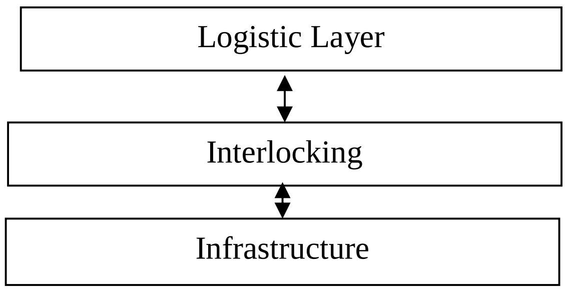

Figure1. 1 Interlocking Layer

Because relay interlocking and computer interlocking are designed based on ladder logic diagrams, Fokkink and Hollingshead proposed to convert ladder logic diagrams to Boolean formulate. Then they applied a theorem prove to verify these Boolean formula.

Winter et al[9] proposed to create two formal models during the design process of interlocking. One is the formal model of the signaling Principles called principle model. The other is the formal model of the functional specification for a specific track layout called interlocking model. The control Tables are translated into an Interlocking model and then checked against the principle model. At first Winter used CSP (Communicating Sequential Processes) as a modeling language but later found that the CSP models of the interlocking system and the signaling principle are difficult to understand and validate. Thus [10] used Abstract State Machine (ASM) notation to model the semantic of control tables. The ASM model was then automatically converted to NuSMV code [10] while the safety properties were modeled in CTL (Computation Tree Logic). Finally [11] they modeled the safety properties in ASM and then translated both ASM models into the NP Proverb tool in order to compare the performance between NuSMV and NP tool. They discovered that if the track layout was divided into smaller segments for verification, the NuSMV outperformed the NP-Tool.

Bor'alv[7] and Peterson Constructed Interlocking programs using a special language called STERNOL, which was developed by ADTranz in Sweden, and verified the interlocking programs using NP-Tools. Because of relay interlocking and computer interlocking are designed based on logic diagrams. The simulated inputs are generated by simulated train movements, system does simple consistency checks on the interlocking outputs, and also verifies the absence of collisions and derailments in the simulate scenarios.

Banci identifies the components required to develop a railway interlocking system as a station yard layout, condition table, state chart design model, model verification, software code and testing [12]. The yard layout displays the geographical layout of the elements. The condition table is based on the yard layout and describes the interlocking principles. State charts are used to model the elements and illustrate how these elements function collectively to form an interlocking system. Faults are injected into the model to verify the behavior of the system.

The software code is then generated and extensively tested in order to guarantee the safety of the interlocking [12].

Moller suggests physical layouts, control tables and ladder logic in specifying and modeling a railway interlocking system [13]. The physical layout illustrates the positioning of the hardware elements such as signals, switch, tracks, etc. These hardware elements have attributes. For example, track circuits can be either occupied or unoccupied. Control tables are derived from the physical layout and define the behavior of hardware elements. Control tables consist of a set of rules of operation that must be obeyed in a railway yard. The physical layout and control tables are used by software developers as input specifications for the software. The software is programmed using ladder logic. Ladder logic is a discrete time linear system. It provides a graphical representation of a logic circuit in the form of Boolean equations. Verification conditions are then designed to validate the software. These conditions are based on entries in the control table and interlocking principles [13].

Interlocking software for electronic railway systems is developed using ladder logic as proposed by Kanso [14]. The software is tested through simulations designed and monitored by signaling engineers. The layout of the railway yard and railway interlocking principles are used as input to the software. Railway yards are composed of track segments, signals and points. Routes consist of connected track segments which begin and end at signals.The routes on a railway yard are defined by a control table which is deduced from the yard layout. The interlocking implements the conditions defined in the control table as Boolean assignments in ladder logic. Propositional logic conditions are defined in ladder logic to verify the safety properties of the system [14].

Eris states that a railway system consists of a TCC (Traffic Control Centre), interlocking system and field equipment [15]. The movement of trains in a railway is monitored in a TCC. The TCC communicates with the interlocking by sending commands such as route requests. The interlocking in turn sends the current state of the system to the TCC which is displayed in the form of a graphical layout. Field equipment refers to signals, tracks and points that are placed in a railway yard. The interlocking monitors and communicates with this equipment.

The interlocking system evaluates the route request received from the TCC, determines if the route can be requested and then sets the positions and indications of the required field elements. State transition charts are used to model the functions of the field equipment. Input-output interlocking functions are deduced based on these states [15].

The conversion of the operational, functional and safety requirements of an interlocking system into a standardized CIS (Computer Interlocking System) has been presented by Lutovac [16]. The CIS monitors and authorizes the movement of trains across a railway. It determines the available routes and signal indications that need to be set. The interlocking software in the CIS is separated into two groups: application driven data and general interlocking software. Application driven data refers to a station layout and control table. The control table must be derived from the station layout by a signal engineer. The data in the control table is then converted into the interlocking software through the use of Boolean interlocking functions. The interlocking software is independent of the application. Signaling principles and railway regulations are incorporated into the interlocking software. A graphical screen layout is designed to illustrate the performance of the system [16].

Minkowitz states that an interlocking is a centralized system which controls the transmission and reception of information from field elements in a rail yard [17]. This information and the state of each element are stored in memory. The interlocking is composed of configuration data and source data. The configuration data is defined for a particular collection of field elements and consists of geographic information regarding the elements and the application of interlocking principles to these elements. The source data creates a visual interpretation of this data at run time. The source data is developed from defined expressions and statements defined by signaling engineers which are used to communicate with the interlocking memory [17].

Winter suggests the development of a principle model and an interlocking model to represent an interlocking system [18]. The principle model consists of the functional requirements of the signaling field elements and the interlocking model consists of the interlocking functions. The interlocking model identifies the key objects in the layout and the relationships between the objects in terms of functions. These functions are strictly based on the signaling objects in the layout. For a different station layout, the functions would change and have to be defined again. Safety properties are built into the principle model.

However the safety variables are derived directly from the station layout and do not ensure that all possible critical events will be covered. Model checking is performed with the aid of test events based on laws of train movement. These laws are assumed and do not ensure all possible events are dealt with [18].

1.7 Thesis Limitation

Some limitations related to the research must be predictable:

❖ Lack of organized data of AA-LRT interlocking signaling system was one of the big limitations of this thesis.

❖ The research focused on the AA-LRT interlocking systems and the data analyses are limited to a particular range of time on a specific railway corridor.

1.8 Thesis Organization

Chapter 1: The introduction, background, problem of statement and objectives of the thesis are presented. More over the methodology, the literature review and the limitation of the thesis discuss in here. In background and motivation parts are description about the railway interlocking and critical safety system. In problem statement, the problem was highlighted. The objective of the thesis indicates safety and verification of the interlocking. Further, literature review with their contribution discussed in here. Lastly the limitations of the thesis are mentioned.

Chapter 2: Type of interlocking system are presented. Some basic Elements of interlocking system are provided. For better understanding of reader a principles for safe train movement, fail safe system, safety standards and fail- safe interlocking system are given. Further, modeling of railway interlocking system is discussed. Finally, Evaluation of the modeling method, the selected methods to developing the tools and introduction of railway routing in Ethiopia are discussed.

Chapter 3: The modeling of railway interlocking system is discussed. In here Modeling of Model-View Controller pattern is described along with the method of agile lifecycle used to develop the verification tools. Lastly for small interlocking system we discusses the physical layout, control tables and state charts.

Chapter 4: Result and discussion: In this chapter, we have selected the development tools that advantage over other development tools. Selected programming language tools have C sharp. Using this programming language are model the interlocking elements. Such as, signals switch and tracks. Lastly result and discussion we looked the simulation of each route and conflicting routes and the result of this simulation. After that, over all the result are discussions.

The last chapter is, about Conclusion and Recommendation: This chapter concludes the thesis with its significance. Further, it gives some future work which gathers during our research.

Chapter Two

Interlocking System

An interlocking system controls and monitors signaling elements in a railway layout. The interlocking establishes the safety of the railway system. As technology has progressed so has the type of interlocking systems and technologies also improve. There are four fundamental types of interlocking System:

- Mechanical Interlocking System
- Electro mechanical Interlocking System
- Relay Interlocking System
- Electronic Interlocking System

In a mechanical interlocking system lever frame connected to the rod and wire pulls and mechanical levers are used to control elements. A person design the systems can visualize the entire system. Collisions are avoided with the use of mechanical blocks which restrict the actions of elements. This type of interlocking system has benefit for ruler area and minimum traffic flow.

Electro mechanical interlocking systems consist of a combination of mechanical and relay interlocking system. Wire pulls are replaced by relays which control the movement of elements. In relay interlocking systems, wire pulls are replaced by push buttons on operator panels and track sections enable electrical monitoring [19]. A relay consists of a number of electrical switches or contacts. Each contact consists of a pair of arms which acts as the switch. If the two arms are not in contact the contact remains open and no current can flow through the switch. However, if the two arms are pushed together by an electric current, the contacts close and current flows through the switch [20]. By interconnecting the relays, the fail safe logic required by the interlocking can be built into the system. However, railway interlocking systems consist of thousands of relays and require a great deal of space to store all the relays. In addition, the relays are physically wire to the signaling field equipment and cannot be stored far away from the station layout. This significantly limits the size of the area and elements that can be controlled. The copper wires used to connect the relays to the field elements are also a significantly high cost factor.

Subsequently, remote control systems have been introduced. These systems enable the monitoring and control of a large area and large group of elements over a single pair of wires. This in conjunction with microprocessors developed into as SSI (Solid State Interlocking) system [20]. SSI systems were introduced in the late 1980s. The SSI equipment for one interlocking consists of several modules housed in a standard sized cubicle. An MPM (Multiprocessor Module) performs the interlocking functions and three of the modules are required in one cubicle.

Each module has identical hardware and software. Each MPM continuously communicates with its two partners comparing operations and memory functions.

All three modules are expected to operate in the same manner. In the event, the behavior of one the modules differs: the specified module intentionally fails and triggers an alarm warning [20]. Four microprocessors are incorporated into each module. One processor in each module processes the interlocking logic and is responsible for communication with the other processors in other modules. The second and third microprocessors manage communication with the field elements. The fourth microprocessor manages communication with the other processors in other modules. [20] However, SSI systems were solely developed for specific processor modules and with constant technological advancements in microprocessors this method is no longer sustainable [21]. As a result electronic interlocking systems became the most advance method of developing a railway interlocking system. Electronic systems can be controlled by arrange of microprocessors and Programmable Logic Controllers (PLC). These micro controllers are programmed in high level languages and are independent of the physical hardware components and the processor modules [21]. These systems electrically monitor railway functions online via a computer screen which provides instantaneous visual feedback to operators. Fundamental operational principles regarding key signaling elements are signals; tracks and switch are described below:

2.1. Element of Interlocking System

Signals

Signal control and monitoring aims to provide necessary information to the diver, central dispatcher and make real-time orders to ensure the real-time action of the system, including:

(1) Manual route control of main line

(2) Signal state monitoring

(3) Switch state monitoring

(4) Track section state monitoring

(5) Route state monitoring

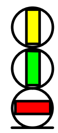

Figure 2. 1 LED signal as the running token in the switch area [3]

LED signal as the running token in the switch area, which has the meaning as follows:

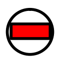

Figure 2. 2. LED signal indicates no passing [3]

Indicates no passing; when seeing the red lamp, the driver must ensure that the train stops in front of the signal.

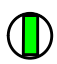

Figure 2. 3. Indicates clear route in the front [3]

Indicates clear route in the front; all the switch directions are straight in the route, and the train must run within the limit speed according to the line.

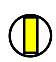

Figure 2. 4. Indicates Clear route in the front [3]

Indicates clear route in the front; at least one of switch directions are lateral. The driver shall reduce the running speed of the train and allow the train run at a lateral speed of passing the switch.

The signal systems have different type of aspect. That is two aspects, three aspects and four aspects. All have there on advantage and disadvantage. The interpretation for three aspect signal indication is explained. A red aspect indicates that the following track section is occupied. A yellow aspect states that following track is clear but the next track section is occupied and a green aspect indicates that the following two track sections are clear. The interlocking monitors and controls the signal aspects to ensure adequate warning are provided for train drivers to stop at a danger signal or to reduce speed in time. Signal aspects are always set to red in emergency situations. Where signals are visible beyond the current approaching signal, more restrictive aspects must be applied to limit the risk of incorrect interpretation of the signal aspect. A signal shall only clear from its most restrictive aspect (red) when the appropriate conditions are fulfilled. These conditions shall continue to be confirmed until the trail has entered the route to which the signal applies. If the conditions are not proved, the signal reverts to a danger indication and displays a red aspect.

A start signal can only display precedes aspect provide the route between the start and destination signals is clear. The defined overlap must be unoccupied and all points in the route must be set in the required position and locked. The start signal of any directly opposing route must be proven to be displaying a red aspect before the start signal can display a proceed aspect.

In addition, for a signal to obtain a proceed aspect, the next signal in the same direction of movement must continuously display a proceed aspect. If no aspect is displayed on a signal (e.g. due to a lamp failure), the preceding signal must shoe a red/stop aspect [5]. In the event of a failure of a signal aspect which could lead the driver to interpret the signal as less restrictive, the aspects shall be tested.

If the test indicates that the aspect is non-functional the signal aspect must be reduced to the most restrictive aspect, i.e. a red aspect and the proceeding signal aspect must be restricted to ensure the signal is not passed at danger [4]. In exceptional circumstances it is permissible for a signal aspect to be extinguished in the event no route has been set up to the signal, no train is approaching within viewing distance of the signal aspects or no confusion will be caused by not showing any aspect indications. If a signal needs to be replacing, it must first be set to display a red danger aspect and the associated route must be cancelled [4].

Train detecting

There are different types of train detection system. Axel counter is one of them. The axle counter is divided into indoor and outdoor equipment, a wheel sensor is arranged respectively at the starting point and ending point of the track section, and used together with the evaluation board for detecting of information of wheel set of the train. Each wheel sensor included two groups of systems, which are used for distinguishing the driving direction of the train. Each wheel sensor is connected with the evaluation board by four core signal cable, which supplies the power for the wheel sensor and transfers the data of axle to the evaluation board.

The axle counting board (ACB) of the axle counting system process the data of axle of all connected counting probes, and outputs the results of vacancy and occupation of the detected section into the relevant system (e.g. interlocking system) in manner of interface conditions of relay (output of vacant section is DC>=21.6V;output of occupied section is DC<=2V)[3]

The block diagram of the system composition is shown as follows:

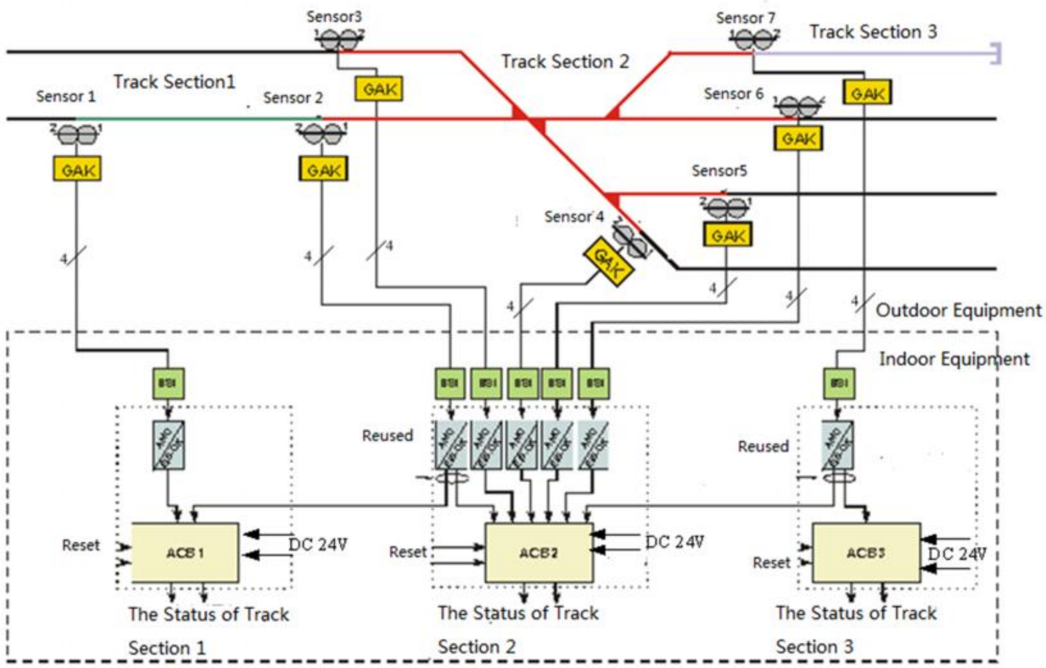

Figure 2. 5 Block diagram of axle counting system [5]

The function of axle counting system is to check and distinguish the state of vacancy and occupation by train on the track section under monitoring. Meanwhile, the distinguishing results are outputted in the form of relay conditions. When the train enter into the track section under monitoring, the condition of loss of excitation representing the occupation of track section is outputted. In case of equipment failure, the condition of loss of excitation for occupation of track section is outputted until the equipment is repaired. The axle counting system can offer two reset ways of pre-reset and direct reset. The resetting way of depends on the edition of ACB software. Pre-reset (simple reset): after pre-resetting, the condition of excitation representing the vacancy of track section is outputted only after the whole train is passing through the section detected. Direct rest (restraint reset-Pre-reset +Reset) after direct reset, the system outputs the condition of excitation representing the vacancy of track section immediately [5].

The interfaces of the axle counting system are divided into internal interface and external interface, in which, the external interface includes the interface for axle counting and interlocking system, as well as interface for axle counting and MSS system. The External interface is between system and interlocking. The interface between host axle counting device and interlocking device is of safety relay interface, including track relay and zero reset relay. In case of initial use of the equipment or fault repairing, it is needed that the person on duty of the station carries out the zero-reset of the section while confirming the state of vacancy of section manually.

The host axle counting device gathers the closed contact of the zero-reset relay for realization of zero reset of the section [5].

Figure 2. 6. Wheel Sensor [3]

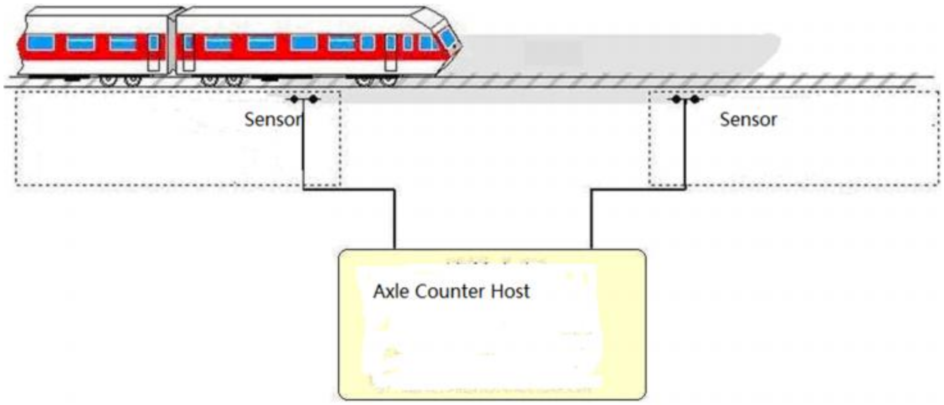

Figure 2. 7. Function diagram of ACS2000 axle counting system [5]

Switch

Switch is a point on the railway network which diverge one rail track in two; or converges two deferent tracks into a single track. The single track is called a stem and the two deferent tracks are classified as left and right branches. In the definition the left and right branches are identified from the stem of the switch. The train can move from any branch to the stem and from stem to any branch while it is impossible for a train to move from one branch to another. Switch operation is controlled by switch rod controller.

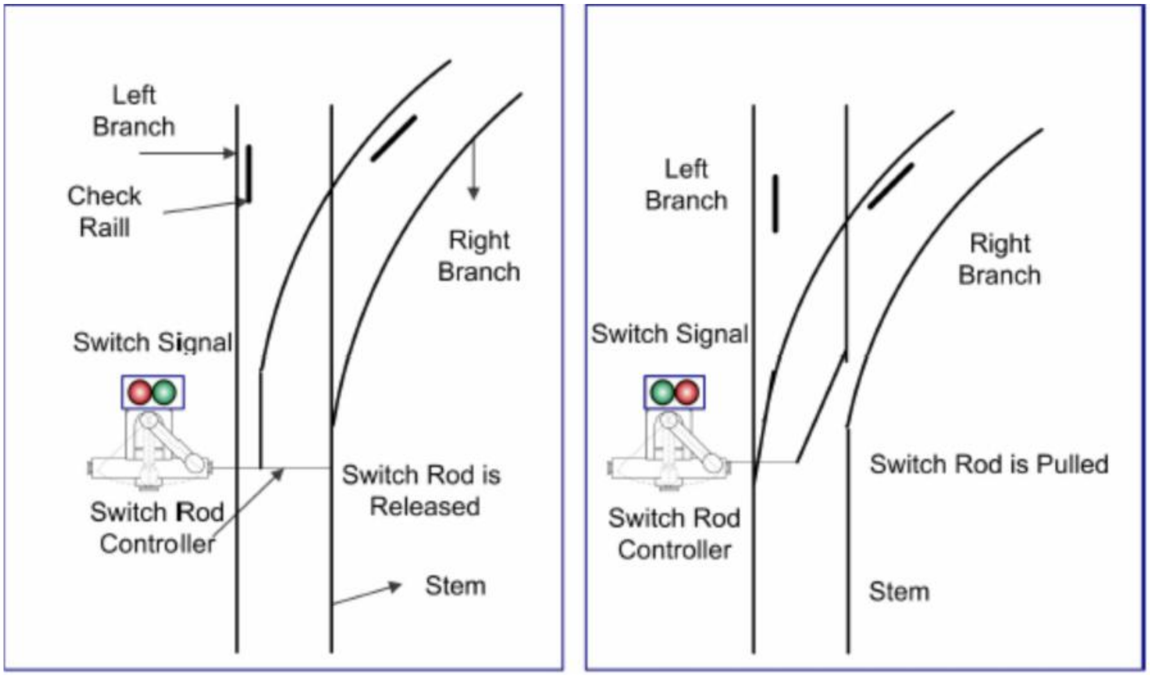

Figure 2. 8. Switch control to left and to right branch [7]

The switch individually operate or locked shall be operated on HMI. The functions include: route lock, section lock and manual lock. The control and indication circuit of the switch machine shall satisfy the requirements. The switch shall be controlled both by single switch calling and route independently, and man's control is prior to route. The interlocking switch is controlled by route lock, section lock, and individually manual lock or other ways. If the switch is locked, it cannot be activated. When the switch starts up, it shall be switched to the correct position. If the switch cannot convert to the required position within the required time because of failure, the alarms will be shown, and the switch shall be operated to its original position. The circuit shall be cut off automatically when the switch completes its switch over. The action current shall be cut off automatically if the circuits of switch machine are fault. The switch has its position display and ensures: the correct display of a switch position must satisfy, the actual position of the switch is in accordance to the operation requirements, and two contacts of the switch machine are in right position. When switch starting, position indication must be shut off in advance. When there is switch split, it shall be displayed. The individually locking of the switch cannot influence the position indication.

2.2 Principles for Safe Train Movement

The most important ideas for safe train movements are the avoidance of safety-critical events and make sure the interlocking responds in a safe manner. Safety-critical events include collisions between trains travelling on the same track or in the same direction [22].In the event a safety-critical event cannot be avoidance, the interlocking must then act to minimize the loss of human life and infrastructure damage by safely controlling the operation of the signaling elements.

The signaling element functions are controlled by preventing the setting of routes with the affected elements [23]. Principles of safe movement on rails state that railway infrastructure must be functional and in good condition such that railway vehicles can safely travel upon the tracks. Railway vehicles must conform to applicable loading specifications. Defined sections in a station must be clear, i.e. sections must be free of all obstacle, required switch must be set in the correct position and the complete train body must be pass/leave a section. Departure and destination points in signal plans and design layouts must be clearly defined. Train drivers and control operators must be aware of the limit of authority given to a train to travel across a designated section. A movement authority shall not allow conflicting train movements along a route. Whilst moving, train drivers must adhere to speed instructions and trackside indications. A train diver must always halt train movement when and where scheduled to do so [23].

2.3 Fail Safe System

A fail-safe system is defined as a control system which either responds in a safe manner by forcing a system into a predefined safe state or continues to function safely after a failure. A safe state refers to the system executing an appropriate response in the event of a critical situation. The fail-safe logic of the interlocking is built into the system. This logic is extracted from the control table conditions which only allows the interlocking to set a route provided all safety requirements are met first [16]. A signaling element that fails should fail in a silent manner and should not affect the operation of other elements. In the event of an element failure, a route which requires this element should not be allowed to be set [12]. An element to fail in a safe manner it must either perform a right or a wrong side of failure. An example of a right side failure is when a signal aspect indicates red when it should actually be showing green. This is incorrect but not a hazardous situation. An example of a wrong side failure is when a signal aspect is displaying green instead of a red indication. This could result in a critical accident [20]. The fail-safe performance of a system can be determined by performing extensive verification through simulation or model checking techniques [12]. Simulations of safety-critical events can also be performed to verify the behavior of the software during these events. Safety-critical situations arise when the maximum number of trains on a route is exceeded, from train derailments, collisions at interlocking system or collisions between trains, etc. Faulty sensors which automatically default the system into a fail-safe state can also lead to safety-critical events. Signaling elements can also cause safety hazards through the incorrect or unrequested switching of signal indication and position of point's machines. Additional hazardous scenarios include if a train does not follow the assigned route specified or if a train driver proceeds through a red signal aspect. These situations all result in a safety-critical situation where in the interlocking must respond in a safe manner. Rail safety is further discussed through the description of applicable safety standards and the integration of safety principles into the interlocking and operation of signaling elements.

2.4 Safety Standard

A safety standard can be achieved by assigning different safety levels to different system functions based on their level of criticality. This is determined based on the overall contribution of the function to the system behavior as defined by the European Committee for Electro technical Standardization (CENELEC) [24].

Developers must follow the CENELEC standards for railway applications to produce safe and reliable systems. The applicable railway applications standards are: EN 50126 (The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) [24], EN 50128 (Software for railway control and protection systems [24] and EN 50129 (Safety related electronic systems for signaling [24]. All graphical system models must be verified to determine required SIL (Safety Integrity Levels) described by standards EN 50126-EN 50129 and IEC 61508 (Functional safety of electrical/electronic/programmable electronic safety-related systems [24].The SIL represents the probability of a system to execute specified safety functions within a specified time interval .The standard defines five critical levels where 0 is a non safety-critical level and 4 is the highest safety-critical level. These levels are defined after an assessment of the level of risk, probability of death or injuries of people, environmental disasters, damage or loss of property, etc. In railway environments, the failure of controllers can cause train derailment or collisions which can result in the loss of human life. Subsequently, the interlocking system for railways is assigned safety critical level of 4.

The interlocking system must satisfy safety requirements for both software and hardware. Software must be developed according to defined standards and in accordance with key fail-safe railway interlocking principles. Safety certified components can be used to satisfy the hardware requirements for the interlocking. The hardware (i.e. signaling field equipment) is also instructed to operate in a defined manner in the event of failure.

These requirements state that route requests received by the interlocking must first proceed through a series of checks which are used to verify the behavior of the interlocking system. These checks ensure that no train can be directed into a route already occupied by another train [12]. To avoid a collision, two trains should never be located in the same track section. Subsequently any two routes cannot share any portion of a track circuit [16]. The train must completely clear the route before another train maybe allowed.

2.5 Modeling of Railway Interlocking

Interlocking system modeling steps according to software engineering methods are: formal methods, life cycle methods, design patterns and software modeling techniques.

2.5.1 Formal Method

The improvement of railway interlocking system from mechanical to computer based interlocking also increased the complexity of hardware and software systems of the interlocking. Moreover, the functionality and scale of such systems have also grown. Because of the increase of complexity, the probability of errors and mistake in the systems becomes greater and design of ultimate bug-free systems becomes harder. Design of such a complex system can be possible with the help of advanced mathematical methodologies and step by step design process. Consequently, the issue of system's complexity attracts the attention of researchers towards the effective methods of modeling and verification of systems. One of the best ways of achieving this goal is by using the formal methods. Formal methods are verbal communications which depend on the use of mathematical procedures and notations to describe and analyze the properties of software systems. These descriptions and notations are usually from the area of discrete mathematics, including set theory, predicate logic, and graph theory.

Following are some important types of formal methods.

2.5.1.1 Sophistication-Based Formal Methods:

Formal techniques can be heavy or light weight techniques. The light weight techniques do not require advance mathematical background and theorem proving skills for usage of them. The light methods techniques that include the feature of light weight are Z-notation and VDM tools. The heavy-duty approach of formal methods uses advance mathematics and theorem proving skills PVS and ACl2 are examples of heavyweight techniques.

2.5.1.2 Semantic-Based Formal Methods:

Formal methods based on semantic foundation are most commonly used for formal specification. Model oriented and properties oriented are two main semantic based approaches of formal methods. VDM, CSP, Stat charts, RAISE, Petri nets and B-Method are common examples of model oriented formal methods. On the other hand, property oriented formal methods enable minimal constraints on the system's behavior, which are necessary. The internal structure of system is not prescribed in detail.

2.5.1.3 Application-Based Formal Methods

According to the system's nature, the different approaches are applied on the system for specification. With respect to the applicability, the formal languages are different from one another. VDM, Z and Larch are used for sequential formalism whereas the Petri net and LOTOS are used for concurrent modeling processes.

2.5.1.4 Graphical-Based Formal Methods:

Petri-nets and SDL are two well-known graphical approaches, which are used for formal specification of systems. Data flow diagram (DFD) and Unified modeling language (UML) are two semi formal methods techniques which are graphical based. Finite state machine (FMS) is considered as formal specification technique.

2.5.2 Life Cycle Method

Typical example to develop software models of life cycle models are agile and waterfall software development method. Jonsson analysed the use of agile practices for software development for European railways [25]. Agile methods are a life cycle model followed during the software development process. These methods follow an iterative development process. Each was iteration produces a usable piece of software that can be released to the customer. Each release goes through the complete development cycle of requirements engineering, design, implementation and testing, Software is developed incrementally and can be validated during early stages of development. There is no limitation to the number of iterations. Agile methods reduce the time and complexity of producing safe systems in hazardous environments [25].

The waterfall life cycle model is a planning-driven process model for the development of software. This model focuses on extensive requirements planning ahead of project implementation. The waterfall model employs a strict phased approach in developing software. Phases are clearly defined and must be adhered to throughout the development process. In each phase, results obtained are compared to those that are actually required. This is known as verification and validation. Verification checks if the system has met its requirements and validation checks if the system has met user requirements. The waterfall process assesses and controls the quality of software produced during each phase [26].

2.5.3 Design Pattern

Railways are considered real-time systems which have bounded response times. These are safety-critical systems, i.e. if the system does not perform in the expected manner during a fault, materials or human lives can be lost. A pattern is a software template which acts as a building block during the development process. A pattern refers to the software itself or to a suite of software processes. Patterns identify key aspects of a design structure and summarize previous successful engineering work which can be reused. Zalewski suggests a top-down approach of applying design patterns when designing a real-time software system [27]. The requirements stage assists in identifying key aspects of a system. The requirements are generally expressed in terms of inputs/outputs. Zalewski assumes that in addition to the functional requirements, every real-time system includes processing and timing components. Together these components form a general architectural pattern for real-time systems. The general real-time architectural pattern consists of a processing interface, user interface, communication link and a database. This pattern can then be adapted for specific real-time applications [27].

Eriksen declares that an interlocking system requires a GUI (Graphical User Interface) through which the state of the interlocking and the system can be monitored and controlled by train operators [19].

This suggests the application of a model-view-controller architectural pattern which consists of three distinct layers. The model layer stores the system data and performs logic operations. The model layer is only aware of its layer. The signaling elements (i.e. signals, tracks, and switch) are created in the model layer. The view layer is aware of the model layer and graphically displays the information stored in the model. The view layer acts as a GUI to the operators. The view layer also displays information communicated between the user and model. The final layer is the controller layer which processes and responds to events in the application.The controller can create changes in both the model and view layers if necessary [19].

Eriksen further suggests modifying the pattern to create a presentation layer. This layer combines the view and controller layers. The presentation layer is then responsible for displaying the information stored in the model and responding to system events [19].

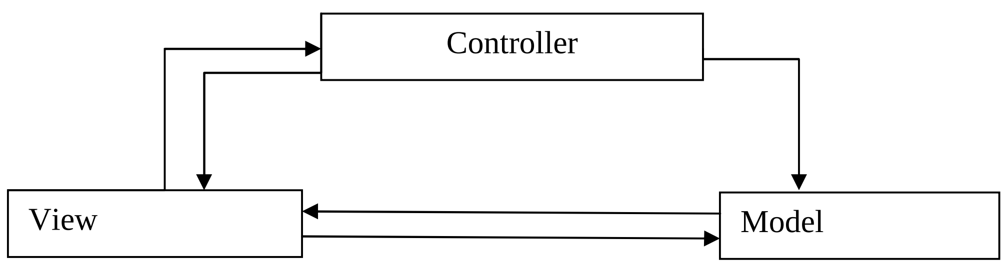

Figure 2. 9. Model-view-controller architectural Pattern

2.5.4 Software Modeling Technique

The use of software modeling techniques in representing an interlocking system is outlined next. Some of these techniques include the use of: State charts, Class diagrams, Automation Petri Nets, Fusion modeling and Z notation. Techniques for the verification of system models and software are also considered.

State charts have been used by Banci to model objects in a railway interlocking system [12]. The functionality and communication between objects are modeled as state charts which collectively represent the interlocking. State charts are extensions of FSMs (Finite State Machines). FSMs are hierarchical parallel state machines which interact with each other. State machines are triggered by events. These events are generated from transitions between states and can refer to states of other machines or to global variables. Global variables are used to send state information throughout the system. UML (Unified Modeling Language) state machines and diagrams are methods of expressing state charts [12].

Eriksen suggests that upon analysis of the railway domain, the signaling elements and their functions can be modeled to describe the operation of the interlocking [19]. Elements are modeled as objects and functions as methods in an object-oriented approach. In this approach, both elements and their functions are modeled together in a UML class diagram [19]. Class diagrams illustrate the static structure of a system by modeling the features and relationships between entities as classes. Objects which have similar features are modeled as abstract classes. Abstract classes allow code to be reused in a system [26].

Automation Petri Nets (APNs) are commonly used to model safety-critical systems such as electronic railway interlocking. APNS are an extension of Petri Nets (PNs) [10]. A PN is a mathematical modeling tool commonly

used in graph and analytic theory. Yildirim uses an APN to graphically model a railway yard [28]. Railway elements such as signals, tracks and points are represented as places (Pxxx). Trains are modeled as tokens. Transitions (txx) represent attributes of elements such as the red indication of a signal or the position of a point. Events (χ) refer to actions such as a route request. An interlocking table consisting of all the routes (χ), required signal indications (txx) and positions of points (Pxxx) is constructed. This table is then used to develop the APN models. The APN models are then converted into a Ladder Logic Diagram (LLD) and a SCADA (Supervisory Control and Data Acquisition) interface is designed to test the software [28].

An interlocking configuration language has been developed by Minkowitz using Fusion modeling with an extension of mathematical notations [17]. This method employs a model-based formal specification technique which integrates easily with object-oriented techniques. The configuration language is composed of semantics specification and syntax specification. The syntax specification defines the structure of the system data and is used as a model on which the semantics specification is based. The model is graphically illustrated using Fusion object model notation. The semantics specification defines the interlocking functions in terms of operational processes executed on the syntax specification. The semantics are defined using Fusion schema notation. Following an object-oriented approach, a set of objects is categorized into a class and the relationship between objects is modeled as a set of tuple [17].

Zafar proposes the analysis of the safety properties of an interlocking system through the use of graph theory and Z notation [29]. This approach is specifically applied to fixed block and moving block interlocking systems. The fixed block interlocking system divides a station layout into fixed blocks and components which are separated by signals. The components within the fixed block are highly dependent on each other. In a moving block interlocking system, the area that a train occupies and the distance in front of it forms part of the moving block.

No other train is allowed to enter this block. Z notation is based on standard mathematical notation used for the specification of abstract properties. This is fundamentally different to description languages [29].

The interlocking software can also be verified through the use of software simulations. Software simulations can be validated using software engineering test techniques to prove the correctness of the software. The adequacy of test techniques is based on the small set of test cases designed to test the software. Dynamic test techniques such as coverage-based testing, fault-based testing and error-base testing can be implemented as these techniques require the execution of a software program. Coverage-based testing evaluates the completeness of the software wherein a number of instructions must be executed or branches followed during the execution of a program. Coverage-based testing also involves the evaluation of program variables through the execution of the software functions. Faults can also be injected into the system to determine the adequacy or performance of the system. Fault-based testing predominantly focuses on the identification of faults in the software[30]. Test stages such as unit testing, integration testing and system testing assess software functions. In unit testing, individual system components are tested. These components are then incrementally integrated into a complete system. This procedure is then tested in the integration testing phase. A system test refers to the testing of software against the user documentation and requirements specification after the integration testing has been completed [30].

2.6 Evaluation of the Method

The examination of the above methods are structured as, first the analysis of interlocking methods and then the examination of applicable software engineering methods. The advantages and disadvantages of each approach method are considered.

2.6.1 Interlocking System

Banci proposes that a yard layout, condition table, state chart design model, model verification, software code and testing are components required to develop a railway interlocking system. A state chart can be used to independently and collectively model the components in an interlocking system. A state chart model effectively illustrates how interlocking functions operate collectively to form an interlocking system. However, modeling all the components can result in a large number of state charts which increases the complexity of the model [12]. In addition, Banci suggests following a geographic approach in developing a distributed railway interlocking system. However, this system does not contain a central database or central equipment which manages the system. In this model the system depends on the geographical layout of the field elements. It also depends on the distribution of the interlocking principles to elements which model the physical field elements. Therefore the interlocking is dependent on the physical railway layout and can only be used for that specific layout. This makes the interlocking hardware dependant and inflexible for use in other station layouts with different elements [12].

Moller suggests physical railway layouts, control tables and ladder logic in specifying and modeling a railway interlocking system. The software is designed using ladder logic. Although ladder logic provides an easily understandable graphical representation, at times a large number of Boolean equations are required to accomplish simple tasks [13]. Kanso also recommends ladder logic for developing interlocking software for railways. The interlocking software is verified through simulations. Simulations are an effective technique for testing software as it mitigates the need for testing on expensive hardware and a greater level of detail can be achieved through simulations. Logic conditions are used to test the safety functions of the system. The logic conditions are similar to Boolean conditions used in ladder logic [14].

Eris proposes that a railway system mainly consists of a TCC, interlocking system and field equipment. State transition charts are used to model the functions of the field equipment and interlocking system. Although state transitions charts increase the ease of traceability in operations, an increase in inputs and states results in more difficulty in generating a state transition table of all the elements. This is especially evident in critical situations [15].

An advantage of the CIS system suggested by Lutovac is that the CIS is independent of the physical hardware layout of the station. This allows it to be used in any country of application. The software is flexible and can easily be adapted. The size of the system is determined by the control table and can therefore be optimized for any application [16]. Although signal engineers are required to complete the control table, simple and easy methods for defining a control table and screen layout are proposed. This reduces the time required for design and input of data. The software is composed of interlocking functions. The input to these functions is the data in the control table. If the control table contains a large amount of data the number of interlocking functions will increase substantially. Nonetheless common variables can be found between functions. This increases the quality output of the system and reduces the probability of errors [16]. Railway and signaling principles are incorporated into the CIS which assists in ensuring the interlocking meets safety regulations. Although these principles are incorporated, signaling engineers are still required to design the station layout, determine the control table and validate the final system [16].

Minkowitz identifies an interlocking as a centralized control system. This system is then separated into configuration data and source data thereby providing a modular and flexible framework. Additional field elements can always be added to the configuration data and would not directly affect the source data. The source data would then just access this data from the memory and graphically display its state. This creates a significantly modular structure independent of the physical hardware. This would also enable easy amendments or additions to the system [17]. Winter also proposes the decomposition of an interlocking system into a principle model and an interlocking model. Both the functional requirements in the principle model and the functions in the interlocking model depend on the station layout and corresponding field elements. Subsequently, for each layout a different model and new functions must be determined. Therefore the model is dependent on the physical hardware and a great amount of time would be required for deducing the models for each station layout. Also, the safety variables incorporated into the system are derived from the field elements in the layout.

2.6.2 Software Engineering Methods

2.6.2.1 Formal Method

The application of distributed formal methods in railway systems provides precise methods for guaranteeing the safe movement of trains. This is achieved by accurately specifying the interlocking rules of operation. These methods provide a list of checks and actions that must be achieved in order to establish a safe functional system. Formal verification is suggested as the most significant way to guarantee the safety of a system [31]. Winter suggests the integration of formal methods into the software during the design phase. Thereafter model checking is used to determine the accuracy of the software prior to development. This is achieved by specifying a formal model of the functional requirements of a physical layout and verifying this against a formal model of the interlocking functionality. The control table is then used to determine the interlocking model.

2.6.2.2 Life Cycle Model

Agile methods used in software development allow software to be validated at an early stage while incorporating feedback and changes into following iterations. The iterative process detects errors early during development and avoids the risk of late system integration. However, these methods have less focus on detailed planning prior to design and implementation of the software. This results in vague requirements analysis and change management. Each iteration is provides a functional release of software. Feedback is incorporated into each release and the software is validated once again. This emphasizes customer collaboration and a faster development process. Whilst agile practices do not offer an effective way of dealing with documentation produced during each iteration, this problem can easily be avoided by only updating documents when necessary, i.e. when notable changes are made. Given that agile methods are not commonly used in safety-critical environments such as rail automation, most agile processes can easily be adapted to suit regulated software development. Subsequently, agile methods can reduce the high costs and time required to produce safe software to meet regulations in hazardous environments [13].

By following the phases in the waterfall life cycle model the quality of software is controlled during each phase. In addition the verification and validation phases ensure the system does not divert from specified requirements. However, in the waterfall model extensive time is spent on planning prior to implementation. Requirements are specified early and cannot be changed during the development process. The waterfall model does not incorporate change. The strict phases are sometimes hard to abide by thereby creating a lengthy process of development [26].

2.6.2.3 Design Pattern

Useful design patterns improve the lack of practice and potential of human knowledge. In addition, defining a generic pattern which can be adapted for real-time applications reduces the complexity of systems. However, design patterns can introduce indeterminism at the design level. Also, the use of generic or multiple patterns may result in insufficient support by software development tools. Conversely, design patterns increase the reusability of software. The patterns are functional and encompass knowledge from skilled engineers and domain experts [27].

In the model-view-controller architectural pattern, field elements are created in the model layer and are only passed to the presentation layer when required. Incorrect inputs entered by a user or received from the GUI can be treated using return values and by throwing exceptions. Although, the presentation layer must always reflect the current state of the model, i.e. the GUI needs to check for changes in the model each time an action is invoked [19].

2.7 Selected Methods to Develop the Tools

From the literature analysis of interlocking methods, essential methods have been selected. The application of these interlocking and software engineering methods in developing automated simulation verification is discussed.

From the above literature review it has been figure out that a Computer based interlocking system consists of a physical layout, control table, system models, and interlocking software and validation techniques. The physical layout symbolize of the geographical topology of the signaling elements on a railway system. Signaling elements refer to signals, tracks and switch which aid in controlling the movement of trains. Based on railway interlocking principles a simple physical layout will be designed. Control table conditions will be deduced from the physical layout. Models of the system will be designed to demonstrate the operation and communication between field equipment and the interlocking software. System components and functions will be modeled using state charts.

Boolean notation will be used to develop interlocking functions. These functions will represent the interlocking principles and define the behavior of the interlocking. Additional Boolean logic equations will be derived to test the software. Simulation of the interlocking software is the preferred method of validating the software as common realistic railway failures and scenarios can be simulated. The tools will be confirmed by introducing faults into the system to determine if the interlocking responds in a fail-safe manner. Formal methods will be used to validate the system and to ensure that the interlocking software meets regulated safety requirements. Agile methods will be followed when developing the interlocking simulation tools. Agile practices offer an iterative development model and enable software to be validated during early stages of development. This is beneficial when developing safety-critical systems [32].

A Model view controller architectural pattern will be followed as it accurately decomposes the functionality of the interlocking system into three layers. The field elements will be created and monitored in the model layer. The interlocking will manage the elements in the controller layer and element attributes and the movement of the trains will graphically be displayed to the operator in the view layer.

2.8 Railway Routing in Ethiopia

In Ethiopia signaling system provide from chinese level standard. The standard gives the signaling system in different place stated in the design. In the depot, when consider the interlocking system the following routes are considered [5].

Route set:

- The correct route shall be set according to the operation intention. It is forbidden to choose conflicting routes at the same time. The conflicting routes are included:
- The opposite train routes in the same track or non switch section.
- The train routes in the same throat section in the opposite directions or same direction.
- Route locking shall be divided into the advance locking and approaching locking.
- Advance locking is constituted when the route is established and interlocking conditions are provided
- Approaching locking is constituted when the approaching section is occupied after the signal is clear. If there is no approaching section, approaching locking shall be constituted immediately after the signal is clear.

Route release

❖ Normal release after the protective signal is closed; each track section shall be released automatically along with train normal running.

❖ Route Cancel when a route that is not in the approach lock status is to be cancelled, it should be unlocked immediately after confirming the signal is closed and the route is unoccupied.

❖ Manual release when a route is approaching locked, cancel the route and the route is only released with delay to ensure the safety of train running. If the train moved into the route, the route cannot be cancelled or released.

❖ Section release Manually(section fault release) when the track sections is power-on, recovery and locked for faults, it shall be released manually by “section Fault Release” and recorded automatically after confirming it is unoccupied and doesn’t be set as a part of the route, release this section.

❖ Call on Route Release after manually confirming the safety of train running, the call on route shall be released via “route release manually” operation.

❖ Route Protection except for protecting the normal train route, the CBI shall set the appropriate overlap to protect the train route according to the need.

❖ Fleet Route the operator shall set parts or all of the signals to automatic status if the route has been set. When cancelling the fleet route, it shall cancel the fleet mode of the route without cancelling the route. The route shall be released by train running.

❖ Cycle Route is a sequence of route triggered automatically by CBI and used to turn back a train. According to the ground signal, the driver controls the train running.

This thesis concentrates at sharing tracks and depot of interlocking system. All the routes in the depot shall be shunting route which include entering into the depot, exiting depot and transferring to another track.CBI has be the complex set the complex shunting route at one time, which is divided into two parts, one is from the signal of entering depot to the stabling siding, the other is from stabling siding to transferring track [5].

The shunting route in the depot shall be released normally. The route shall be locked in advance locking or approach locking and the locked route shall be released section by section with the train running. When the route shall not be released properly, after confirming there is no train on the route via equipment or operator, the sections which are not released normally shall be released manually via the relevant security operation. If the route is approaching locked, cancel the route through Route Release time to ensure the route safety. If the train has crossed the signal, it shall not be cancelled or released.

Chapter Three

3. Modeling of Interlocking

3.1 Chapter Overview

Based on the previous discussion and supporting of the literature investigation, verification using software interlocking model is proposed. The model is expected to cover both railway principles and fail safe logic to guarantee the interlocking performs in a safe manner. These components are built according to CBI system that the current ALRT used.

Firstly the interlocking model specifications are specified. Then the application of appropriate software engineering techniques to the interlocking model is described. These techniques include the application of a Model-View-Controller design architecture pattern and the incorporation of agile life cycle methods during development. The proposed tools model consists of a physical layout, control table and simulation of the train route.

Interlocking Model specifications

The software interlocking modeling is expected to accurately simulate interlocking functions during the movement of a train on a basic physical station layout. The software tool is expected to respond in a safe manner in the event of element failures or conflict route set simultaneously. Interlocking functions must not be permitted if all functional and safety requirements have not been met. The interlocking modeling must hold on to railway signaling principles and conform to CENELEC safety standards for railways [24]. Regarding the functional performance of the software, certain interlocking attributes are assumed in order to simplify the design of the interlocking modeling. These assumptions include that a train can only occupy one track section at a particular time along a route.

The interlocking model follows the two and three-Aspect railway signaling system, i.e. Red and Green for two Aspect and Red (Stop), Yellow (warning) and Green (Proceed) for three aspect.

Software Engineering Methods

Software engineering methods to model have been incorporated to follow an efficient software development process. The application of a model view controller (MVC) design pattern and the integration of agile methods in developing the solution are described.

3.2 Modeling-View Controller Pattern

The interlocking verification tools are designed as an interactive system and the interlocking elements handle input/output data.

After that, an MVC architecture pattern has been followed which divided the system into three layers. That is Model, View and Controller layers. The MVC architecture is illustrated in figure 3.1. **Model Layer**

The model layer stores the data required by the interlocking system such as the current state, the position of the switch and the route status of the elements in the interlocking layout. Database system we use is Sql Server 2005. In this database we create the following tables.

i. Employee Information Table
ii. Routing Table
iii. Track Table
iv. Switch Table
v. Conflicts Route Table
vi. Signals Table

View Layer

The view Layer holds GUI forms and user interactions with the tools. The GUI can display the current state of the interlocking elements and enables users to set and control interlocking operation. User can run the simulation of the train route. In the GUI user can Add, Update and delete the interlocking elements. Using the menus can inject different fault and situation in to the system.

Controller

Use permission to implement changes in both the Model and View layers. This layer is responsible for handling input actions received from the GUI. We use different classes in this layer. Some of classes are Switch position checking, Switch locking, Switch release, Track status checking, Track locking, Track release, and Signal status checking.

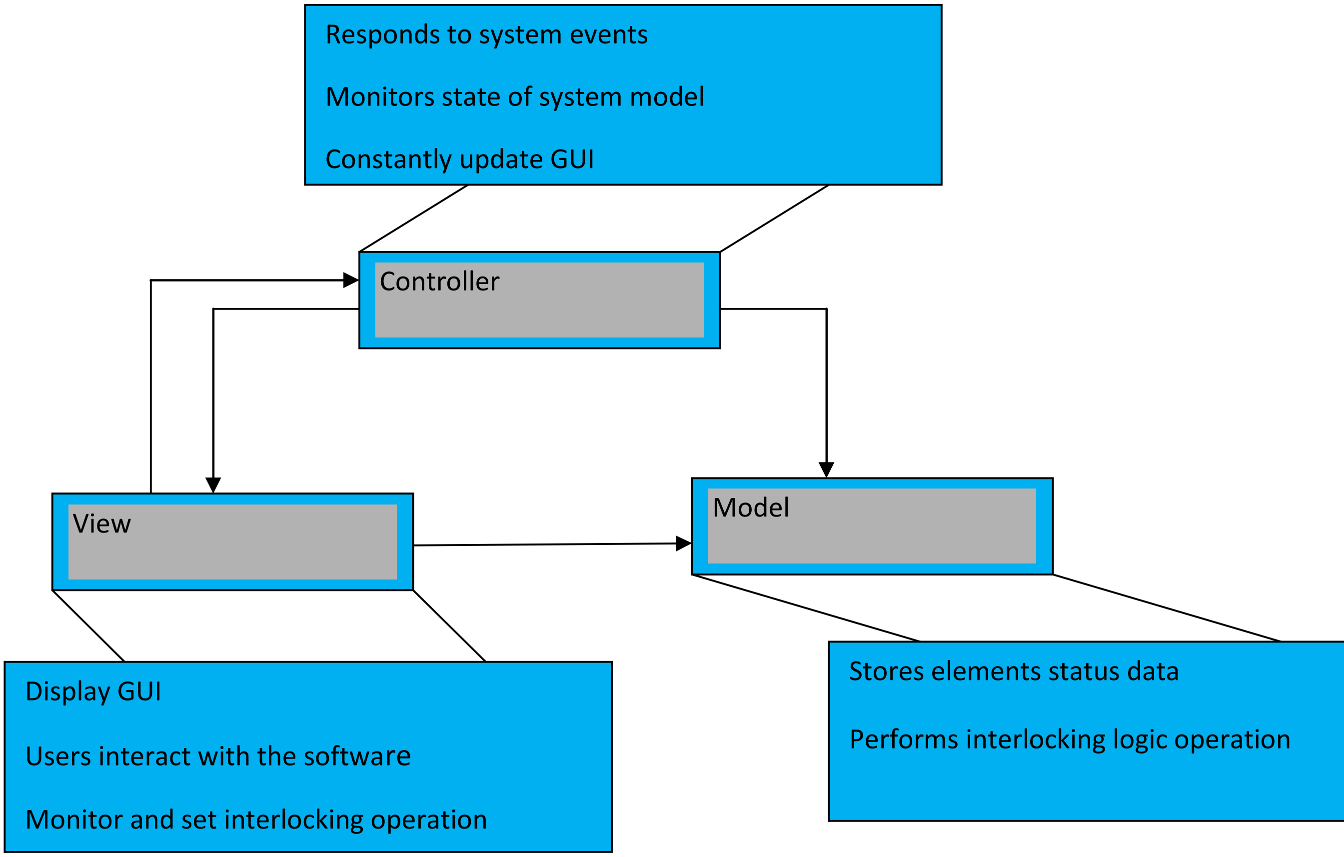

Figure 3. 1 Modeling View Controller Pattern

3.3 Agile Lifecycle Methods

An agile life cycle process has been followed during the development of the software interlocking modeling. Agile methods consist of small and incremental development cycles which are applicable when developing a software interlocking modeling [26]. These methods allow for the incremental development of software by first ensuring that the main interlocking functions are correctly developed. Essential aspects of differing agile methods such as prototyping, incremental development and RAD (Rapid Application Development) are integrated to assist in developing the interlocking software modeling.

Prototyping refers to the development of a working model of a software system with focus on certain aspects of the system. This focus allows for essential requirements to be determined [26]. For example, the software simulation requires a user interface in order to interact with the system. Prototyping methods are used to determine what vital information is required to be provided to the user to interact with the tools. Incremental development methods provide a framework wherein software functionality is developed in small increments. These methods first focus on the essential features and avoid the development of excessive functionality [26].This is beneficial when developing the interlocking software modeling as key functions will be developed first with additional functionality built onto this foundation.

RAD involves aspects of prototyping, incremental development and the reuse of systems. RAD employs the concept of a time box wherein certain functionality must be achieved within a given time frame [26]. This technique is adapted to the interlocking software solution in the form of iterations. The interlocking modeling system is hierarchically grouped into sections of functionality. Each group is referred to as iteration and consists of a set of functions. The fundamental aspects of agile methods such as incremental development, developing a working model and the grouping of system functions, the interlocking modeling software has been decomposed into main and auxiliary functions. These functions are used as a guideline for dividing the model into iterations of development. Each development produces a functional release of tools with key function. The improvement cycle consists of requirements specification, design, and implementation and validation stages as shown in the figure 3.3.

The requirements specification phase discover the major methods required to develop an interlocking modeling. Specific main and auxiliary task is classified as requirement goals per iteration. The outcome of iteration further contributes to the development of the system model. During the implementation phase, the interlocking models are developed. The final stage is validation wherein the model is validated against the requirement goals.

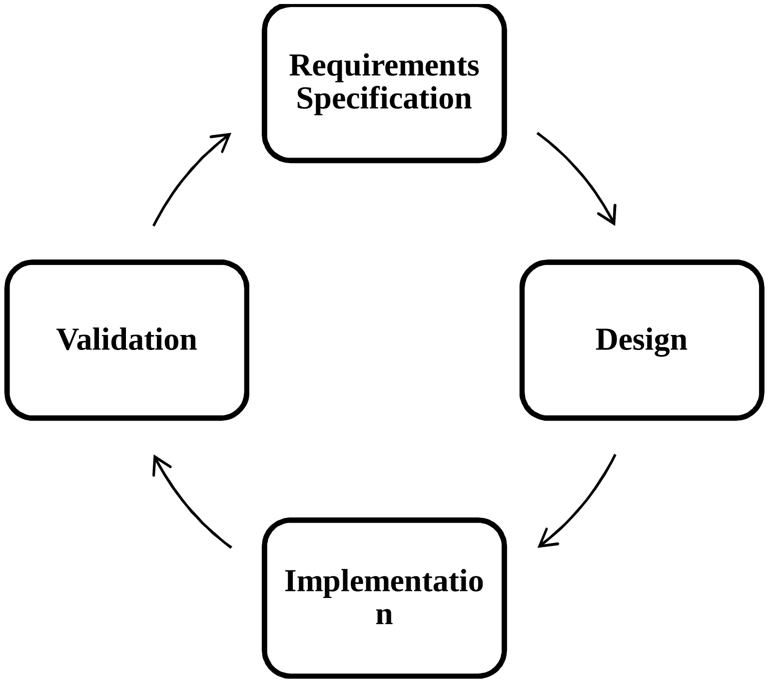

Figure 3. 2. Agile life cycle

In order to develop the interlocking model, a basic physical layout of a railway station has been designed taken from AA-LRT. The layout encompasses all the fundamental signaling elements arranged into a simple configuration. There are four iterations defined in the development of the tools. Firstly physical layout is modeled, class to determine the availability of the signals, track and switch are developed. Thereafter the route request and set route functions' are implemented for each route. The main deliverables in this release include the development of methods to control and set the availability of signaling elements for each route in the control table 3.1.

Table 3. 1 Iteration One

Iteration One Function
Determine the current state of a signal
Determine the availability of a track section
Determine the availability of a Switch
Physical Layout
Route Request and Set Route functions

Table 3. 2 Iteration Two

Iteration Two Function
Adding Updating and deleting of interlocking elements
Employee login and Registration
Route Calling

Table 3. 3 Iteration Three

Iteration Three Function
Locking of Switch
Locking of Track
Locking of the Route

Table 3. 4 Iteration Four

Iteration Four Function
Switch Position change
Track unavailable
Switch status unavailable
Signal unavailable

3.4 Physical Layout

The layout illustrates the geographical arrangement of signaling field elements in a railway system. Railway layouts have been analyzed and model to develop a basic physical layout show in the figure 3.4 and 3.5. The field elements in the layout consist of signals, track and switch. The elements are positioned according to railway interlocking principles and allow for bi-directional travel.

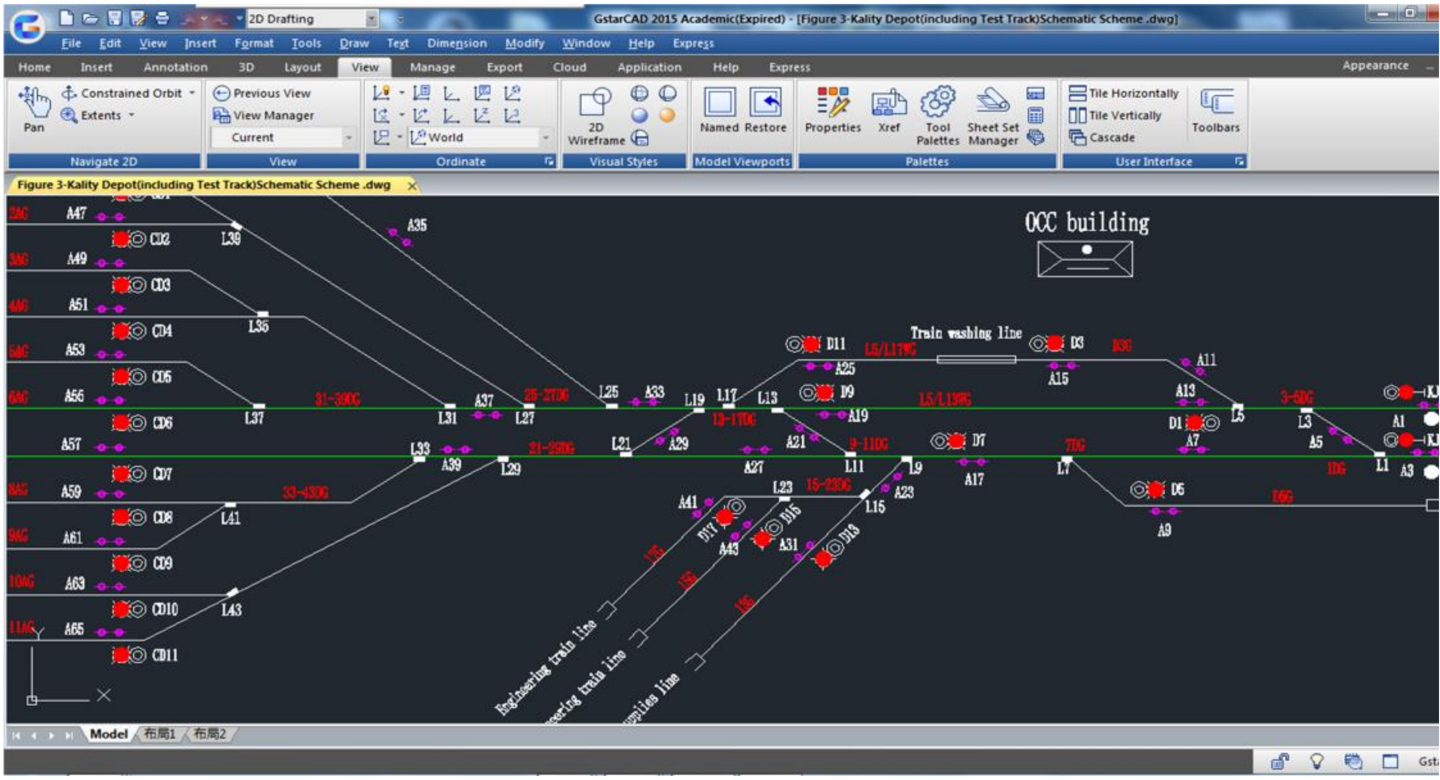

Figure 3. 3 Kality Depot Schematic Scheme [12]

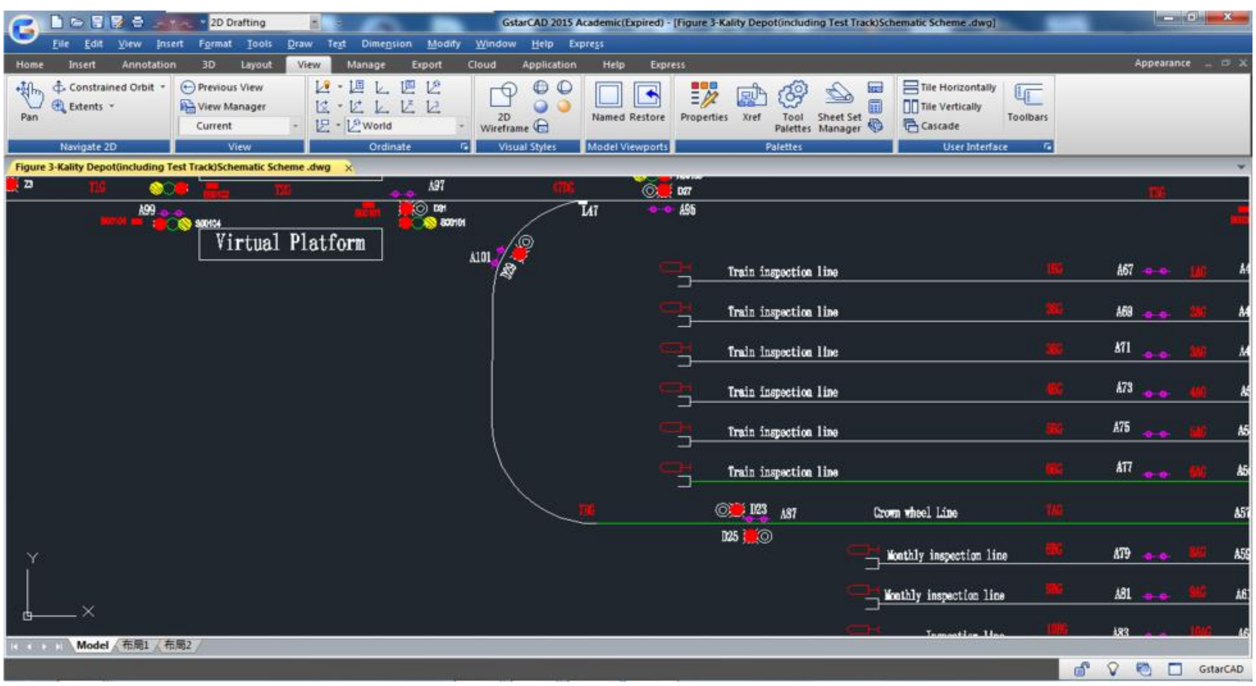

Figure 3. 4 Kality Depot Schematic Scheme [12]

From the above AutoCAD drawing we develop physical layout using C sharp programming language

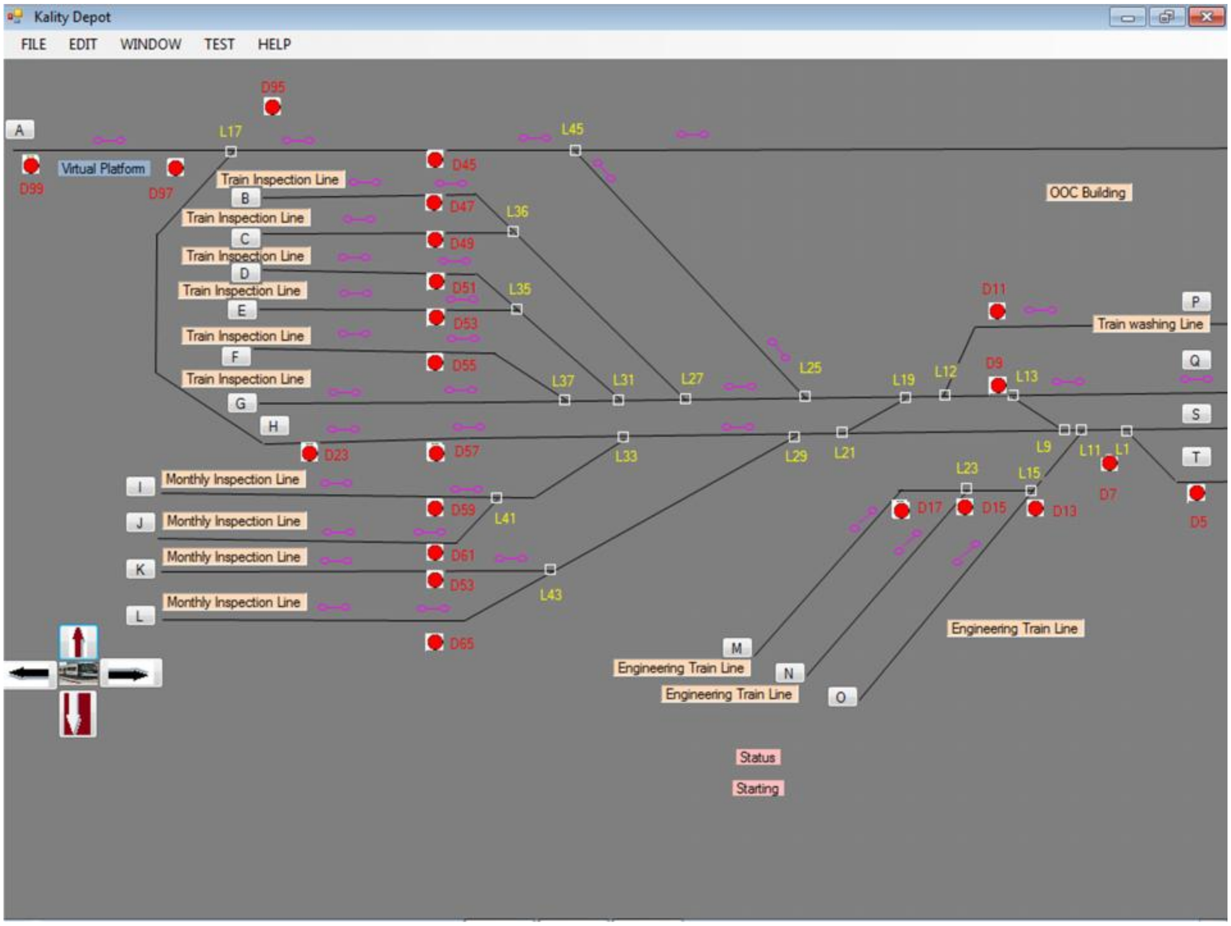

Figure 3. 5 Physical Layout Using C Sharp

Table 3. 5 Control Table Of Track

Route no	Direction		T1 G	T2 G	T4 G	T1 4G	T1 5G	T1 9G	T2 4G	T2 6G	T2 8G	T2 9G	T3 2G	T3 5G	T37 G	T38 G	T39 G
	Start	Dest..															
R1	A	Q	1	1	0	1	0	0	0	0	1	0	0	0	1	0	0
R2	B	Q	0	0	1	0	1	1	0	0	1	0	0	0	1	0	0
R3	L	S	0	0	0	0	0	0	1	1	0	1	0	0	0	1	0
R4	O	S	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
R5	P	G	0	0	0	0	0	1	0	0	1	0	0	1	0	0	1
R6	S	H	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
R7	Q	G	0	0	0		0	1	0		1	0	0	0	1	0	0

Table 3. 6 Control Table of Switch

Route no	Direction Start	Dest..	L1	L11	L12	T13	L17	L19	L25	L27	L29	L31	L33	L36	L37	L43	L45
R1	A	Q	0	0	1 S	0	1 S	1 S	1 R	0	0	0	0	0	0	0	1 R
R2	B	Q	0	0	1 S	1 S	0	1 S	1 S	1 R	0	0	0	0	0	0	0
R3	L	S	1 S	1 S	0	0	0	0	0	0	1 R	0	0	0	0	1 S	0
R4	O	S	1 S	1 R	0	0	0	0	0	0	0	0	0	0	0	0	0
R5	P	G	0	0	1 R	0	0	1 S	1 S	1 S	0	1 S	0	0	1 S	0	0
R6	S	H	1 S	1 S	0	0	0	0	0	0	1 S	0	1 S	0	0	1 S	0
R7	Q	G	0	0	1 S	1 S	0	1 S	1 S	1 S	0	1 S	0	0	1 S	0	0

Table 3. 7 Control Table of Signal

Route no	Direction Start	Dest..	D7	D9	D11	D13	D45	D47	D59	D65	D95
R1	A	Q	0	1	0	0	1	0	1	0	1
R2	B	Q	0	1	0	0	0	1	0	0	0
R3	L	S	0	0	0	0	0	0	0	1	0
R4	O	S	0	0	0	1	0	0	0	0	0
R5	P	G	0	0	1	0	0	0	0	0	0
R6	S	H	1	0	0	0	0	0	0	0	0
R7	Q	G	0	1	0	0	0	0	0	0	0

Table 3. 8 Boolean Equation

oute no	Boolean Equation
R1	R1=T1G*T2G*T14G*T28G*T37G*L12*L17*L19*L25*L45
R2	R2 = T4G*T15G*T19G*T28G*T37G*L12*L13*L19*L25*L27
R3	R3 = T24G*T26G*T29G*T38G*L1*L11*L29*L43
R4	R4= T32G*T37G*L1*L11
R5	R5= T19G*T28G*T35G*T39G*L12*L19*L25*L27*L31*L37
R6	R6=T29G*T38G*L1*L11*L29*L33*L43
R7	R7=T19G*T28G*T37G*L12*L13*L19*L25*L27*L31*L37

3.5 Control Tables

Interlocking Tables or Control Tables are the tabular representation specifying how the train moves together with the states and actions of related equipment. The concept of a train route is the key idea behind an interlocking system. It is used to determine which elements are used with that route. When a route is locked and the signal displays the drive aspect, it means that a train can travel along the route safely i.e. the safety properties will not be violated. When the train has reached the end of the route, the route is released. It is possible to lock several routes at the same time if they are not conflicting. Two routes are said to be conflicting if locking them at the same time potentially lead to a collision.

The control tables have the following list of column header:

- ❖ Route number (The Id number that identify the route)
- ❖ Signaling elements (All the signaling elements shown in the physical layout are listed in the table below. Signals are denoted by D, tracks by T and switch by L. If elements are required for a particular route, the elements are assigned a binary value of 1. A 0 value indicates that the element is not required for the specified route.)
- ❖ Direction of Switch (Direction in which a Switch must lie for a given route such that a train can travel safely across the switch. i.e. Straight or Reverse.) The conditions defined in the control table and the physical layout is used as input specifications for the interlocking software tools.

Based on the Boolean equations determined for each route, a general Boolean equation is defined for all routes in a given station, i.e.

$R_{i=}\pi^{k}{}_{j=1}\, s_j x\, \pi^{l}{}_{j=a}\, T_j x\, \pi^{m}{}_{j=1}\, w_{j,\ i=1\ldots n,}$ where n is the total number of routes

$\pi^{k}{}_{j=1}\, s_{j,}$ where k=i+2 is the total number of signals required for route i.

$\pi^{l}{}_{j=a}\, T_{j\ ,}$where l is the total number of tracks required for route i.

$\pi^{m}{}_{j=1}\, w_{j\ ,}$ where m is the total number of switch required for route i.

where s is the signal interlocking elements

w is the switch interlocking elements

3.6 State Charts

State a condition or way of being that exists at particular time and charts are information given in the form of a graph, diagram or picture to show the information more clearly. Therefore, State charts are used to model the operation of the railway interlocking system. The tool help with the layout of the states and highlights transitions amongst them. Main and auxiliary operations are modeled to show the functional activities of the system.

Main operations refer to the execution of a route request, calling of a route, occupation of a train, etc. Auxiliary operations consist of the sub function involved main operation, i.e. the procedure for determining the availability and current state of signaling elements.

Main Functions

The interlocking functions are modeled to work on railway layout. The following is a description of the state charts developed for the main operations:

❖ Route Request

A route request check if a route can be set for a train to travel in the demand route. The request first determines the starting and destination signals of a route. After that all possible paths that connect these signals are identified. The required sequence of command is illustrated in Figure 3.10.

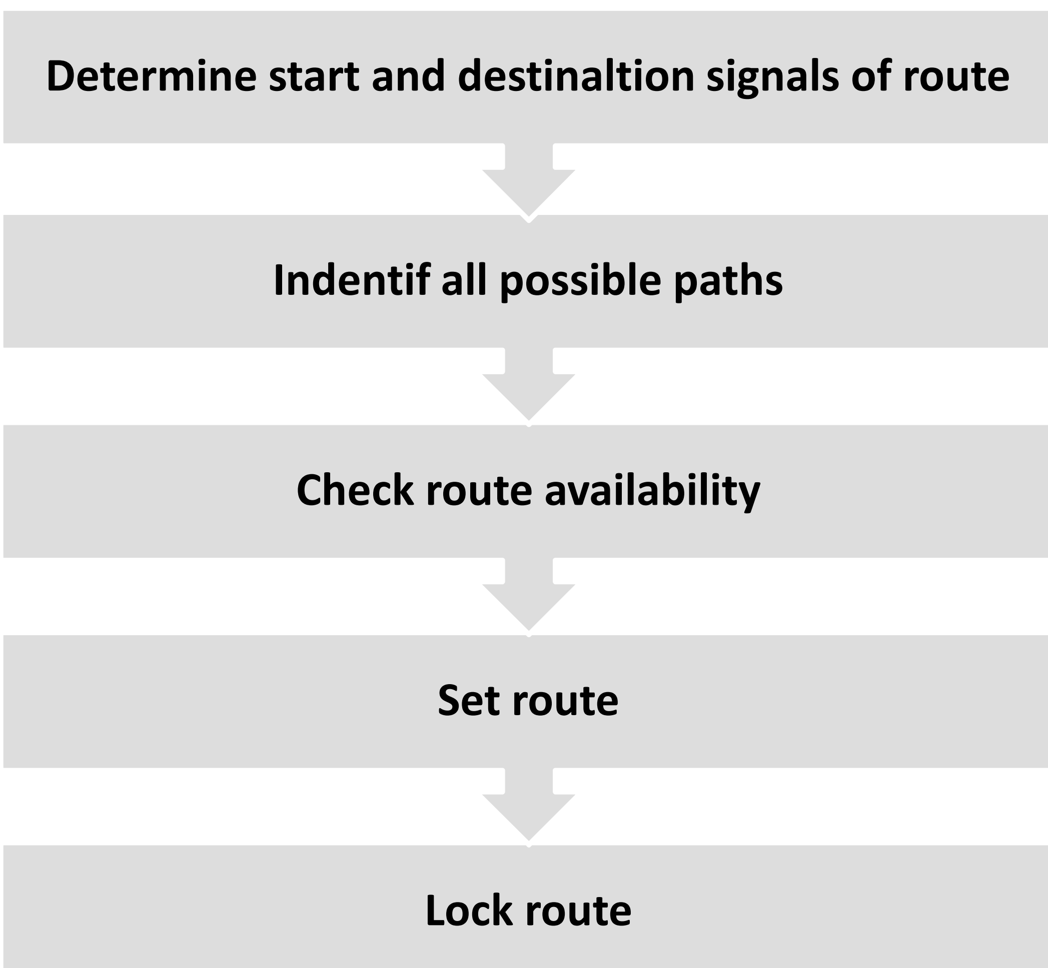

Figure 3. 6. Route Request Sequence

The route request function is divided into the following sub-functions

Route Availability

1. Are there are conflicting routes

Yes then, route is unavailable go to step 1

Else route is available go to step 2

2. The route is available , check each interlocking elements

2.1 Are the signals available

No then, signal unavailable go to step 2

Else signal is available go to step 2.2

2.2 Are the Switch available

No then, Switch unavailable go to step 2.2

2.3 Are the switches in the right position

Yes then, go to step 2.4

Else command the switches in the right position then go to step 2.4

2.4 Are the tracks occupied with another route

Yes then go to step 2

Else go to setting route process

Set Route

1. Select the starting and the destination of signal
2. Command to the system to set the route
3. Set to the element reserved for Request route
4. The cancelation of the route is available
5. If cancelation request go to number 1

Else go to route locking

Lock Route

1. Locked the switch
2. Locked the track sections
3. Finally the entire route is locked

Route call

1. The start signal clears change to green. This indicate that the route is ready for occupation
2. The cancelation of the route still available

2.1. Is the route cancelation requests go to set route

Else continuous the routing

Train Occupation.

Provided the start signal is displaying a green aspect, and the route can be utilized. The train begins by occupying the first track section after the start signal. As the train occupies this track section, it turns Green to indicate that it’s occupied. The start signal then turns red which specifies that the following route is currently being used. The train continues along the route following the same procedure until it stops at the last track section before the destination signal.

Safety-critical Events

In the event of a safety-critical situation, the interlocking is expected to respond in a fail-safe manner, i.e. the state of the signaling elements is reset to a 0 value which indicates that they are occupied. This ensures the elements or route are not used for another route request.

This procedure forces the signals to display a red aspect, the points to remain in the current position and the track sections are set to occupy. A signal can only be in either a red, yellow of green state. A state chart illustrating the possible states of a signal is illustrated in Figure 3.10

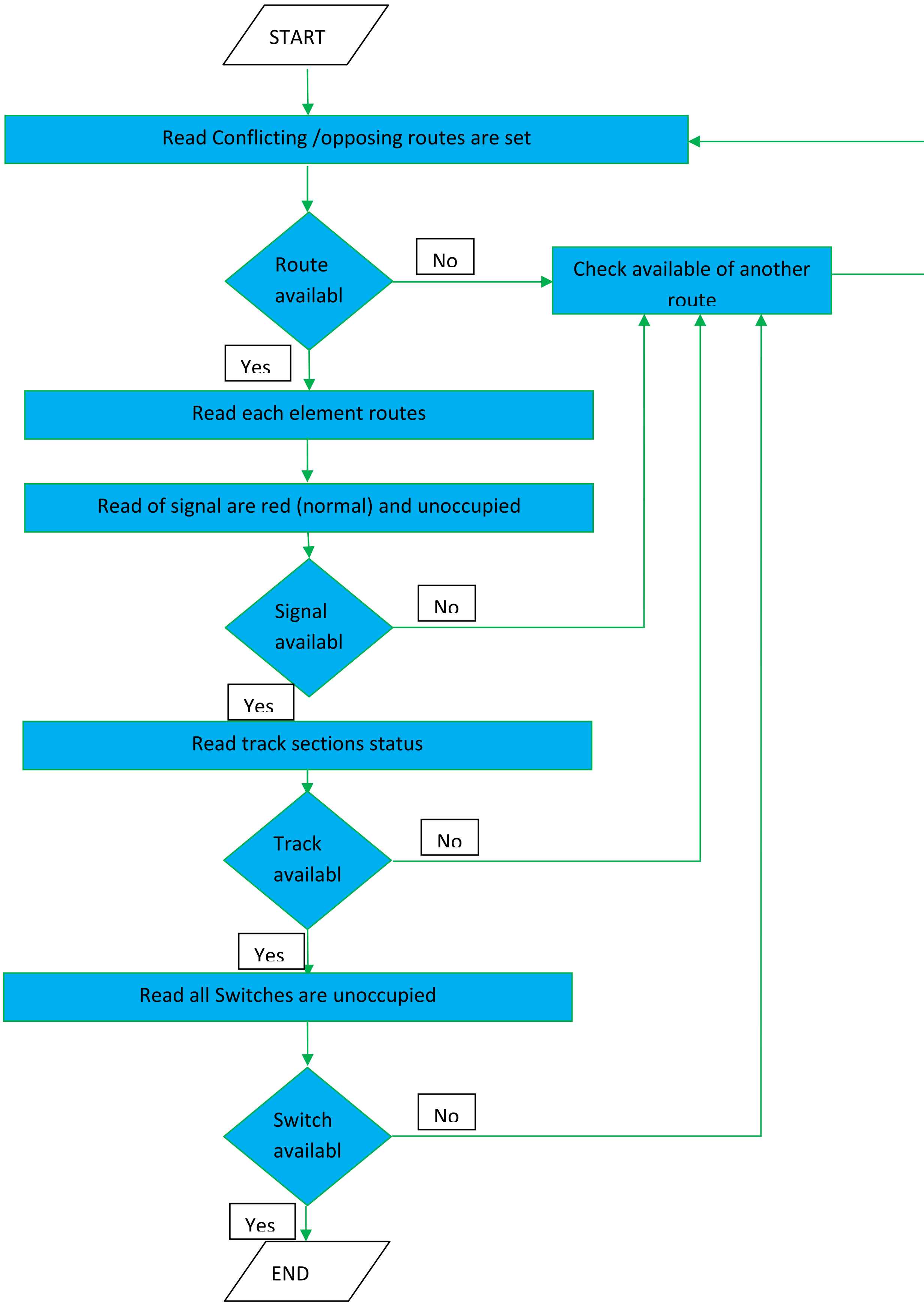

Figure 3. 7 Route Availability

Determine the availability of a track section

1. Determine the current state of a track section if it is free or occupied
2. If the track section is free send to available

Else send to unavailable

Determine the availability of a Switch

1. Are the switch occupied
2. If the switch is free set to available

Else set to unavailable and go to number one

3. Check the position of the switch

3.1. If it is in the correct position, used as it is

Else change to the correct position

3.7 Modeling of Interlocking using a Software

3.7.1. Introduction

There are different types of programming language that can be used to model interlocking system to simulation in given specification. This paper's tool was built using C Sharp or C# programming languages. C# presents a number of benefits that made it seem like the most suitable option. The advantages that C# offers are free and open source, easy to use, powerful and object-oriented and flexible. The Useful classes are that enable developers to develop all kinds of applications ranging from Consol to Web application. C Sharp or C# programming language had been stated in January 1999 by Microsoft for .NET Platform and for the first time published in 2000. C# is common to be a multi-paradigm programming language, covering the concepts of object oriented, functional, component oriented paradigms and etc. C# is general purpose language which suited for developing a wide variety of robust applications for .NET platform from Windows-based to Web-based applications, particularly suits to develop software piece to use in distributed systems. C# is object oriented programming language with strong type checking. Therefore developers can take advantage of creating reusable codes. One of the important design goals of C# is to support internationalization. It supports automatic memory management using Garbage Collector as well. C# is a modern programming language which among the other programming languages is mostly compared with Java. The reason of this comparison is for having many common features similar to Java programming language. But since C# has been invented later than java, the creators have inspired from both the strengths and weaknesses of Java. C sharp similar to Java, it has 70% C++, 10% Visual Basic and 20% new.

3.7.2. Modeling of Interlocking element

In here we will see modeling of each interlocking elements in C Sharp programming language using different tools boxes. The tools boxes have properties and methods that we use for different purpose. Let see one by one:

Track

To model track we use from the tools box line shape that is a graphical control display. We can draw line shape horizontal, vertical and diagonal. For the status of the track on the properties of the line shape Border Color indicates Red for restrict to occupy of the section, Black free of the section and Green Occupied of the section. As seen in the figure

Figure 3. 8. The track free

Figure 3. 9. The Safety Critical Occupied

Figure 3. 10 Occupied of the section

Signal

To model signals we use from the tools box Picture Box that can display Yellow, Red and Green indication. Yellow for warning, Red you have stop, Green it has save to move forward.

Figure 3. 11 Green signal indication

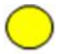

Figure 3. 12 Yellow signal indication

Figure 3. 13 Red signal indication

Axel counter

To model Axel counter we use from the tools box oval shape and line interconnected each other. When the section has occupied we use red color and when the section is free we use purple.

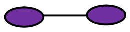

Figure 3. 14 Axel counter that when the section free

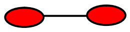

Figure 3. 15 Axel counters that when the section occupied

User interaction

Button

The user can interact with the system using buttons. It has event that can handle button click.

Graphical User Interface

This section is dedicated to the Graphical User Interface design process according to the CBI system. Beside the technical point of view and requirements that have been discussed in previous chapter, from the practical point of view the graphical user interface shall fulfill the following functional requirements.

- ❖ The application shall enable the user to log in.
- ❖ The user shall be enabled to view the track and switch status with all condition.
- ❖ The application shall enable the user to view the interlocking elements in the database and can change any of their data.

This GUI contains the components of the main windows; including classes that manipulating the main window and its sub-components including file menu, View menu, and help menu also the user login window View package is containing Login View.

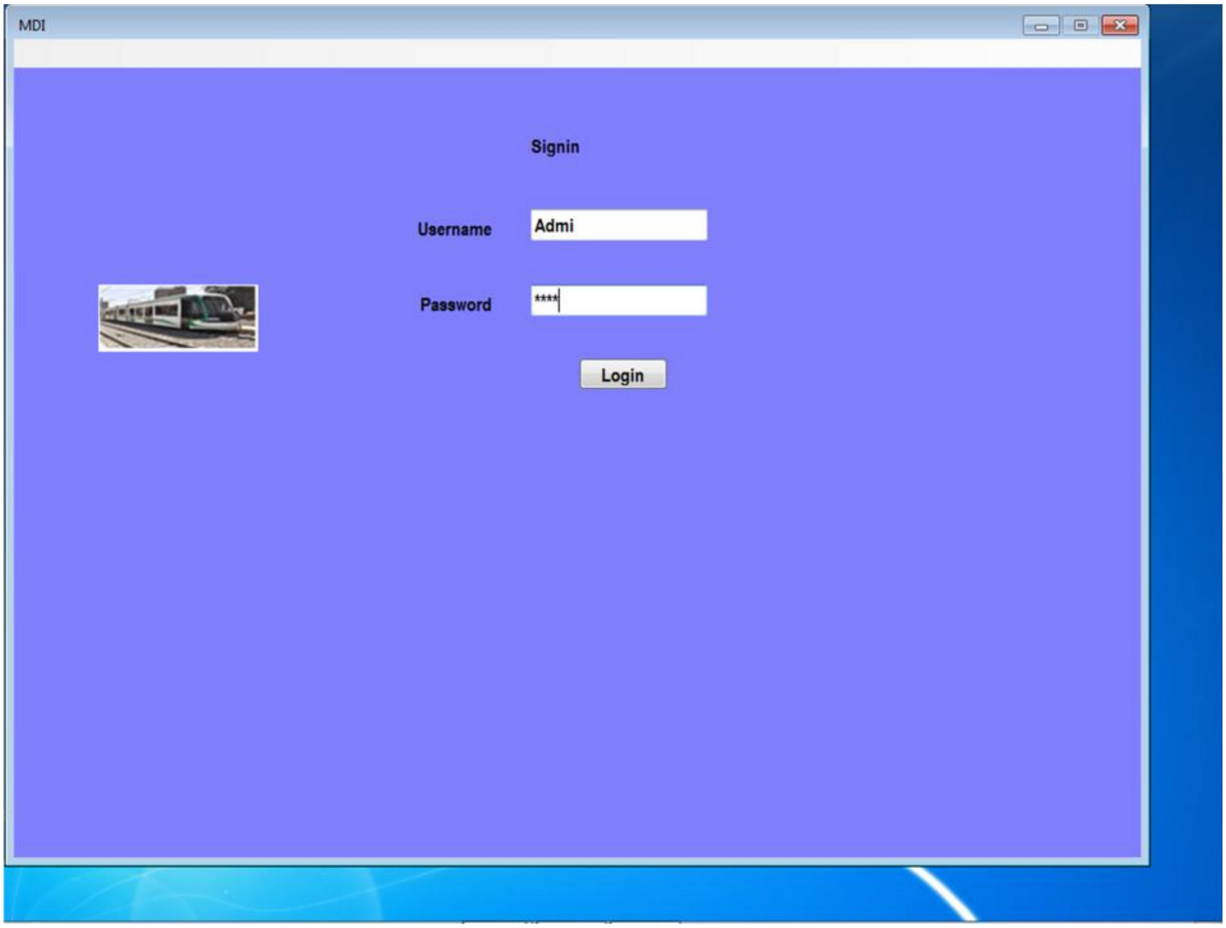

Figure 3. 16 Login View window

User authorization window will appear asking for the username, password. The user information will be sent by Login message and then the validation will be checked in the database. We can implement in several access levels have been defined to access the functionalities within the employee including Controller, Signal Engineer, Track Maintenance, Staff, Administration and Personnel. But our purpose we give equal privilege.

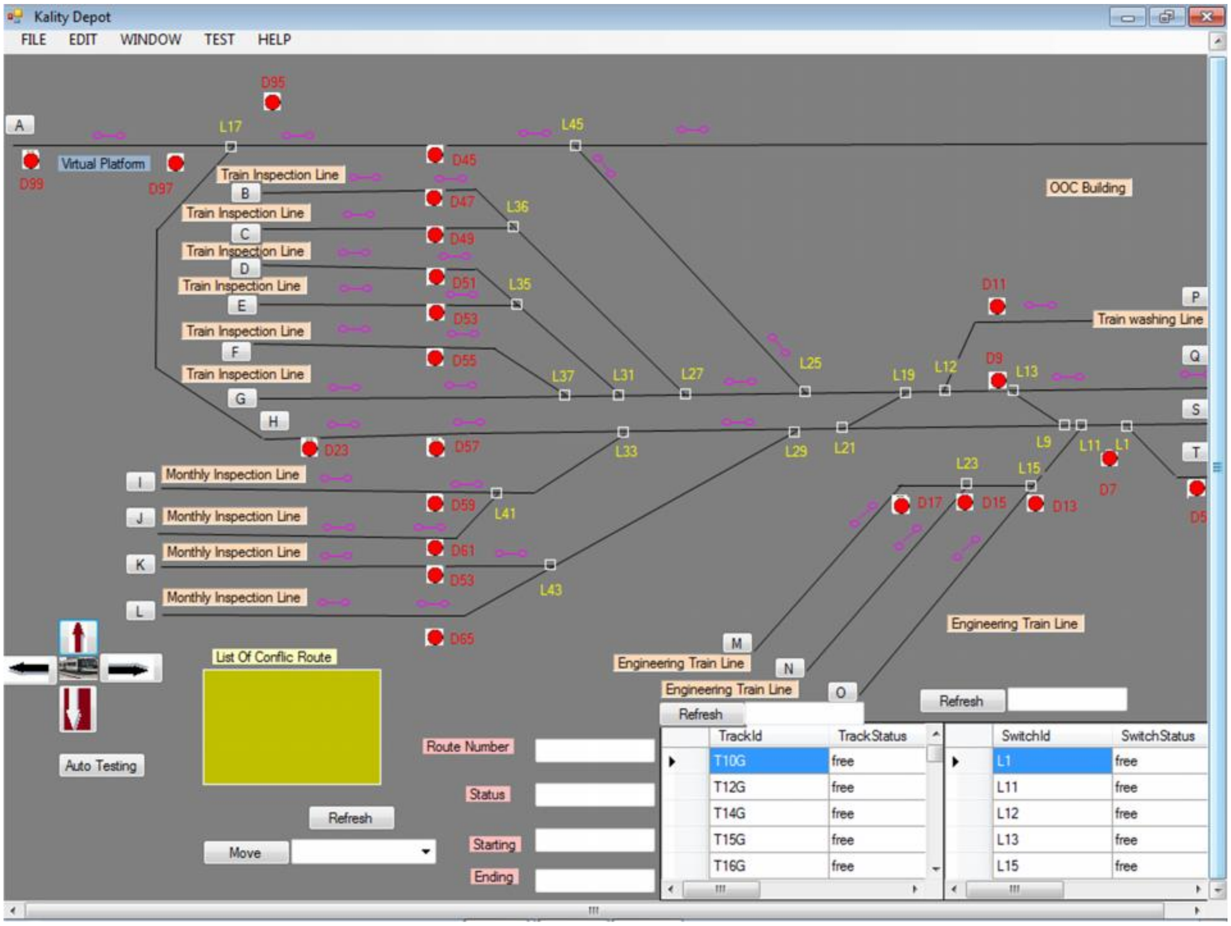

Figure 3. 17 Main Simulation Window

When user authorization is done, the main simulation window will appear. Main window provides user with a menu bar, combo box to select route, buttons, text box and list box that give the information to user. Menu bar control placed at the top of the main window the user to manipulate add, update and delete interlocking elements and also to inject faults to the system.

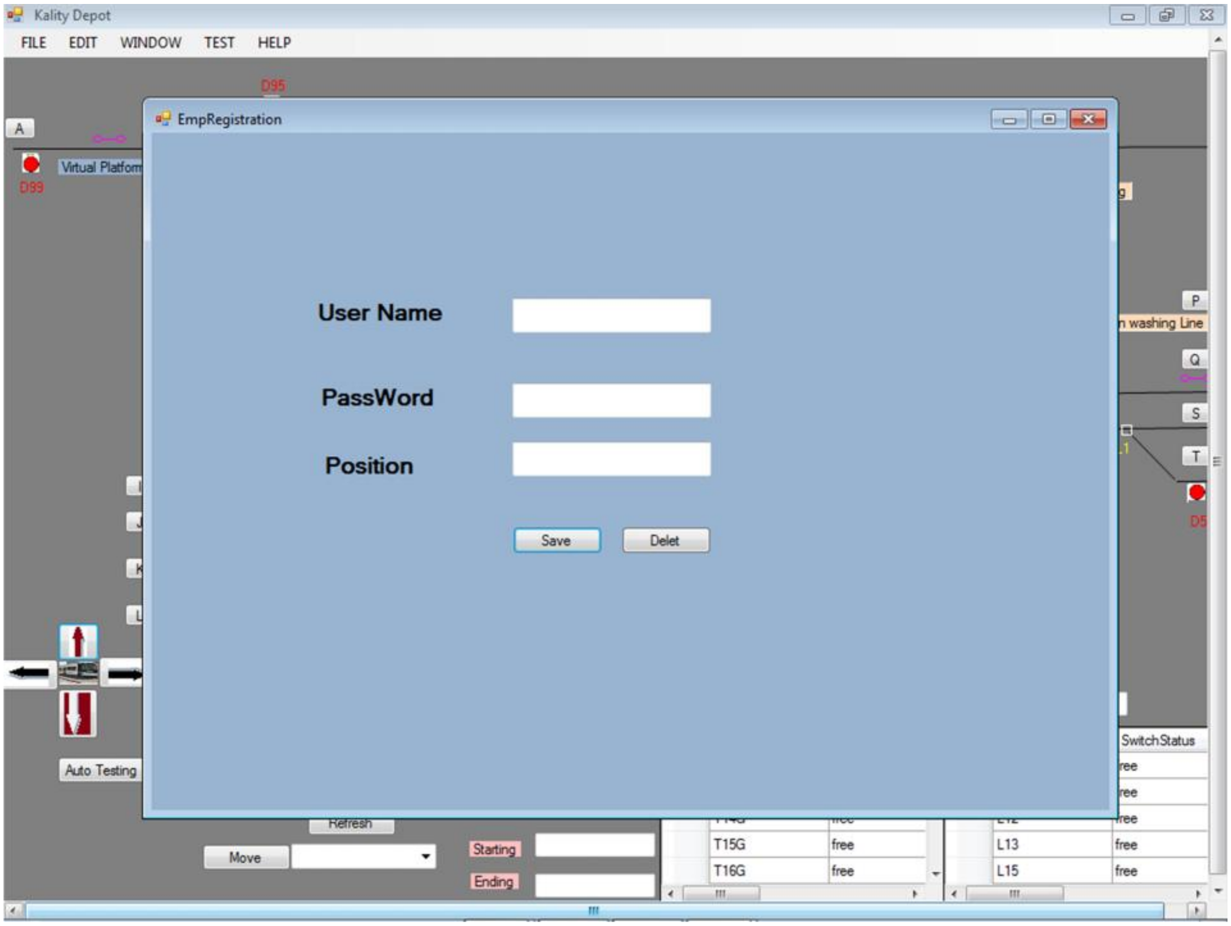

Figure 3. 18 Employee Registration Interface

From the main interface window we can open the employee registration interface that is used to add user's information to the database. This registration needs when login the system he can get authentication to use the simulation tools.

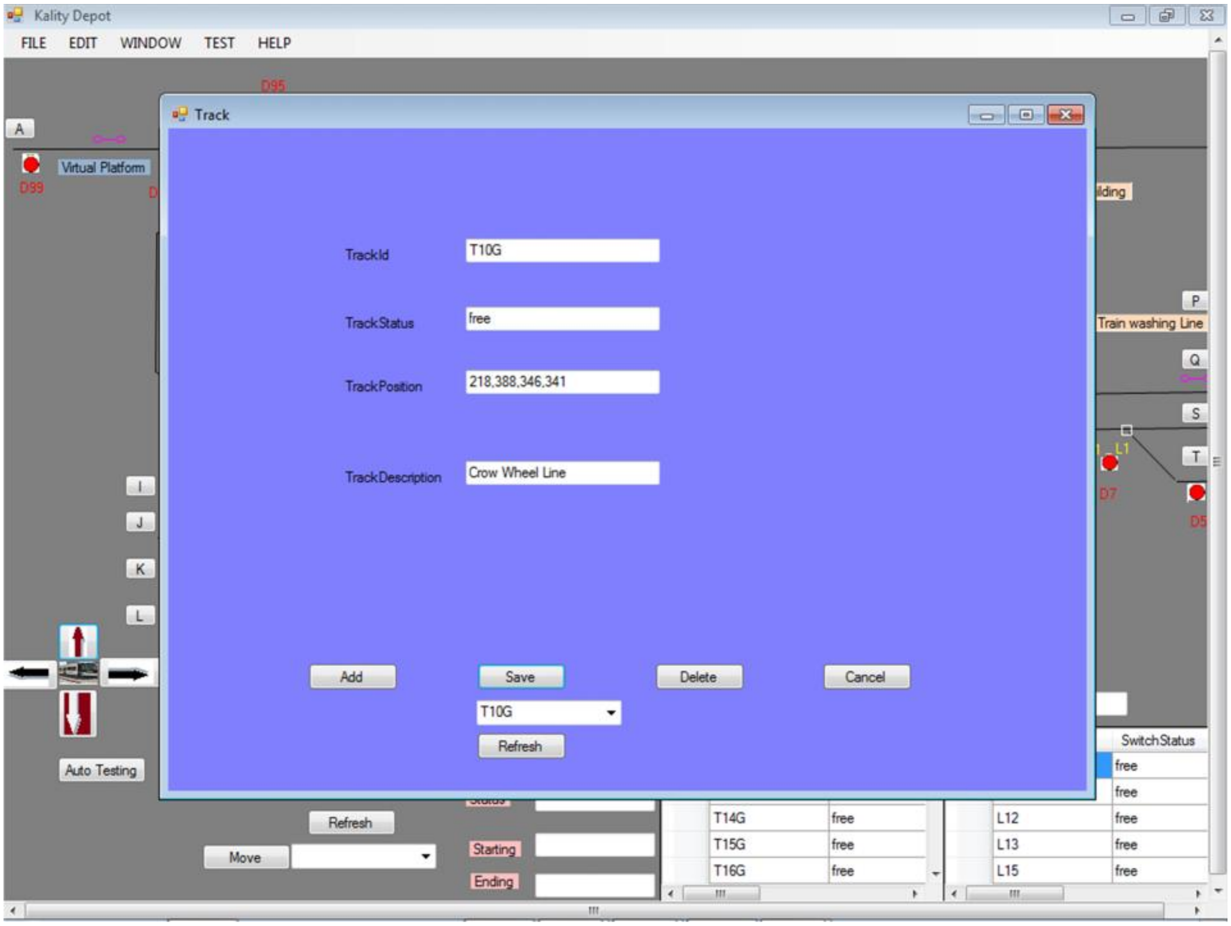

Figure 3. 19 Tracks elements Interface

This is the interface of Tracks elements. You can Add, Update and Delete the track information from the database. When you delete the track the route that used track cannot set to in the simulation. Just like track elements for Switch, Signal, Rout List and Axel counter have the interface that can Add, Update and delete the interlocking elements.

Chapter Four

4. Result and Discussion

4.1. Analysis the Result

The software simulation has been divided into different test cases designed to verify key interlocking functions. The set of test cases has been applied to the sample all seven routes developed in the software tools. The main interlocking functions for requesting, setting and calling a route have been successfully evaluated by simulating these functions and comparing the output to the processes defined in the state charts and the control tables. The occupation of a train in a station is accurately simulated. The train moves in a sequential manner along the route as expected the software simulation tools for verification of CBI interlocking system for railways have been demonstrated. This simulation considers fundamental interlocking functions and main components that form part of an interlocking system.

Analyzing of Kality Depot track layout

The Kality Depot has many tracks, switches and signals. When analyzing the route and the chance of the combination of the interlocking elements: more than 100 acceptable and prohibit combinations can occur. Permitted means the route is safe when the interlocking system work properly. Prohibit means it has combination state of the interlocking elements but the path has unsafe. The safety requirement is to ensure collision-free operation of trains in the kality depot. This is interpreted to mean that each section is occupied by exactly one train or it is empty at any given instance. The following safety properties consider for the Kality Depot in the interaction between the Interlocking and the train.

- Train never derails: Once the switch set to right position and locked for that route never change its position until the train finished its path. For example in our case Route one use L17, L45, L25, L19 and L13 Switch. Therefore we have to make sure always the train move in this route checking of the right position and don't change their position until release that route.
- Train never collides: Two trains cannot occupy the same track at the same time.
- A Switch in not allowed to move when it has the track is occupied with the train.
- In conflict routes are never enabled at the same time.
- Sub routes in opposite directions are never locked at the same time.
- In order to clear the origin signal of a route all the track must be clear(not occupied by a train)

4.2. Test Cases & Results

4.2.1. Route One

Case 1: Route Request

When route request selected we are looking the starting and destination of the route. The status of the route and the route number inform that the selected route. The screen shoots that route one request shown below.

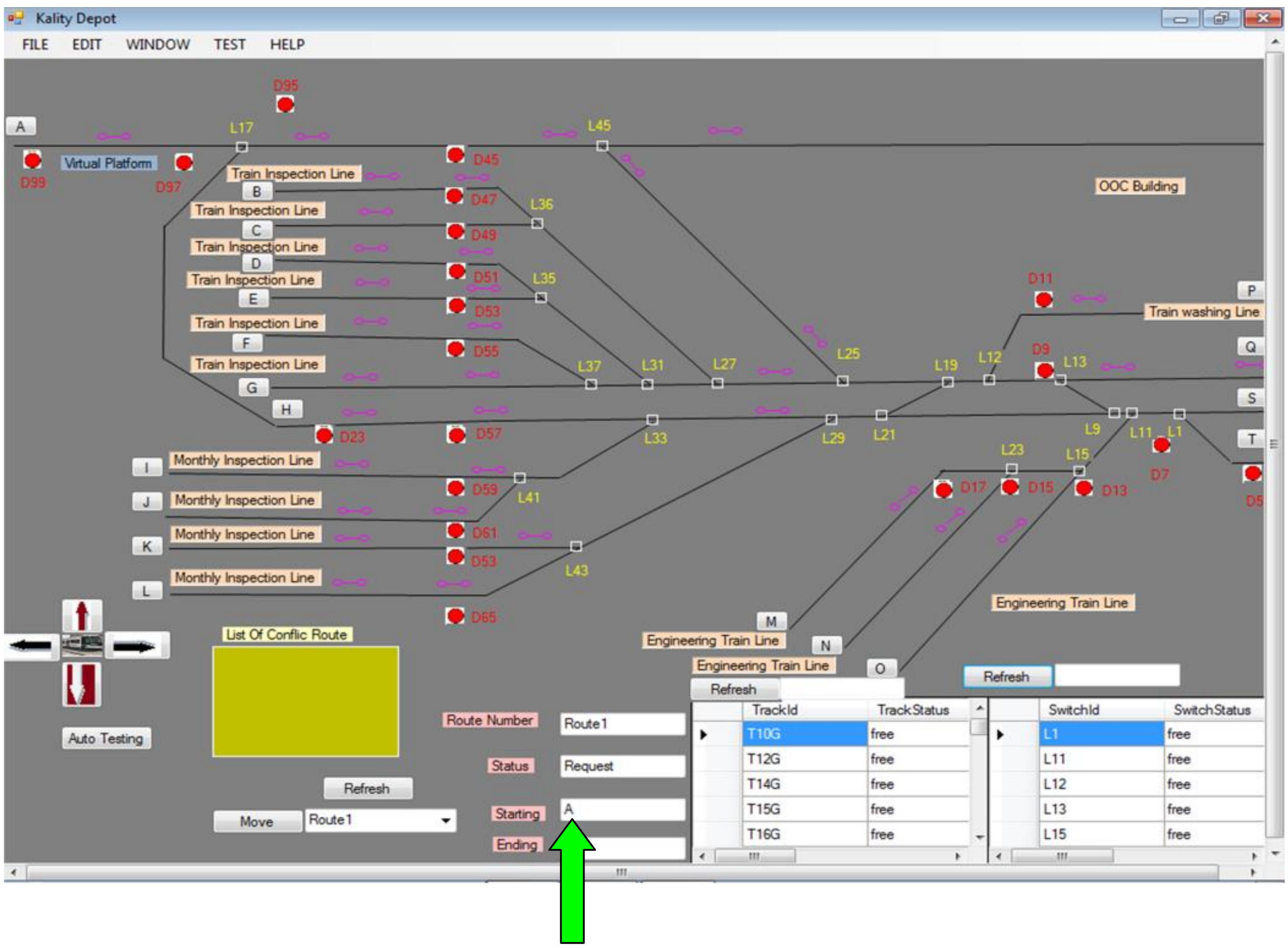

Figure 4. 1 Route One Request

Case 2: Set route

When we click set route that the button move first checking the tracks are free, the switches are free and check the position switch correct if the position not correct command to precise position otherwise leave it. Then Lock the track, the switch and the entire route then give right of entry to move a train on the simulation track.

Table 4. 1 Route one test case parameters

Expected State	Actual State
T1G,T2G,T14G,T28G,T37G=Lock in the Database	Lock
T1G,T2G,T14G,T28G,T37G=Yellow	Yellow(Reserve)
L17,L45,L19,L13,L12= Lock in the Database	Lock
Route one Lock	Lock
Route one move in the track	Route one move in the track

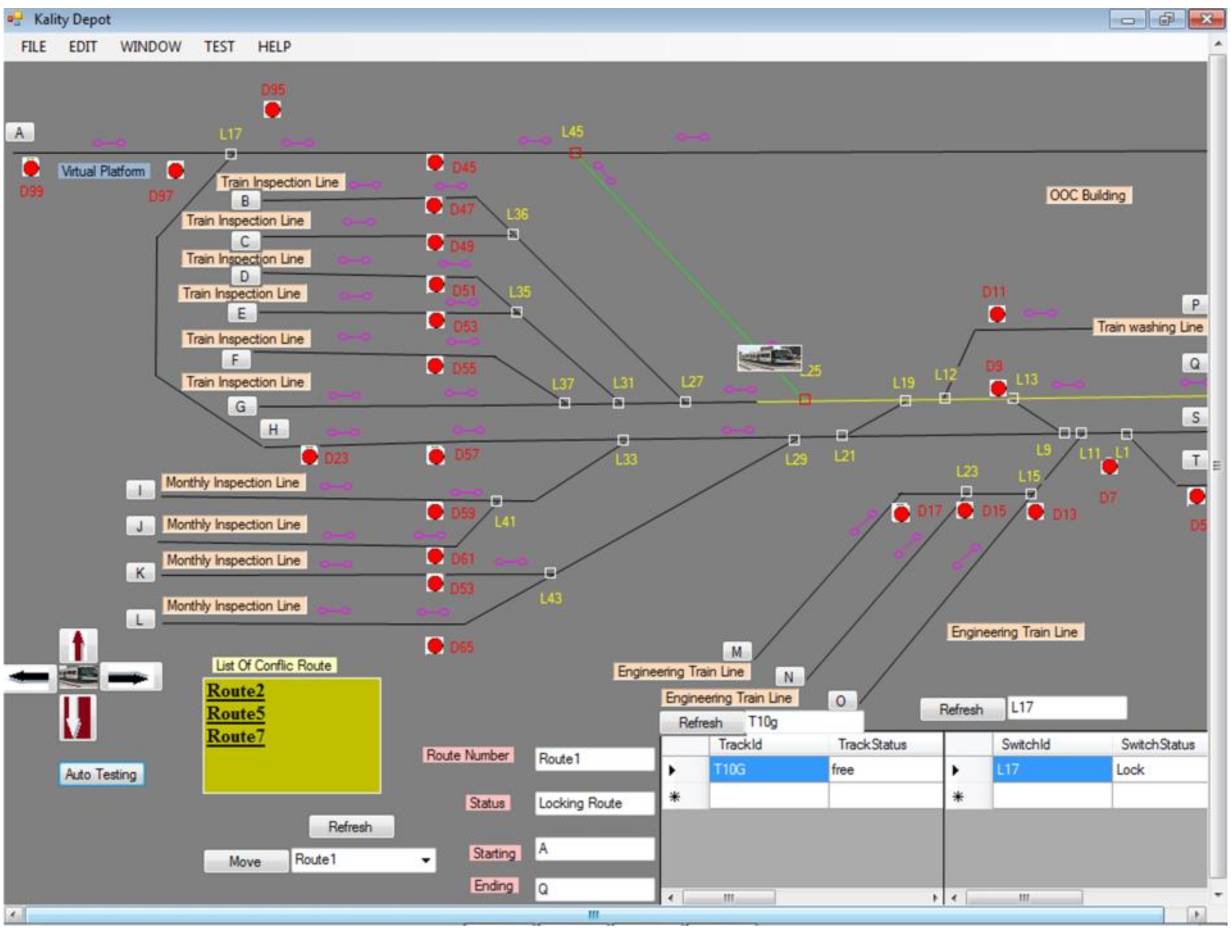

Figure 4. 2 Route One Test Case

Table 4. 2 Route One conflict

Expected result	Actual result
Route Two(Conflict)	Route Two(It is Conflict cannot set the route)
Route Three(Not conflict)	Route Three(It can set the route)
Route Four(Not conflict)	Route Four(It can set the route)
Route Five(Conflict)	Route Five(It is Conflict cannot set the route)
Route Six(Not Conflict)	Route Six(It can set the route)
Route Seven(Conflict)	Route Seven(It is Conflict cannot set the route)

Case 4: Track Occupied Injection

This test case is designed to evaluate randomly injection of track occupation in the database. Some of the Track elements for route 1 are occupied and the remaining element is set to free. This technique is aimed at verifying the logic of the route request function as a route cannot be set if all the Track elements requited for the given route are not available. After the track elements are reset, the set route functions must be selected and the behavior of the system assessed. For all route we checked this fault injection and the result was what we expected.

Case 5: Track elements Unavailable

Just like the occupation of the track we inject the unavailable track elements in the database. Then checking of the route if it is set or not. Each route checked with this fault. The result is what we expected.

Case 6: Change the Switch Position

This is not fault but when the switch position change route must be checked the position of the switch if it is not correct position change the position of the switch in the correct position then lock in sequence of the process. This test case applied for all the seven routs.

Case 7: Switch elements Unavailable

Similar to the above case, after the Switch elements unavailable in the database what we expect is the route cannot set. These testing mechanisms have been done for all routes. The actual results have exactly what we expected.

Case 8: Resetting of switch and track elements

In the medial of interruption or fault injection there is a mechanism that can resetting the switch and the track elements to normal state. This is done during the injection of faults of the switch and the track.

Table 4. 3 Testing Cases Result in the routes

Testing cases	Routes						
	One	Two	Three	Four	Five	Six	Seven
Switch Position	✓	✓	✓	✓	✓	✓	✓
Switch available	✓	✓	✓	✓	✓	✓	✓
Switch locking	✓	✓	✓	✓	✓	✓	✓
Signal available	✓	✓	✓	✓	✓	✓	✓
Track available	✓	✓	✓	✓	✓	✓	✓
Track locking	✓	✓	✓	✓	✓	✓	✓
Route locking	✓	✓	✓	✓	✓	✓	✓
Conflict Route	✓	✓	✓	✓	✓	✓	✓
Track fault injection	✓	✓	✓	✓	✓	✓	✓
Switch fault injection	✓	✓	✓	✓	✓	✓	✓
Checking of status of Track	✓	✓	✓	✓	✓	✓	✓
Checking of status of Switch	✓	✓	✓	✓	✓	✓	✓

4.3. Discussion

4.3.1. Flexibility

The software tools of the interlocking model are flexible and can easily be adapted to suit any interlocking with a minimum modification. You can add, delete and edit the interlocking element such that signal, switch and track.

If we want to upgrade the railway system before go the implementation we can verify, test and analyze the interlocking system easily.

4.3.2. The Verification needs Signaling Knowledge

The verification processes are able to developing the software and can same time learn the railway interlocking principles through execution. Moreover, sufficient signaling knowledge is still required to interpret the station layout, control tables and the GUI. Although certain elements of the interlocking system still require the knowledge of skilled signal engineers

Chapter Five

Conclusion and Recommendation

Major conclusions on the molding of CBI interlocking solution are discussed here. Recommendations for further work and additional improvement to the software model are discussed.

5.1. Conclusion

This paper has described a tool set that support computer aid verification of Computer based interlocking system. This interlocking system have complex and need the means of testing and verification. This verification has been done automatically with aid of computer software using C sharp and Sql. This modeling of interlocking systems include: a physical layout, control table, development models and lastly verification of this model done according to safety standard. This tools support that if fundamental safety standard full fill you can sure safe to move your train confidently.

On this thesis a typical railway interlocking layout encompassing key signaling elements have been developed. From this layout, conditions for train movement are derived and recorded in a control table. The control table conditions for each route in the layout are interpreted as Boolean interlocking functions. These functions are generalized to determine a general Boolean equation for determining which signaling elements are required for each route. The algorithms in the form of state charts have been definitely for key interlocking functions and can be applied to any route configuration. In software simulation modeling each interlocking elements are model using C sharp programming language. A software simulation of the interlocking models has been executed. This simulation proved to be flexible and can apply to most interlocking system. The GUI allows easily be customized for any interlocking application. However, in this customization design additional consideration must be made to include our country specific railway regulations. The verification also incorporates standardized interlocking principles required by all railway systems. The safety standards and fail safe techniques are integrated into the system design which significantly reduces the probability of rail accidents. The proposed system enables the development of a modular system that is easily maintainable and reliable as all functional and safety requirements are incorporated into the system at various stages.

5.2 Recommendation

- On this thesis we are considering the fixed block section; in the future it is recommend studying moving blocks section. Specially connecting of Lebu to Djibouti line.
- The future work include the Study of the automatic generation of test Sequences.

Reference

[1] Prof. Dr-Ing I.A Hansen Real-time Railway Traffic Management: dispatching in complex, large and busy railway networks, December 2010 Rome True University

[2] Zafar N A, Khan S A, Araki K. Towards the safety properties of moving block railway interlocking system. International journal of innovative computing, Information and Control. Volume 8, Number 8, August 2012.

[3]China Railway Group Limited, Ethiopian Railways Corporation (ERC) Project Brief Outline, July 2012 Addis Ababa

[4] China Railway Group Limited, Current projects shaping the future of the rail industry in Africa, Africa Railways Project Report 201

[5] CHINA RAILWAY GROUP LIMITED Certificate of Engineering Design: Comprehensive Class A No.: A151000179 June 2013 Addis Ababa E-W & N-S (Phase I) Light Rail Transit Project EPC (ET/ERC/CREC/0901) Preliminary Design

[6]Wikipedia. List of rail accidents (2000□2009)| Wikipedia, the free encyclopedia, 2010, Last Access Data Juan, 2016

[7] K.M Hansen Formalizing Railway Interlocking Systems, In Nordic Seminar on Dependable Computing Systems, Department of Computer Science Technical University of Denmark, 1994

[8] W.J Fokkink and P R.Hollingshead, Verification of Interlocking: from Control Tables to Ladder Logic Diagrams, In Proceedings of 3rd workshop on Formal Methods for Industrial Critical System(FMICS'98) Amsterdam, May 1998 Stitching Mathematics Centrum

[9] K. Winter Model Checking Railway Interlocking Systems. In Proceeding of the 25th Australian Computer Science Conference (ACSC 2002), volume24, Australian Computer Science Communications, 2002

[10] K.Winter and N.Robinson. Modeling Large Railway Interlocking and Model Checking Small Ones In Proceeding of the Australian Computer Science Conference (ACSC 2003), Volume 25, Australian computer Science Communication 2003

[11] K.Winter, W.Johnston,P.Robinson,P.Strooper,and L.van den Berg, Tool Support for Checking Railway Interlocking Designs, In Proceeding of the 10th Australian workshop on Safety Related Programmable System(SCS'05) Australian Computer Science Communications,2005

[12]Banci M, Fantechi A, Gnesi S. The role of formal methods in developing a distributed railway interlocking system. http://fmt.isti.cnr.it/WEBPAPER/FORMS_04.pdf April.2010,Last Accessed Date April, 2016

[13] Moller F, Kanso K, Setzer A. Specifying railway interlocking systems. Department of Computer Science, Swansea University, May 2010.

[14]Kanso K, Moller F, Setzer A. Automated verification of signaling principles in railway interlocking systems. Science Direct, 2009, pp 1-5.

[15] Eris O, Mutlu I. Design of signal control structures using formal methods for railway interlocking systems. IEEE, December 2010, pp 1-2.

[16] Lutovac D, Lutovac T. Towards an universal computer interlocking system. Facta Universitatis, Vol 11, No.1, 1998.

[17] Minkowitz C, Atkiss J. An object oriented formal specification of a configuration language for interlockings. 3rd Northern formal methods workshop, ALSTOM Signaling

[18]Winter K. Model checking railway interlocking systems. Software verification research centre. Australian Computer Society, Inc, Queensland, Australia, 2001.

[19] Eriksen L E, Perdersen N. Simulation of relay interlocking systems. Lyngby, June 2007.

[20] Kerr D, Rowbotham T. Introduction to railway signalling. Institute of Railway Signal Engineers. London, 2009.

[21] Mitchell I. Managing obsolescence of electronic equipment in signalling. IRSE International Technical Committee, November 2010.

[22] Winter K, Johnston W. Robinson P. Strooper P. Van den Berg L. Tool Support for checking railway interlocking designs. School of Information Technology and Electrical Engineering. University of Queensland, Australian Computer Society, Inc, 2005, pp 4.

[23]Woolford P. Interlocking Principles. Railway Group Standard, GK/RT0060, Issue 4, June 2003, pp 9-23.

[24]Cenelec, Standards Development - Technical Bodies - EN61508-2:2010,- EN50128:2011, - EN50126-1:2012, - EN50129:2003,

[25] Jonsson H, Larsson S, Punnekkat S. Agile practices in regulated railway software development. IEEE, 2012.

[26] Van Vliet H V. Software Engineering Principles and Practice. John Wiley & Sons Ltd, England, Third edition, 2008, pp 50-450.

[27]Zalewski J. Real-time Software design Patterns. June,2009

[28]E.R. Boer and T. Murata. Generating basis siphons and traps of petri nets using the sign incidence matrix. IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications, 41:266{271, 1994.

[29] J. Desel and J. Esparza. Free Choice Petri Nets, volume 40. Cambridge University Press, 1995.

[30] P. Cousot. The role of abstract interpretation in formal methods. In Proc. Fifth IEEE Int. Conf. Software Engineering and Formal Methods SEFM 2007, pages 135{140, 2007.

[31] D. Bjrner. The fme rail annotated rail bibliography. Department of Information Technology, Technical University of Denmark, 1998.

[32]Jonsson H, Larsson S, Punnekkat S. Agile practices in regulated railway software development. IEEE, 2012.

Appendix A Definition

Agile method: – The method of software life cycle process for the development of a software tools. Agile methods follow an iterative method process. The iteration consists of the complete development process of requirements engineering, design, implementation and testing.

Boolean - A data type which consists of either one or two values, i.e. True of False

CENELEC – European Committee for Electro technical Standardization. Defines safety principles that must be followed to make sure safe and reliable systems are developed. These safety standards are applicable to both software and hardware systems. System models must be evaluated to decide the required SIL to be achieved.

Conflicting Movement: – two or more train that share the same track or a movement of trains that may result in an accident or collision.

Control table – A Control Tables are the tabular representation specifying how the train move together with the states and actions of related equipment. A control table consists of a set of rules of operation which must be obeyed for the safe movement of trains in a . The conditions defined in the control table are derived from the station layout. Conditions include route number, required field elements per route, position of elements, etc. The conditions in the control table are used as input for the interlocking software and can also be used to validate the developed software.

Design Pattern – in software development, a design pattern serves as a software template upon which the system is developed. The pattern highlights main aspects of the system architecture and creates a reusable framework which can be applied to other systems. The general design architecture consists of functional requirements, a processing system, user interface and data communication links between objects.

Fail-Safe System – The system will not cause danger to lives or property when it fails. A system’s being “fail-safe” does not mean that failure is impossible, but rather that the system’s design prevents or mitigates unsafe consequences of the system’s failure.

Route – A route consists of start and destination signals between successively connected axel counter and possible switch sets. A train may only be permitted to travel upon an available route. The conditions required for a route are defined in a control table.

Safety-critical System – Is a system whose failure may cause injury or loss of human life, loss or severe damage to equipment or the environment.

SIL – A SIL is determined based on the level of criticality of a system. The SIL level also represents the probability of a system to execute specified safety functions within a defined time interval.

State charts – A modeling technique which illustrates the logical flow of processes in a system. State charts enable the display of multiple hierarchical processes and the different states of system components.

Appendix B Some Software Code

This appendix contains the C Sharp source code for Login input. The program contains connection of Database and check the user has authorized or not to login. If the user name and password match page go to the main interface.

```
using System;

using System.Drawing;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Data;

using System.Threading.Tasks;


namespace Kality_Depot

{

    public partial class MainInterface : Form

    {

        public string r;

      public   string cs = "Data Source=.;Initial Catalog=KalityInterlocking;Integrated
Security=True";

        SqlConnection con1a;


        SqlDataAdapter adapt;

        DataTable dt;

        void textSelect()

        {
```

```
 con1a=new SqlConnection (cs);
            con1a.Open();
            adapt = new SqlDataAdapter("select * from Track where 
TrackId='"+this.txtSelectElement.Text +"' ",con1a);
            DataSet ds = new DataSet();
            adapt.Fill(ds,"Track");
            this.dataGridView2 .DataSource =ds.Tables [0];
            con1a.Close();
        }

        void textSwitchSelect()

        {
            con1a = new SqlConnection(cs);
            con1a.Open();
            adapt = new SqlDataAdapter("select * from Switch where SwitchId='" + 
this.txtElementSwitchSelect.Text  + "' ", con1a);
            DataSet ds = new DataSet();
            adapt.Fill(ds, "Switch");
            this.dataGridView3.DataSource = ds.Tables[0];
            con1a.Close();

        }
        public MainInterface()
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

            con1a = new SqlConnection(cs);
            con1a.Open();
```

```
            adapt = new SqlDataAdapter("select * from Track", con1a);

            dt = new DataTable();

            adapt.Fill(dt);

            dataGridView2.DataSource = dt;

            con1a.Close();

            con1a = new SqlConnection(cs);

            con1a.Open();='


            adapt = new SqlDataAdapter("select * from Switch", con1a);

            dt = new DataTable();

            adapt.Fill(dt);


            dataGridView3.DataSource = dt;

            con1a.Close();

            // TODO: This line of code loads data into the
'kalityInterlockingDataSet5.RoutingPath' table. You can move, or remove it, as needed.


this.routingPathTableAdapter3.Fill(this.kalityInterlockingDataSet5.RoutingPath);

            // TODO: This line of code loads data into the
'kalityInterlockingDataSet4.RoutingPath' table. You can move, or remove it, as needed.

this.routingPathTableAdapter2.Fill(this.kalityInterlockingDataSet4.RoutingPath);

            // TODO: This line of code loads data into the
'kalityInterlockingDataSet3.RoutingPath' table. You can move, or remove it, as needed.

this.routingPathTableAdapter1.Fill(this.kalityInterlockingDataSet3.RoutingPath);

            // TODO: This line of code loads data into the
'kalityInterlockingDataSet1.RoutingPath' table. You can move, or remove it, as needed.
this.routingPathTableAdapter.Fill(this.kalityInterlockingDataSet1.RoutingPath);

            //rtxbInfo.Visible = false;

            comRoutList.Items.Clear();

            SqlConnection con = new SqlConnection("Data Source=.;Initial
Catalog=KalityInterlocking;Integrated Security=True");

            try

            {
```

```
                con.Open();
                SqlCommand command = new SqlCommand();
                command.Connection = con;
                string query = "SELECT  RoutNumber FROM RouteTable";
                command.CommandText = query;
                SqlDataReader reader = command.ExecuteReader();
                while (reader.Read())
                {
                    comRoutList.Items.Add(reader["RoutNumber"].ToString());



                }
                reader.Close();
                con.Close();


            }
            catch (Exception ex)
            {
                MessageBox.Show("Error " + ex);
            }

        }


        private void btnleft_Click(object sender, EventArgs e)

        {

                Train0.Left = Train0.Left - 5;
                label64.Text = Train0.Left.ToString();
```

```
        label65.Text = Train0.Top.ToString();
}
private void btnup_Click(object sender, EventArgs e)
{
    Train0.Top = Train0.Top - 5;
    label64.Text = Train0.Left.ToString();
    label65.Text = Train0.Top.ToString();
}
private void btndown_Click(object sender, EventArgs e)
{
    Train0.Top = Train0.Top + 5;
    label64.Text = Train0.Left.ToString();
    label65.Text = Train0.Top.ToString();
}
private void button1_Click(object sender, EventArgs e)
{
    Train0.Left = Train0.Left + 5;
    label64.Text = Train0.Left.ToString();
    label65.Text = Train0.Top.ToString();
}
    switch (comRoutList.SelectedIndex .ToString ())
    {
        case "0":
            RouteUnavailable1 ru1 = new RouteUnavailable1();
           RouteCheck1 tc11 = new RouteCheck1();
            SwitchChecking1 w=new SwitchChecking1();
            TrackLock tl = new TrackLock();
            SwitchPosition1 s1 = new SwitchPosition1();
            SwitchLock1 sL = new SwitchLock1();
            if (!tc11.routecheck ())
            {   MessageBox.Show("The Route not Available");
                break;
```

```
        }
        else if (!st1.SignalStatus())
        {
            MessageBox.Show("One of the Siganl is not Available");
            break;
        }

        else if (!t.trackchecking())
        {
            MessageBox.Show("One of the Track is not free wait");
            break;
    }
        else if (!w.switchchecking1())
        {
            MessageBox.Show("One of the Switch is not free wait");
            break;
        }

        else
        {
            txtStatus.Text = "Checking Track";
            var t1 = Task.Run(async delegate
                {
                    await Task.Delay(1000);
                    return 42;
                });
            t1.Wait();

            txtStatus.Text = "Checking Switch";
            s1.switchposition1();
            var t21 = Task.Run(async delegate
```

```
{
    await Task.Delay(1000);



    return 42;
});
t21.Wait();
txtStatus.Text = "Checking Switch Position";
sL.switchlock1();
var t3 = Task.Run(async delegate
 {
     await Task.Delay(1000);
     return 42;
 });
t3.Wait();
txtStatus.Text = "Locking Switch";
var t4 = Task.Run(async delegate
  {
      await Task.Delay(1000);
      return 42;
  });
t4.Wait();
txtStatus.Text = "Locking Track";
tl.tracklocking1();
var t5 = Task.Run(async delegate
 {
     await Task.Delay(1000);
     return 42;
 });
t5.Wait();
ru1.routeunavailable();
txtStatus.Text = "Locking Route";
```

```
                T1G.BorderColor = Color.Yellow;

                T2G.BorderColor = Color.Yellow;

                T14G.BorderColor = Color.Yellow;

                T28G.BorderColor = Color.Yellow;


                T37G.BorderColor = Color.Yellow;

                R1 = true;

            }

            break;

        case "1":

            Routecheck2 rc2 = new Routecheck2();

            RouteUnavailable2 ru2 = new RouteUnavailable2();

            SignalStatus2 st2 = new SignalStatus2();

            TrackStatusChecking2 t2 = new TrackStatusChecking2();

            SwitchStatusChecking2 s2 = new SwitchStatusChecking2();

            SwitchPostion2 sp = new SwitchPostion2();

            SwitchLocking2 sl2 = new SwitchLocking2();

            TrackLocking2 tl2 = new TrackLocking2();

            if (!rc2.routecheck())

            {

                MessageBox.Show("The Route not Available");

                break;

            }


            else

                if (!st2.SignalStatus())

            {

                MessageBox.Show("One of the Siganl is not Available");

                break;

            }

           else  if (!t2.trackstatuschecking2())

            {
```

```
    MessageBox.Show("One of the Track is not free wait");
    break;
}
else if (!s2.switchstatuschecking2())
{
    MessageBox.Show("One of the Switch is not free wait");

    break;
}
else
{
    txtStatus.Text = "Checking Track";
    var t1 = Task.Run(async delegate
    {
        await Task.Delay(1000);
        return 42;
    });
    t1.Wait();
    txtStatus.Text = "Checking Switch";
    t1 = Task.Run(async delegate
    {
        await Task.Delay(1000);
        return 42;
    });
    sp.switchpostion2();
    txtStatus.Text = "Checking Switch Position";
    t1 = Task.Run(async delegate
    {
        await Task.Delay(1000);
        return 42;
    });
```

```
            sl2.switchlocking2();
            txtStatus.Text = "Locking Switch";
            t1 = Task.Run(async delegate
            {
                await Task.Delay(1000);
                return 42;
            });
            tl2.tracklocking2();
            txtStatus.Text = "Locking Route";

            t1 = Task.Run(async delegate
            {
                await Task.Delay(1000);
                return 42;
            });
            ru2.routeunavailable();
            T4G.BorderColor = Color.Yellow;
            T15G.BorderColor = Color.Yellow;
            T19G.BorderColor = Color.Yellow;
            T28G.BorderColor = Color.Yellow;
            T37G.BorderColor = Color.Yellow;
            R2 = true;
        }
        break;
    case "2":
        RouteCheck3 rc3 = new RouteCheck3();
        RouteUnavailable3 ru3 = new RouteUnavailable3();
        SignalStatus3 sit3 = new SignalStatus3();
        TrackChecking3 tr3=new TrackChecking3 ();
        SwitchStatusChecking3 st3=new SwitchStatusChecking3 ();
        SwitchPostion3 sp3 = new SwitchPostion3();
```

```
SwitchLock3 sl3 = new SwitchLock3();

TrackLock3 tl3 = new TrackLock3();

if (!rc3.routecheck())

{

    MessageBox.Show("The Route not Available");

    break;

}


else if (!sit3.SignalStatus())

{

    MessageBox.Show("One of the Siganl is not Available");

    break;


}

else if(!tr3.trackchecking3 () )

{

     MessageBox.Show("One of the Track is not free wait");

    break;

}

else if (!st3.switchstatuschecking3())

{

    MessageBox.Show("One of the Switch is not free wait");

    break;

}

else

{

    txtStatus.Text = "Checking Track";

    txtStatus.Text = "Checking Switch";

    sp3.switchpostion3();

    txtStatus.Text = "Checking Switch Position";

    sl3.switchlock3();

    txtStatus.Text = "Locking Switch";
```

```
            tl3.tracklock3();

            txtStatus.Text = "Locking Route";

            T24G.BorderColor = Color.Yellow;

            T26G.BorderColor = Color.Yellow;

            T29G.BorderColor = Color.Yellow;

            T38G.BorderColor = Color.Yellow;

            ru3.routeunavailable();

            R3 = true;

   }                    break;

    case "3":

        RouteCheck4 rc4 = new RouteCheck4();

        RouteUnavailable4 ru4 = new RouteUnavailable4();

        SignalStatus4 sit4 = new SignalStatus4();

        TrackStatusChecking4 ts4 = new TrackStatusChecking4();

        SwitchStatusChecking4 ss4 = new SwitchStatusChecking4();

        SwitchPosition4 sp4 = new SwitchPosition4();

        SwitchLock4 sl4 = new SwitchLock4();

        TrackLocking4 tl4 = new TrackLocking4();

        Train4.Top = 544;

        Train4.Left = T32G.X1;

        if (!rc4.routecheck())

        {                        MessageBox.Show("The Route not Available");

            break;

        }

        else if (!sit4.SignalStatus())

        {           MessageBox.Show("One of the Siganl is not Available");

            break;

        }

        else if (!ts4.trackstatuschecking4())

        {

            MessageBox.Show("One of the Track is not free wait");

            break;
```

```
    }
    else if (!ss4.switchstatuschecking4())
    {

        MessageBox.Show("One of the Switch is not free wait");
        break;
    }
    else
    {
        txtStatus.Text = "Checking Track";
        txtStatus.Text = "Checking Switch";



        sp4.switchposition4();
        txtStatus.Text = "Checking Switch Position";
        sl4.switchlock4();
        txtStatus.Text = "Locking Switch";
        tl4.tracklocking4();
        txtStatus.Text = "Locking Route";
        T32G.BorderColor = Color.Yellow;
        T38G.BorderColor = Color.Yellow;
        ru4.routeunavailable();

        R4 = true
    }



    break;
case "4":
    RouteCheck5 rc5 = new RouteCheck5();
    RouteUnavailable5 ru5 = new RouteUnavailable5();
```

```
SignalStatus5 sit5 = new SignalStatus5();
TrackStatusChecking5 ts5=new TrackStatusChecking5 ();
SwitchStatusChecking5 ss5=new SwitchStatusChecking5 ();
SwitchPosition5 sp5 = new SwitchPosition5();
SwitchLocking5 sl5 = new SwitchLocking5();
TrackLocking5 tl5 = new TrackLocking5();
if (!rc5.routecheck())
{
    MessageBox.Show("The Route not Available");
    break;
}

else  if (!sit5.SignalStatus())
{
    MessageBox.Show("One of the Siganl is not Available");
    break;



}
else  if (!ts5.trackstatuschecking5 ())
{
    MessageBox.Show("One of the Track is not free wait");
    break;
}
else if (!ss5.switchstatuschecking5 ())
{
    MessageBox.Show("One of the Switch is not free wait");
    break;

}
else
{
```

```
            txtStatus.Text = "Checking Track";

            txtStatus.Text = "Checking Switch";

            sp5.switchposition5();

            txtStatus.Text = "Checking Switch Position";

            sl5.switchlocking5();

            txtStatus.Text = "Locking Switch";

            tl5.tracklocking5();

            txtStatus.Text = "Locking Route";

            T39G.BorderColor = Color.Yellow;

            T35G.BorderColor = Color.Yellow;

            T28G.BorderColor = Color.Yellow;

            T19G.BorderColor = Color.Yellow;

            T9G.BorderColor = Color.Yellow;

            ru5.routeunavailable();

            R5 = true;                   }

        break;


    default :

        MessageBox.Show("You have to select Route Number");

        break;  } }
```