İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

RAILWAY SIGNALIZATION SCADA and SIMULATION PROJECT for INTERLOCKING SYSTEMS

M.Sc. Thesis by
Ömer Eren AVŞAROĞULLARI

Department : Control Engineering

Programme : Control and Automation Engineering

Thesis Supervisor: Assoc. Prof. Dr. Mehmet Turan SÖYLEMEZ

JUNE 2009

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

RAILWAY SIGNALIZATION SCADA and SIMULATION PROJECT for INTERLOCKING SYSTEMS

M.Sc. Thesis by
Ömer Eren AVŞAROĞULLARI
(504041115)

Date of submission : 04 May 2009
Date of defence examination: 03 June 2009

Supervisor (Chairman) : Assoc. Prof. Dr. Mehmet Turan SÖYLEMEZ (ITU)
Assoc. Prof. Dr. Salman KURTULAN (ITU)
Assis. Prof. Dr. Berk ÜSTÜNDAĞ (ITU)

JUNE 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ANKLAŞMAN SİSTEMLERİ için RAYLI SİSTEM SİNYALİZASYON SCADA TASARIMI ve SİMÜLASYONU PROJESİ

YÜKSEK LİSANS TEZİ
Ömer Eren AVŞAROĞULLARI
(504041115)

Tezin Enstitüye Verildiği Tarih : 04 Mayıs 2009
Tezin Savunulduğu Tarih : 03 Haziran 2009

Tez Danışmanı : Doç. Dr. M. Turan SÖYLEMEZ (İTÜ)
Doç. Dr. Salman KURTULAN (İTÜ)
Yrd. Doç. Dr. Berk ÜSTÜNDAĞ (İTÜ)

HAZİRAN 2009

FOREWORD

I would like to express my deep appreciation and thanks for my advisor.

June 2009

Ömer Eren Avşaroğulları
Control Engineering

TABLE OF CONTENTS

Page

ABBREVIATIONS ..... **ix**
LIST OF FIGURES ..... **xi**
SUMMARY ..... **xiii**
ÖZET ..... **xv**
1. INTRODUCTION **1**
1.1 What is Interlocking? 1
1.2 Importance of Simulation 2
1.3 Scada 2
2. TECHNOLOGIES USED in SIMULATION **5**
2.1 PLC Tools: 5
2.2 Simulation and PLC Communication Bridge : 5
2.3 Programming Language : 5
2.4 Database: 5
2.5 IDE: 5
2.6 Logging Tool : 5
2.7 Design Tools : 6
2.8 Other Technologies : 6
3. DATABASE MODEL and UML DIAGRAMS of RASIGSIM-SCADA **7**
3.1 Database Model 7
3.1.1 Relational DB Schema 7
3.1.2 Tables 8
3.2 UML Diagrams 14
3.2.1 General 14
3.2.2 Data Objects 14
3.2.3 Jpns 16
3.2.4 Opc 17
3.2.5 Sql 19
3.2.6 Util 19
3.2.7 Menu 20
4. INTERFACES **25**
4.1 Petri Nets 25
4.2 Transformation from Railway Signalization Layout to Petri Nets 25
4.3 Opc 26
4.4 JEasyOpc Java-OPC Api 27
4.5 STL 28
5. ILLUSTRATION OF THE APPLICATION **29**
5.1 Simulation Demo 29
5.2 Transition from Program Layout to Petri Nets 32
6. CONCLUSION AND RECOMMENDATIONS **35**
REFERENCES **37**
APPENDICES **39**

File Menu - New Project Menu Item 41
File Menu - Open Project Menu Item 42
File Menu – Exit Menu Item 42
Petri Nets Menu – Convert To Petri Nets Menu Item 43
Simulation Menu – Simulate Menu Item 43
Simulation Menu - Traffic Control Center Menu Item 43
Insert Menu – Line Menu Item 44
Insert Menu – Switch Menu Item 44
Insert Menu – Signal Box Menu Item 45
Insert Menu – Track Circuit Menu Item 46
Insert Menu – Station Menu Item 47
Insert Menu – Train Menu Item 48
Insert Menu – Connection Menu Item 49
Help Menu – Help Menu Item 49
Help Menu – PLC Inputs Help Menu Item 50
CV 51

ABBREVIATIONS

PLC	**:** Programmable Logic Controller
SCADA	**:** Supervisory Control and Data Acquisiton
GUI	**:** Graphical User Interface
SQL	**:** Structured Query Language
OPC	**:** OLE for Process Control
JPNS	**:** Java Petri Net Simulator
API	**:** Application Program Interface
STL	**:** Statement Language
TCC	: Traffic Control Center
SB	: Signal Box
TC	: Track Circuit

LIST OF FIGURES

Page

Figure 3.1 : Relational Database Schema. 7
Figure 3.2 : Table Project Informations. 8
Figure 3.3 : Table Line Configurations.....8
Figure 3.4 : Table Station Configurations. 9
Figure 3.5 : Table Signal Box Configurations. 9
Figure 3.6 : Table Switch Configurations. 10
Figure 3.7 : Table Track Circuit Configurations. 11
Figure 3.8 : Table Train Configurations. 12
Figure 3.9 : Table Simulation Step Train Position Values. 12
Figure 3.10 : Table SB TC Associations. 13
Figure 3.11 : General UML Diagram of the project. 14
Figure 3.12 : Data Object UML Diagram of the project.....16
Figure 3.13 : Jnps Interface UML Diagram of the project.....17
Figure 3.14 : OPC(JEasyOPC) API UML Diagram of the project. 18
Figure 3.15 : Database Layer UML Diagram of the project. 19
Figure 3.16 : MyUtil API UML Diagram of the project.....20
Figure 3.17 : Screen UML Diagram I of the project. 22
Figure 3.18 : Screen UML Diagram II of the project. 23
Figure 4.1 : OPC Bridge between application and PLC 27
Figure 5.1 : New Project and Open Project Screens.....29
Figure 5.2 : Configuration Parameters Menu Item.....29
Figure 5.3 : Train Parameters Menu Item 30
Figure 5.4 : PLC Inputs Help Sample. 30
Figure 5.5 : Start Connection Menu Item 31
Figure 5.6 : Simulation Panel 31
Figure 5.7 : Traffic Control Center(TCC) Screen 31
Figure 5.8 : Convert To Petri Nets Menu Item 32
Figure 5.9 : Transformation of interlocking system model to basic Petri Nets. 33
Figure A.1.1 : New Project Menu Item.....41
Figure A.1.2 : New Project Screen 41
Figure A.1.3 : Open Project Menu Item 42
Figure A.1.4 : Open Project Screen 42
Figure A.1.5 : Exit Menu Item 42
Figure A.1.6 : Simulate Menu Item 43
Figure A.1.7 : Traffic Control Center Menu Item 43
Figure A.1.8 : Line Parameter Screen.....44
Figure A.1.9 : Switch Parameter Menu Item 44
Figure A.1.10 : Switch Parameter Screen 45
Figure A.1.11 : Signal Box Parameter Menu Item 45
Figure A.1.12 : Signal Box Parameter Screen 46

Figure A.1.13 : Track Circuit Parameter Menu Item 46
Figure A.1.14 : Track Circuit Parameter Screen..... 47
Figure A.1.15 : Station Parameter Menu Item..... 47
Figure A.1.16 : Station Parameter Screen 48
Figure A.1.17 : Train Parameter Menu Item 48
Figure A.1.18 : Train Parameter Screen..... 49
Figure A.1.19 : Start Connection Menu Item 49
Figure A.1.20 : Help Menu Item..... 49
Figure A.1.21 : PLC Inputs Help Menu Item 50

RAILWAY SIGNALIZATION SCADA and SIMULATION PROJECT for INTERLOCKING SYSTEMS

SUMMARY

In this thesis Scada and Simulation applications for Railway Interlocking Systems have been developed. In particular, the following properties are taken into account :

- Testing interlocking software of interlocking systems in real-time.
- The Scada software that sends the data to the interlocking side to be tested and receives the data back. Also the program is able to communicate with the external devices such as PLCs.
- A feature that provides to be simulated the signalization model of railway systems with lines, switches, track circuits, signal boxes, trains and station components.
- A feature that provides transformation from interlocking system model to basic Petri Nets.
- A feature that provides infrastructure for transformation from basic Petri Nets to STL PLC program by an interface.

Initially, interlocking, simulation and scada systems are explained in detail and program menus are presented graphically. In the second section, the technologies used to develop the project, the data model and the UML diagrams of the general architecture of the system are specified. The next section explains Petri Nets, OPC and STL and the processes of convertions between interlocking model and basic Petri Nets. Finally, a demo section is presented to explain the program step by.In addition to these; database installation, the user manual of the software and the development documentation(javadoc) is added to the software CD.

ANKLAŞMAN SİSTEMLERİ için RAYLI SİSTEM SİNYALİZASYON SCADA TASARIMI ve SİMÜLASYONU PROJESİ

ÖZET

Bu Yüksek Lisans Tez Projesinde aşağıda belirtilen uygulamalar hedeflenerek, geliştirilmeleri sağlanmıştır :

- Gerçek zamanlı olarak anklaşman sistemlerinin anklaşman yazılımlarının test edilmesi,
- Verilerin anklaşman tarafına gönderilerek test edildikten sonra tekrar toplanarak görüntülenmesini ve harici cihazlarla haberleşmeyi sağlayan arayüzü içeren Scada programı,
- Raylı sistemler için oluşturulan sinyalizasyon modelinin, tren hatları, makaslar, ray devreleri, sinyalizasyon lambaları, trenler ve istasyon elemanları ile simüle edilebilme özelliği,
- Operator tarafından girilen anklaşman sistem modelinin, basit Petri Ağına dönüştürülebilmesi özelliği,
- Basit Petri Ağına dönüştürülen anklaşman sistem modelinin STL PLC yazılımını geliştirecek arayüz programı için altyapı oluşturulması.

Anklaşman, Simülasyon ve Scada Sistemleri açıklandıktan sonra program menuleri grafiklerle beraber ayrıntılı olarak açıklanmıştır. Projenin geliştirilmesinde kullanılan teknolojiler belirtildikten sonra veritabanında oluşturulan tablolar, ilişkisel veritabanı şeması ve programın genel mimarisini belirten UML diagramları ayrıntılı olarak ifade edilmiştir. Petri Ağları, OPC ve STL Teknolojileri açıklandıktan sonra operator tarafından girilen anklaşman sistem modelinin, basit Petri Ağına nasıl dönüştürüleceği açıklanmıştır. Ilerliyen zamanda, projenin daha da geliştirilebilmesi amacıyla; basit Petri Ağından STL kodunun üretilebilmesi için gerekli altyapı da hazırlanmıştır. Demo kısmı adım adım açıklandıktan sonrada proje sonuçlandırılarak kısaltmalar ve referanslar belirtilmiştir. Ayrıca Veritabanı kurulumu ile programın kullanımını içeren yardım dokumanı ve Programın ilerliyen zamanda geliştirilebilmesi amacıyla yazılım içeriğini açıklayan dokuman (javadoc) program CD' sine eklenmiştir.

1. INTRODUCTION

1.1 What is Interlocking?

Interlocking *is defined as the, or an, "arrangement of signals and switches such that they are constrained to be operable only in a safe order."* ***An interlocking*** *is a place where tracks join (****switches****), and denotes the switches, the surrounding signals, and the control machinery which connects them and enables their operation (by a* ***tower operator*** *(formerly* ***towerman****), or sometimes automatically) and ensures that operation is safe, regardless of the operator's actions.*[1]

Some of the fundamental principles of interlocking include:

- When there are inconsistent train movements, signals are not permitted to take place at the same time
- Before a signal may allow train movements to enter that route, switches and other appliances in the route must be properly 'set' (in correct position)
- Once a route is set and a train is given a signal to proceed over that route, all switches and other movable appliances in the route are locked in position until either
    - the train passes out of the portion of the route affected, or
    - the signal to proceed is withdrawn and sufficient time has passed to ensure that a train approaching that signal has had opportunity to come to a stop before passing the signal.

Interlocking types are that :

- Mechanical interlocking
- Electro-mechanical interlocking
- Relay interlocking
- Electronic interlocking

1.2 Importance of Simulation

System design, analysis and performance have important cost for the purposes of time, effort and economy. Simulation is used in order to decrease this cost Simulation does not solve problems on its own but defines the problem and evaluates alternative solutions. Before a new system is developed, simulation helps to demonstrate a lot of ambushs. Simulation feature which collects a lot of parameters of the system in a single model helps solving high-complexity problems.

Simulation can be used in the following situations :

- When results of specified decision are guessed
- When causes of observed decision are specified
- Before being realized an investment, when problems are specified
- When impacts of changes are seeked
- When all system parameters are found
- When system efficiency is calculated
- When all probabilities of system are tested.

Simulation is also very important for the purposes of composing railway signalization systems, which require huge cost. Before system is developed, possible problems can be determined and solved by the help of simulation. This project aims for this purpose.

1.3 Scada

The term SCADA stands for Supervisory Control and Data Acquistion. Scada Systems are formed by computers, communication devices, sensors and other automation devices. Generally, it can be classified as energy scada(electric-water-natural gas) and process scada(plant automation).

Scada systems, which are used for the collecting system data and controlling processes, composes a background for production control and production performance of the plant. This background is supported by Materials Requirement Planning(MRP) and Enterprise Resource Planning(ERP) Applications. The most importantgoal of SCADA systems is composing robust and safe systems with

minimum cost and maximum efficiency. Towards this goal SCADA provides detailed data to operators in real time.

More detailed information on SCADA can be received from [2].

2. TECHNOLOGIES USED in SIMULATION

The project is developed using the following technologies:

2.1 PLC Tools:

Siemens WinLC S7 300 SoftPLC : Computer program that tries to provide the programming environment and performance of an expensive PLC on a lower cost PC platform.

SIMATIC NET OPC Server : It provides communication between application and PLC.

SIMATIC IDE : Simatic Integrated Development Environment

2.2 Simulation and PLC Communication Bridge :

OLE for Process Control (OPC) : A standart for industrial communications.

2.3 Programming Language :

Java-Swing : The programming language used to develop the application.

2.4 Database:

MySQL Server : Database server.

2.5 IDE:

Eclipse : Integrated Development Environment used to develop the application.

2.6 Logging Tool :

Log4j : This logging tool is used to handle error and informations at run time.

2.7 Design Tools :

UML : Unified Modelling Language is a high level programing language that is used to design the application.

2.8 Other Technologies :

JPNS : An open source editor for petri nets

JEasyOPC: Open Source JAVA – OPC Bridge API

3. DATABASE MODEL and UML DIAGRAMS of RASIGSIM-SCADA

3.1 Database Model

3.1.1 Relational DB Schema

Relational Database Schema is seen via Figure 3.1 :

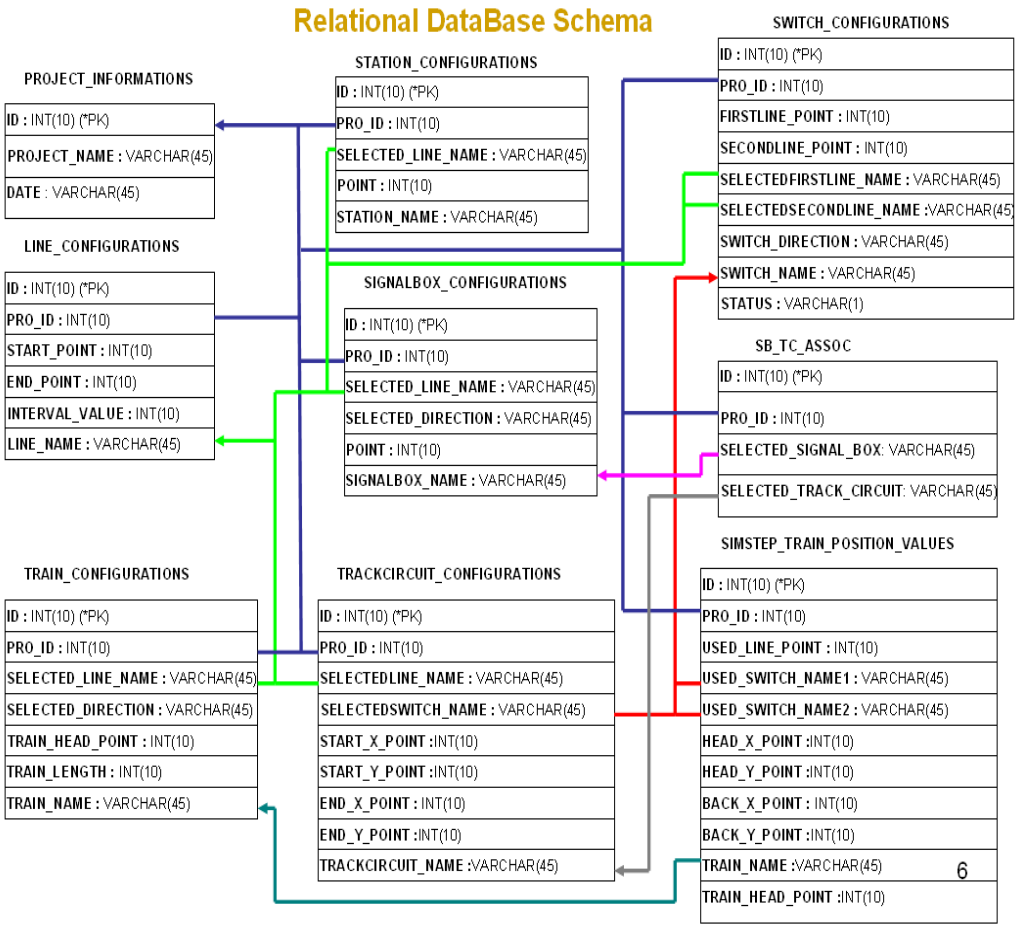

Figure 3.1 : Relational Database Schema.

3.1.2 Tables

3.1.2.1 Table PROJECT_INFORMATIONS

General information on the project is saved on the table PROJECT_INFORMATIONS.

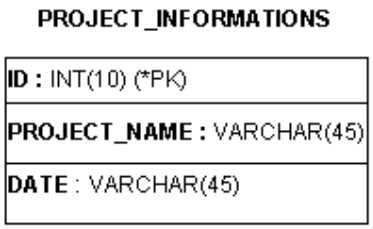

Figure 3.2 : Table Project Informations.

ID : Project ID.

PROJECT NAME : Field which holds Project Name.

DATE : Creation date of Project

3.1.2.2 Table LINE_CONFIGURATIONS

Information on lines is held with table LINE_INFORMATIONS.

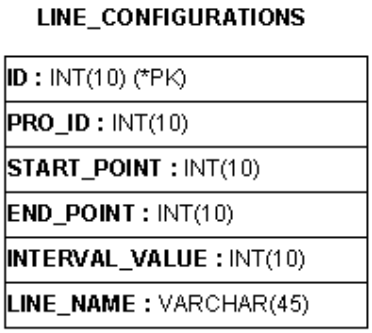

Figure 3.3 : Table Line Configurations.

ID : Line Record ID.

PRO_ID : Project ID.

START_POINT : Starting point of line.

END_POINT : End point of line.

INTERVAL_VALUE : Field which points ordinate of line according to origin.

LINE NAME : Line Name

3.1.2.3 Table STATION_CONFIGURATIONS

Information on stations is held in table STATION _INFORMATIONS.

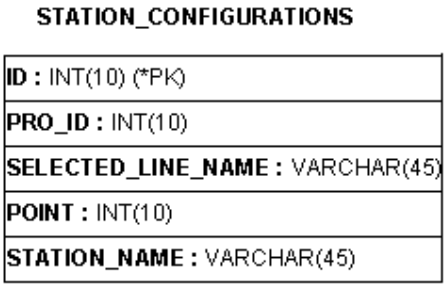

Figure 3.4 : Table Station Configurations.

ID : Station Record ID.

PRO_ID : Project ID.

SELECTED_LINE_NAME : Name of the line on which the station is positioned.

POINT : Station starting point on the line

STATION_NAME : Station Name

3.1.2.4 Table SIGNALBOX_CONFIGURATIONS

Information on signal boxes is held in table SIGNALBOX _INFORMATIONS.

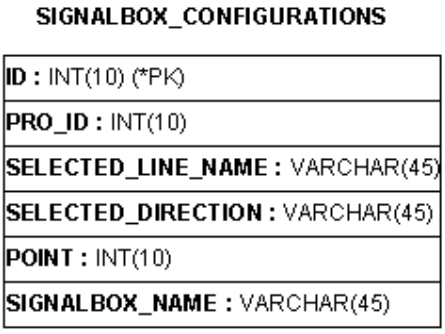

Figure 3.5 : Table Signal Box Configurations.

ID : Signal Box Record ID.

PRO_ID : Project ID.

SELECTED_LINE_NAME : Name of the line on which the signal box is positioned.

SELECTED_DIRECTION : Direction of the signal box

POINT : Position of the signal box

SIGNALBOX_NAME Name of the Signal Box.

3.1.2.5 Table SWITCH_CONFIGURATIONS

Information on switches is held in table SIGNALBOX _INFORMATIONS.

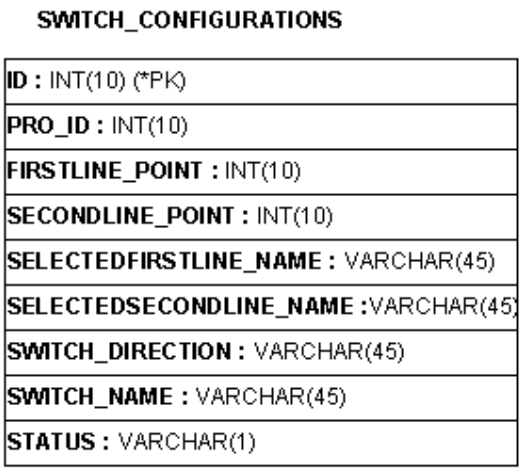

Figure 3.6 : Table Switch Configurations.

ID : Switch Record ID.

PRO_ID : Project ID.

FIRSTLINE_POINT : Position on the first line

SECONDLINE_POINT : Position on the second line

SELECTEDFIRSTLINE_NAME : Name of the first line

SELECTEDSECONDLINE_NAME : Name of the second line

SWITCH_DIRECTION : Direction of the switch

SWITCH _NAME : Switch Name

STATUS : Status of switch (?????)

3.1.2.6 Table TRACKCIRCUIT_CONFIGURATIONS

Information on Track Circuits is held in table TRACKCIRCUIT_CONFIGURATIONS.

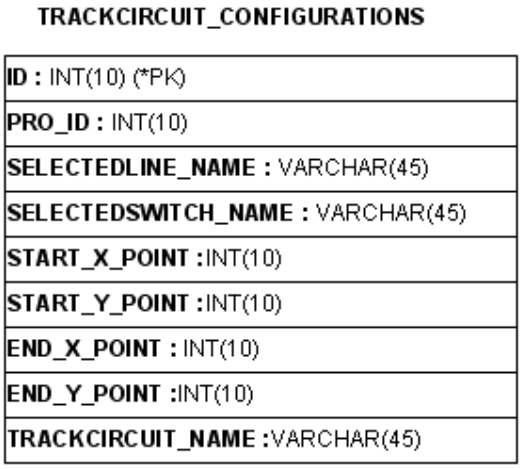

Figure 3.7 : Table Track Circuit Configurations.

ID : Track Circuit Record ID.

PRO_ID : Project ID.

SELECTEDLINE_NAME : Name of the selected line.

SELECTEDSWITCH_NAME : Name of the selected switch.

START_X_POINT : Starting point' s apsis of Track Circuit

START_Y_POINT : Starting point' s ordinate of Track Circuit

END_X_POINT : End point' s apsis of Track Circuit

END_Y_POINT : End point' s ordinate of Track Circuit

TRACKCIRCUIT _NAME : Track Circuit Name

3.1.2.7 Table TRAIN_CONFIGURATIONS

Information on trains is held in table TRAIN_CONFIGURATIONS.

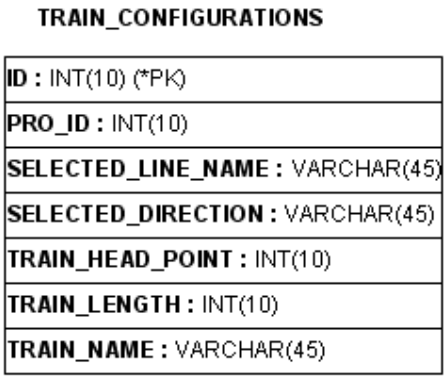

Figure 3.8 : Table Train Configurations.

ID : Train Record ID.

PRO_ID : Project ID.

SELECTED_LINE_NAME : Name of selected line

SELECTED_DIRECTION : Name of selected direction

TRAIN_HEAD_POINT : Starting point' s apsis of Train' s head side

TRAIN_LENGTH: Length of the Train

TRAIN _NAME : Train Name

3.1.2.8 Table SIMSTEP_TRAIN_POSITION_VALUES

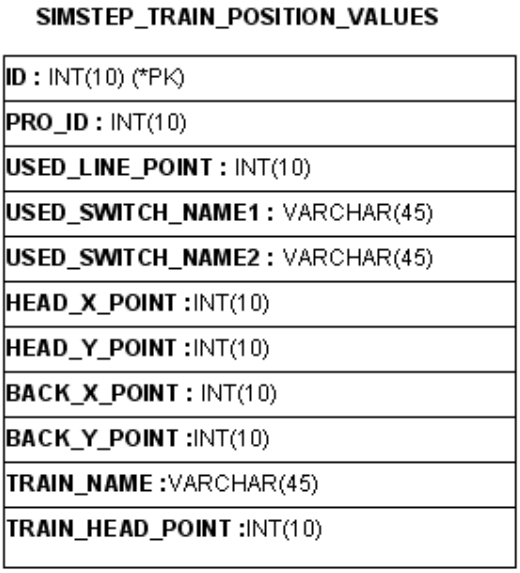

Figure 3.9 : Table Simulation Step Train Position Values.

ID : Simulation step Record ID.

PRO_ID : Project ID.

USED_LINE_NAME : Name of line used

USED_SWITCH_NAME1 : Name of switch used.

USED_SWITCH_NAME2 : Name of switch used.

HEAD_X_POINT : Head point' s apsis of Train

HEAD_Y_POINT : Head point' s ordinate of Train

BACK_X_POINT : Back point' s apsis of Train

BACK_Y_POINT : Back point' s ordinate of Train

TRAIN_NAME : Train Name

TRAIN_HEAD_POINT : Starting point of Train

3.1.2.9 Table SB_TC_ASSOC

Relation between Signal Box and Track Circuit is held with SB_TC_ASSOC.

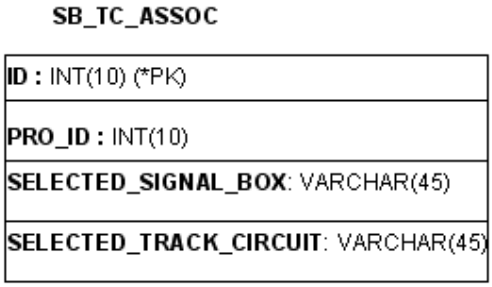

Figure 3.10 : Table SB TC Associations.

ID : Record ID.
PRO_ID : Project ID.
SELECTED_SIGNAL_BOX : Name of selected signal box.
SELECTED_TRACK_CIRCUIT : Name of selected track circuit.

3.2 UML Diagrams

3.2.1 General

Detailed java documentation of the program is available and can be reached via Program CD. A scetch of a class (UML) diagram of the program is given in Figure 3.11.

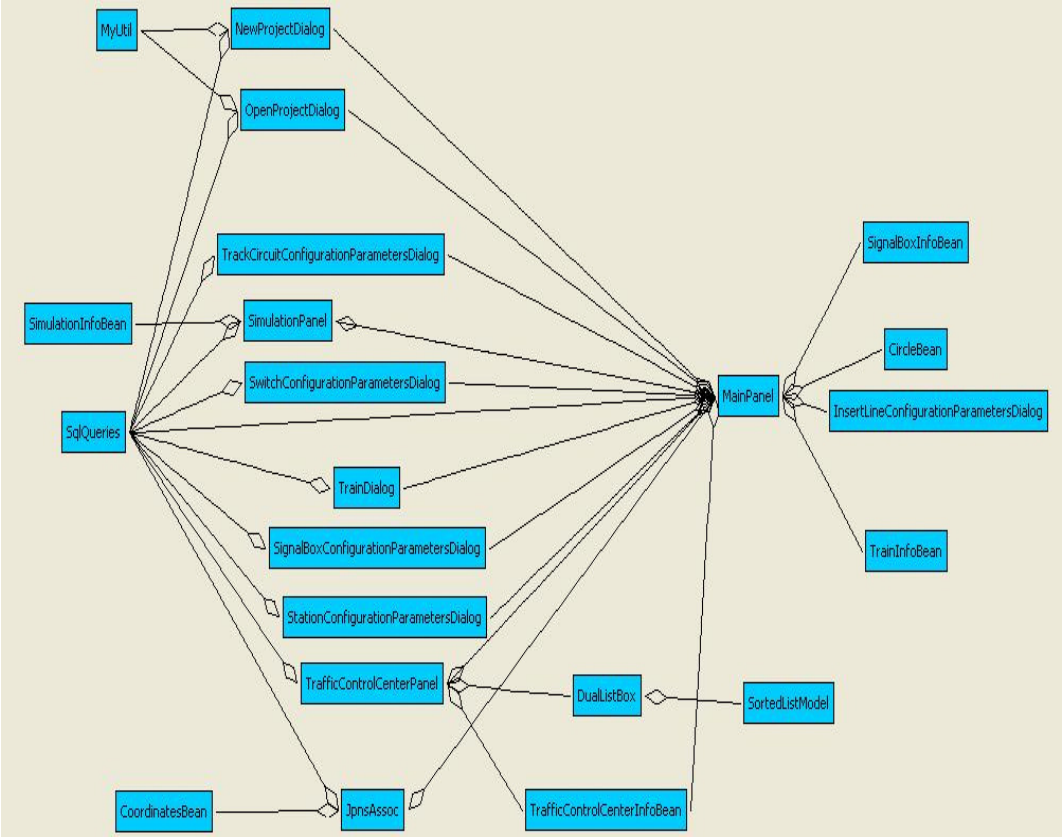

Figure 3.11 : General UML Diagram of the project.

3.2.2 Data Objects

The most important data objects that are used in the program are summarized in Figure 3.12. These objects are

CircleBean : Data Object which holds values of Circle Component.

CoordinatesBean : Data Object which holds values of Track Circuit Component.

LineInfoBean : Data Object which holds values of Line Component.

ProjectIdBean : Data Object which holds values of Project Object.

SignalBoxInfoBean : Data Object which holds values of Signal Box Component.

SimulationInfoBean : Data Object which holds values of Simulation Component.

StationInfoBean : Data Object which holds values of Station Box Component.

SwitchInfoBean : Data Object which holds values of Switch Component.

TrackCircuitInfoBean : Data Object which holds values of Track Circuit Component.

TrafficControlCenterInfoBean : Data Object which holds values of TCC Component.

TrainInfoBean : Data Object which holds values of Train Component.

TrainLastPositionBean : Data Object which holds the last position values of Train Component

Figure 3.12 : Data Object UML Diagram of the project.

3.2.3 Jpns

JpnsAssoc : This class is used as an interface between Program and JNPS Petri Nets Editor. A UML diagram of the class is given in Figure 3.13.

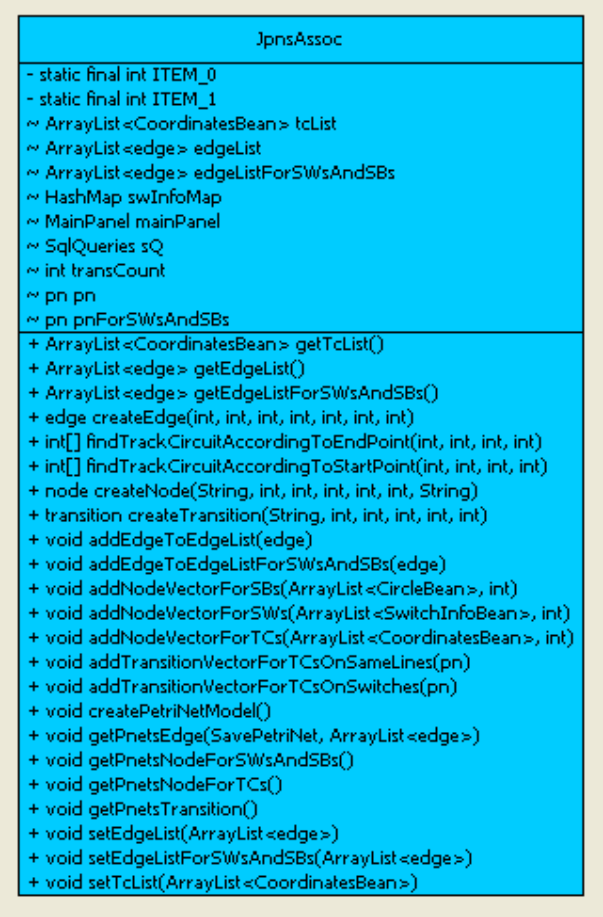

Figure 3.13 : Jnps Interface UML Diagram of the project.

3.2.4 Opc

SynchSimulationAndPlc : This class is used as an interface between the program and PLC.

JOpc : This class implements OPCDA standard (2.0, 3.0) and is extended from Class JCustomOpc.

JCustomOpc : This class is base class for OPC Clients. It contains native methods.

A UML diagram of the OPC classes is given in Figure 3.14.

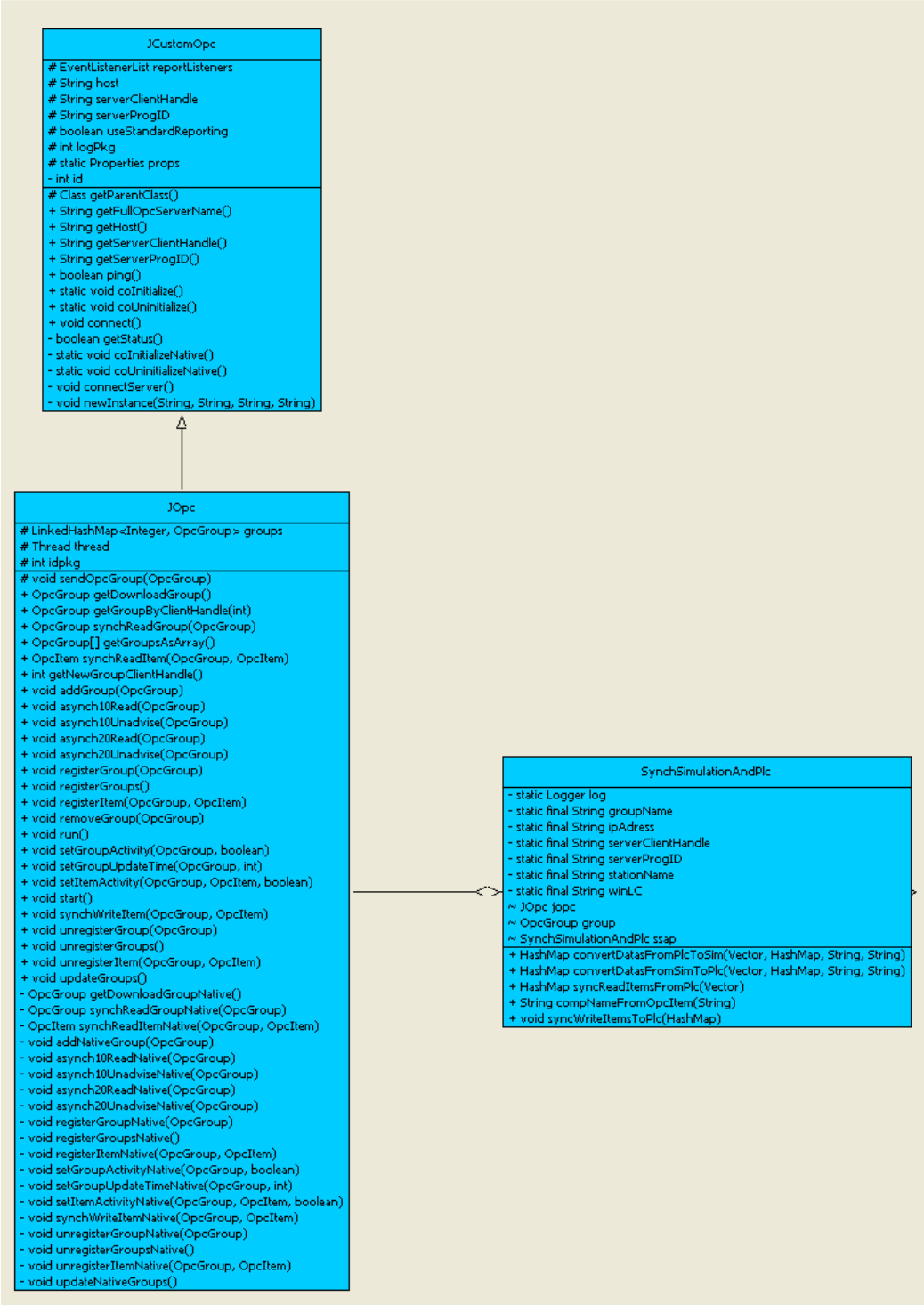

Figure 3.14 : OPC(JEasyOPC) API UML Diagram of the project.

3.2.5 Sql

DBConnection : Connection between the program and the database is obtained via this class.

SqlQueries : SQL Methods are contained by this class.

A UML diagram of the database layer classes is given in Figure 3.15.

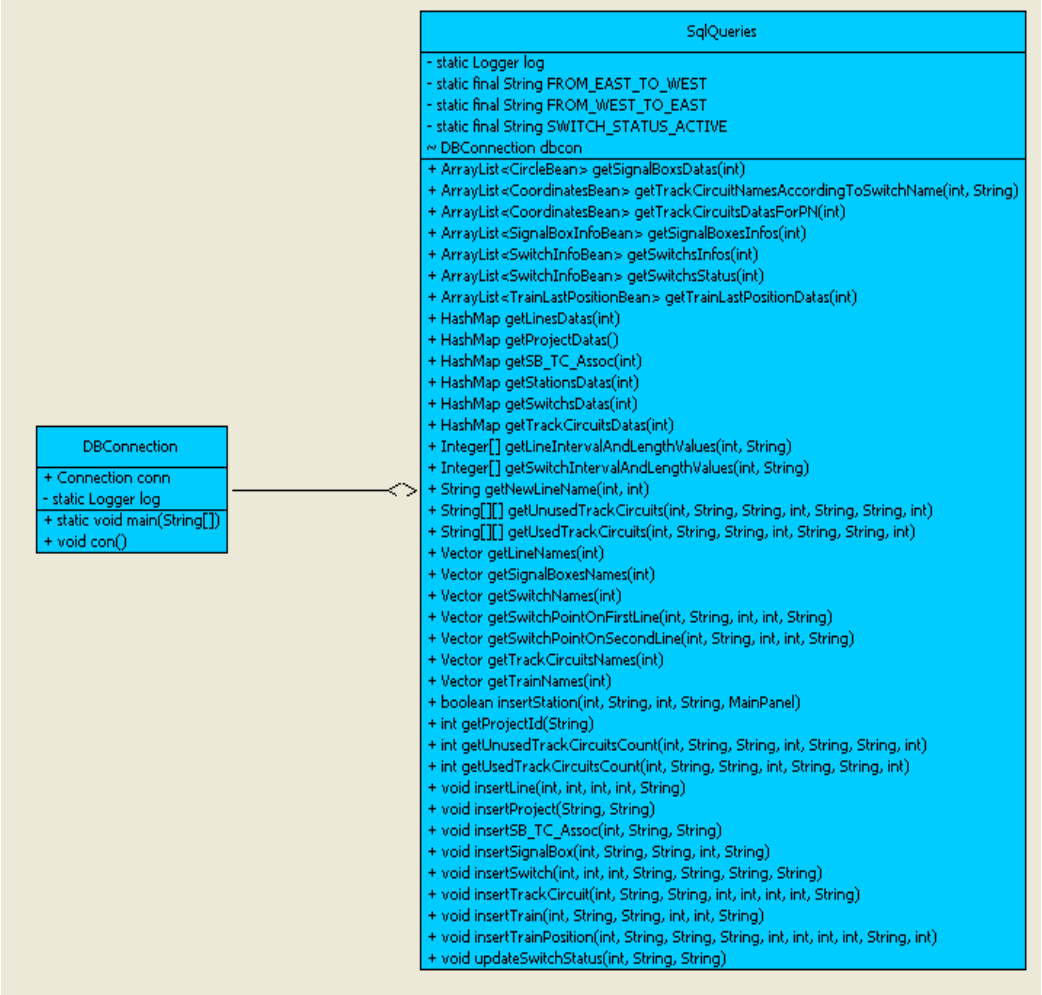

Figure 3.15 : Database Layer UML Diagram of the project.

3.2.6 Util

MyUtil : Frequently used methods are contained by this class.

DualListBox : The class of DualListBox object.

A UML diagram of the MyUtil API classes is given in Figure 3.17.

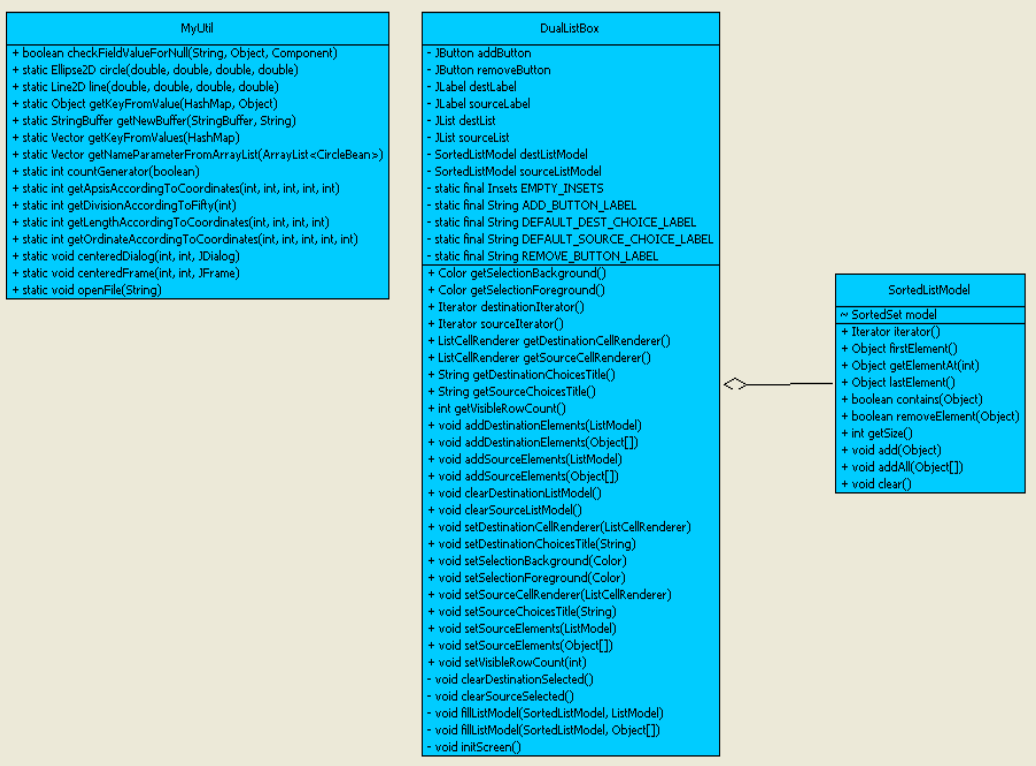

Figure 3.16 : MyUtil API UML Diagram of the project.

3.2.7 Menu

Two class diagrams related to user interface part of the program are given in Figure 3.17 and Figure 3.18. Some of the classes shown in these diagrams are

InsertLineConfigurationParametersDialog:The dialog of Line Component

MainPanel : Main Panel of the project

NewProjectDialog : The dialog of New Project. Menu Item

OpenProjectDialog : The dialog of Open Project. Menu Item

SignalBoxConfigurationParametersDialog : The dialog of Signal Box Component.

SimulationPanel : The panel of Simulation Menu Item.

StationConfigurationParametersDialog : The dialog of Station Component .

SwitchConfigurationParametersDialog : The dialog of Switch Component.

TrackCircuitConfigurationParametersDialog : The dialog of Track Circuit Component.

TrafficControlCenterPanel : The panel of Traffic Control Center Menu Item.

TrainDialog : The dialog of Train Component.

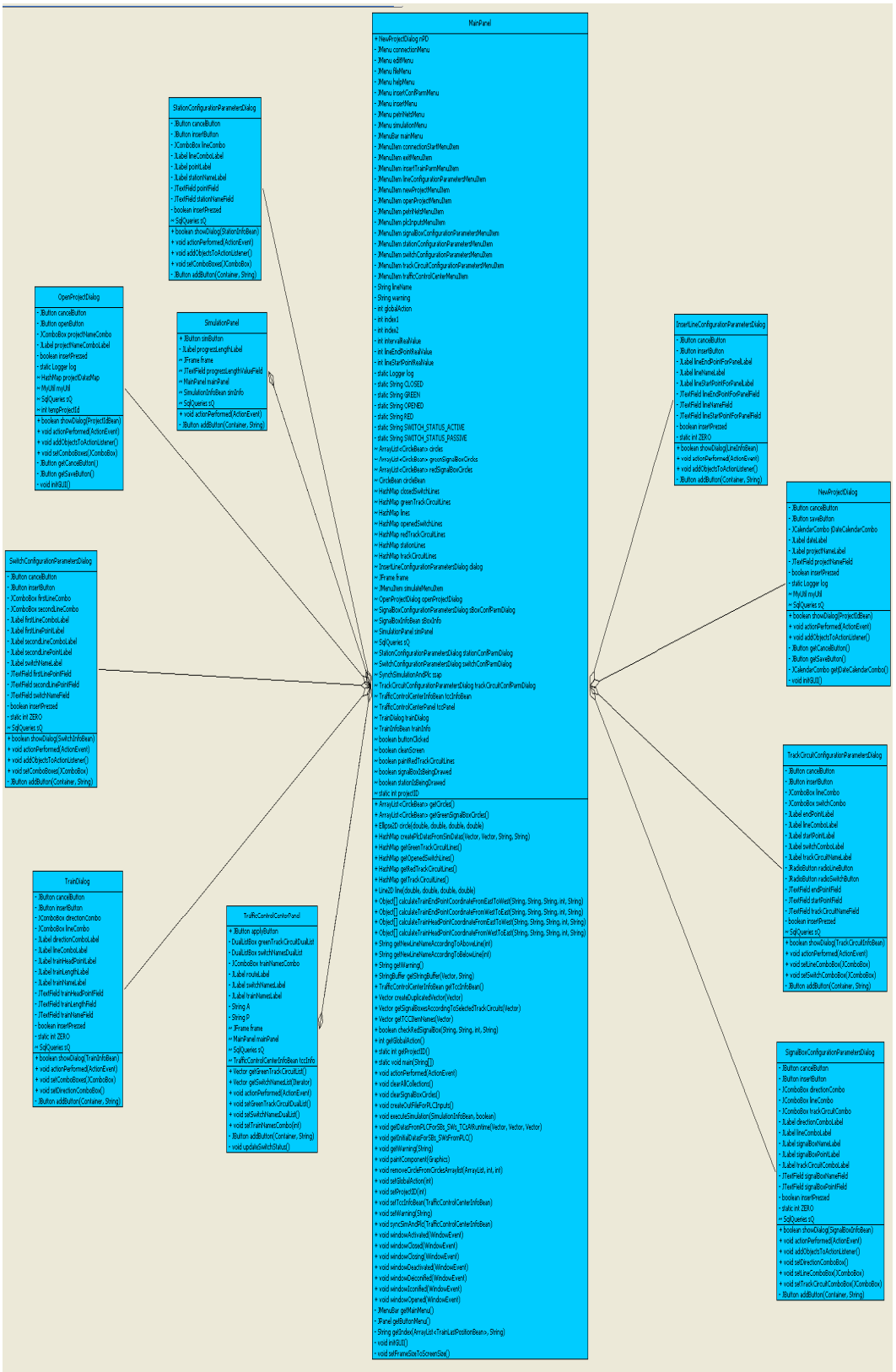

Figure 3.17 : Screen UML Diagram I of the project.

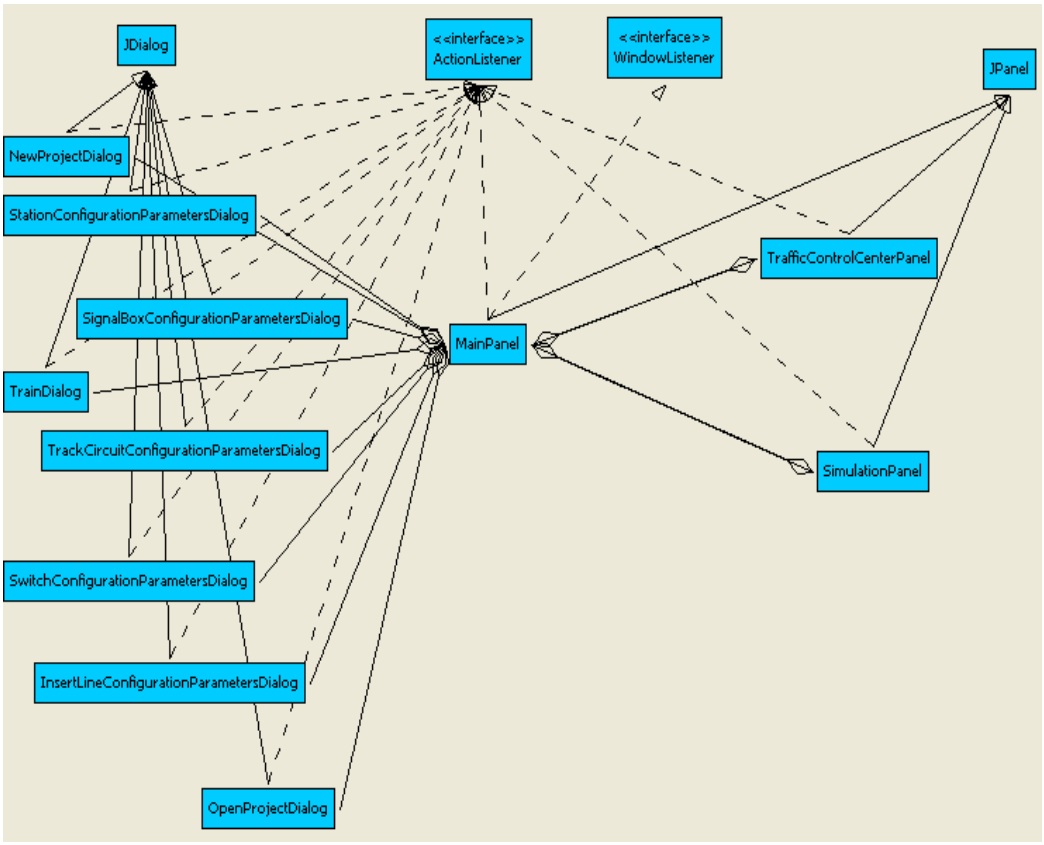

Figure 3.18 : Screen UML Diagram II of the project.

4. INTERFACES

4.1 Petri Nets

Petri Nets is used in order to model and analyze discrete-event systems as a graphical and mathematical instrument. Graphically, Petri Nets helps the designer to track the behavior of the system easily. Mathematically, it is quite easily possible to obtain-the behavior of a system as a set of linear equations and other mathematical models. It can be used for all systems that can be described with a flow-chart. As a result it is possible to use Petri Nets for the analysis and design of railway signaling systems as for other countless systems.

Application areas of Petri Nets include software design, workflow management, data analysis, concurrent programming, reliability engineering, diagnosis and discrete process control. In addition, Main Petri Net Types are below :

- State Machine(SM)
- Marked Graph(MG)
- Free Choice(FC)
- Extended free choice(EFC)
- Asymmetric Choice(AC)
- Multiple Asymmetric Choice(MAC)
- Petri Net(PN)

More detailed informations can be received from [3].

4.2 Transformation from Railway Signalization Layout to Petri Nets

It is important to be produced interlocking PLC Software automatically for the purposes of railway signalization but this is a big and complex problem. If Petri Model of Interlocking System can be developed as a whole or limited, that converting from Petri Nets to STL code can be developed as a whole or limited.

Railway Signalization Simulation Project recommends a limited solution because that developing general rule is a rather difficult problem due to a lot of private rules. Petri Nets of interlocking system is allowed to two group as named Basic Petri Nets and Automation Petri Nets. Basic Petri Nets is developed via basic rules and Automation Petri Nets is developed via private rules specified special behaviors of interlocking system. The limited solution contains only basic Petri Nets.

4.3 Opc

OPC(OLE for Process Control) which is supported by Microsoft OLE / COM is open connectivity via open standards. They fill a need in automation like printer drivers did for Windows. It describes a standart interface for connection between different hardware components and HMI/SCADA applications. There are currently seven standards specifications completed or in development.

Current and emerging OPC Specifications include:

OPC Data Access : *The originals! Used to move real-time data from PLCs, DCSs, and other control devices to HMIs and other display clients. The Data Access 3 specification is now a Release Candidate. It leverages earlier versions while improving the browsing capabilities and incorporating XML-DA Schema.*

OPC Data eXchange : *This specification takes us from client/server to server-to-server with communication across Ethernet fieldbus networks. This provides multi-vendor interoperability! And adds remote configuration, diagnostic and monitoring/management services.*

OPC Historical Data Access : *Where OPC Data Access provides access to real-time, continually changing data, OPC Historical Data Access provides access to data already stored. From a simple serial data logging system to a complex SCADA system, historical archives can be retrieved in a uniform manner.*

OPC Security : *All the OPC servers provide information that is valuable to the enterprise and if improperly updated, could have significant consequences to plant*

processes. OPC Security specifies how to control client access to these servers in order to protect this sensitive information and to guard against unauthorized modification of process parameters.

OPC XML-DA : *Provides flexible, consistent rules and formats for exposing plant floor data using XML, leveraging the work done by Microsoft and others on SOAP and Web Services.*[4]

4.4 JEasyOpc Java-OPC Api

The JEasyOpc project provides an OPC (OLE for process control) communication.

The first OPC standard specification resulted from the collaboration of a number of leading worldwide automation suppliers working in cooperation with Microsoft. Originally based on Microsoft's OLE COM and DCOM technologies, the specification defined a standard set of objects, interfaces and methods for use in process control and manufacturing automation applications to facilitate interoperability.

The COM/DCOM technologies provided the framework for software products to be developed. There are now hundreds of OPC Data Access servers and clients. The JEasyOpc project wants to provide this technology for everyone with easy way.

The JEasyOpc project provides OPC for Java technology. The project is based on JDK 1.5 (or higher) and a Delphi (2005) native code (JCustomOpc.dll library). The application uses 32-bit MS Windows (95/98/NT/2000/XP). The operating system Linux isn't supported.

JEasyOPC API' s usage in the project is seen via Figure 4.1.

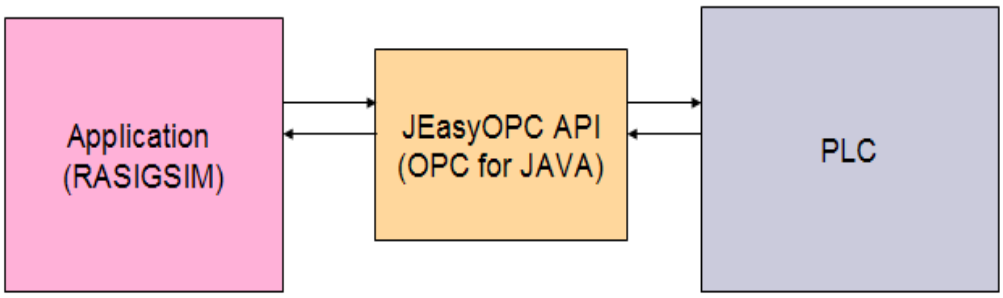

Figure 4.1 : OPC Bridge between application and PLC

4.5 STL

Statement list is a programming language using mnemonic abbreviations of Boolean logic operations. Boolean operations work on combination of variables that are true or false. A statement is an instruction or directive for the PLC.

Statement List Operations

- Load (LD) instruction.
- And (A) instruction.
- Or (O) instruction.
- Output (=) instruction.

5. ILLUSTRATION OF THE APPLICATION

Simulation demo and transition from program layout to petri nets are explained with this chapter.

5.1 Simulation Demo

The following steps should be executed in order to simulate :

1) A new project is created via *New Project* item under *Project Menu* or a project can be selected via *Open Project* item under *Project Menu. The dialogs specified at Figure 5.1 are seen after selecting menu item.*

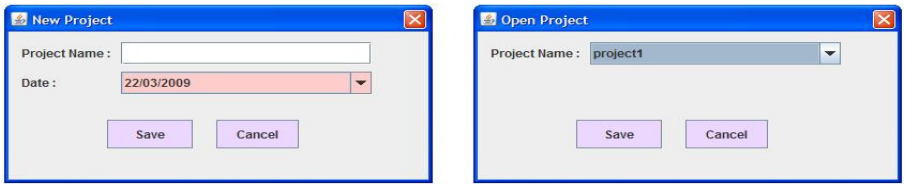

Figure 5.1 : New Project and Open Project Screens

2) Components (Line, Switch, Signal Box, Track Circuit and Station) are added via *Configuration Parameters* under *Insert Menu as shown Figure 5.2*.

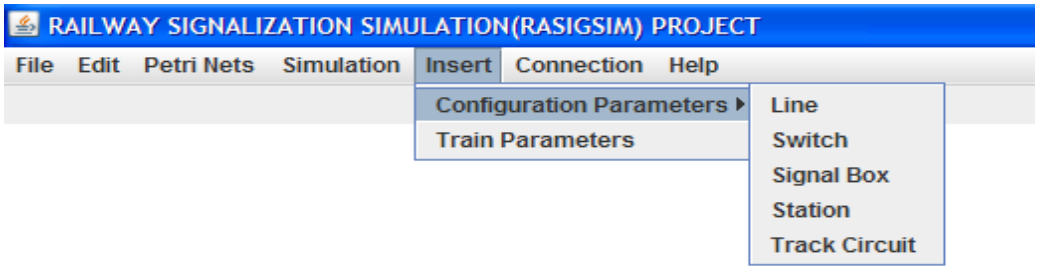

Figure 5.2 : Configuration Parameters Menu Item

3) Trains are added via *Train Parameters* under *Insert Menu as shown Figure 5.3*..

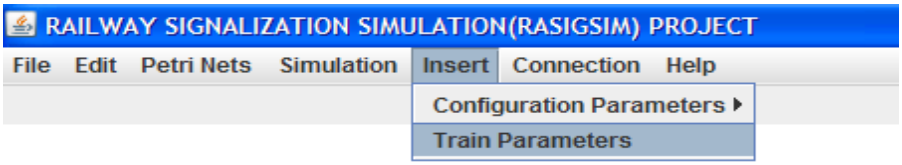

Figure 5.3 : Train Parameters Menu Item

4) Interlocking PLC Software should be written according to layout and used components. It is important that PLC inputs' names should be the same as PLC out file, which is opened via *PLC Inputs Item* under *Help Menu as shown Figure 5.4*..

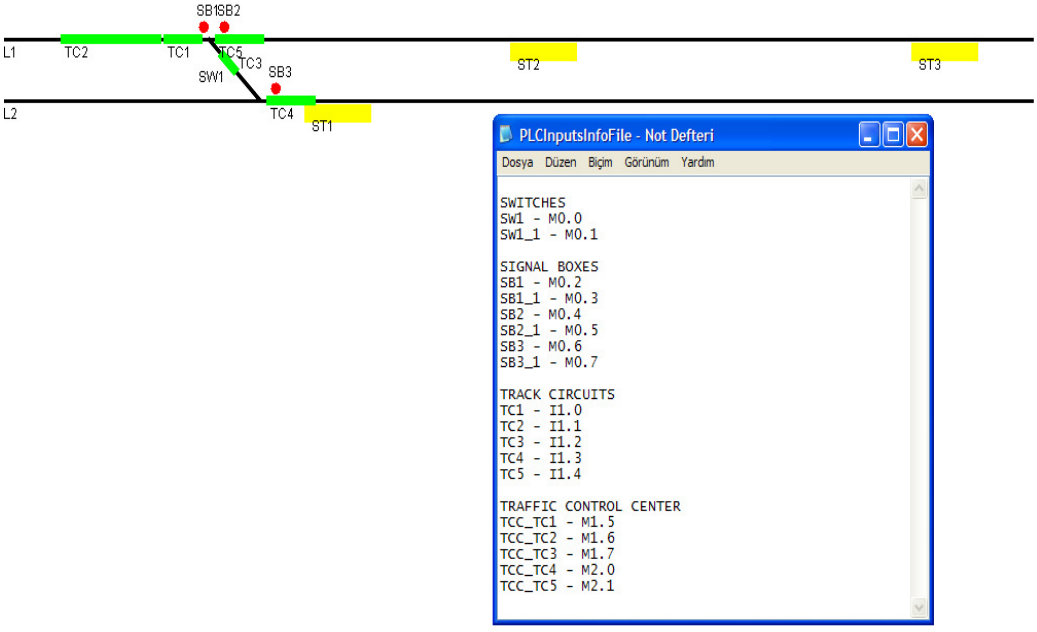

Figure 5.4 : PLC Inputs Help Sample.

5) Interlocking PLC software is uploaded to PLC and PLC is set to "**Run**" status.

6) Initial values are taken from PLC via *Start Connection* item under *Connection Menu as shown Figure 5.5*.. Thus, simulation and PLC has been synchronized.

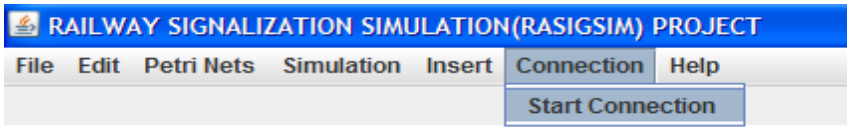

Figure 5.5 : Start Connection Menu Item

7) Now, simulation can be started via *Simulate Item* under *Simulation Menu as shown Figure 5.6.*. Progress length of train are taken between 0 and 50. This length, which is randomly produced for each train in each simulation step is shown in Simulation Panel. If the *Progress Length* is 0, trains will not move.

Figure 5.6 : Simulation Panel

8) If a train encounter a signal box as it moves, a warning occurs on the *Main Panel* such as "**Train1 is waiting at Red Signal Box**". In this situation, *Traffic Control Center* (*TCC*) sets in and moves the train via, *Traffic Control Center* item under *Simulation Menu as shown Figure 5.7.*.

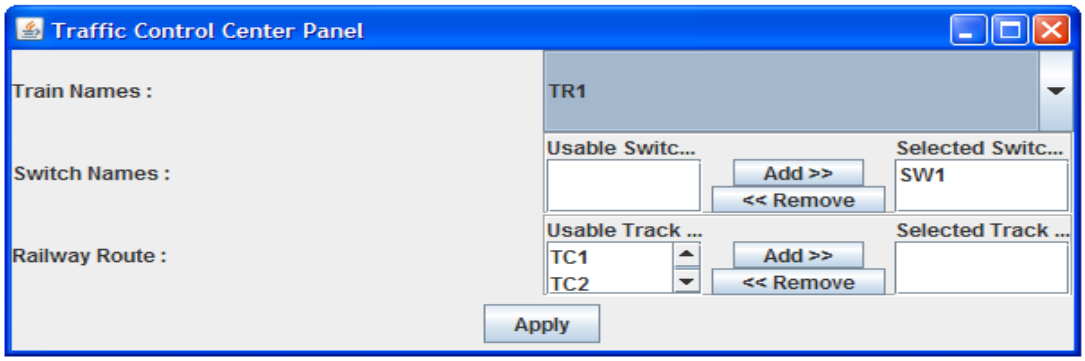

Figure 5.7 : Traffic Control Center(TCC) Screen

5.2 Transition from Program Layout to Petri Nets

The following steps should be executed in order to obtain transition from program layout to Petri Nets:

1) Following the first 3 steps of the previous section a project is opened or created and all necessary components are added to simulate the railway yard under consideration.

2) *Petri Nets* item is selected via *Convert To Petri Nets* under *Petri Nets Menu* as shown Figure 5.8 and JPNS Editor is seen as shown Figure 5.9. After a project is opened, its basic Petri Nets can be seen by selecting *Convert To Petri Nets* item.

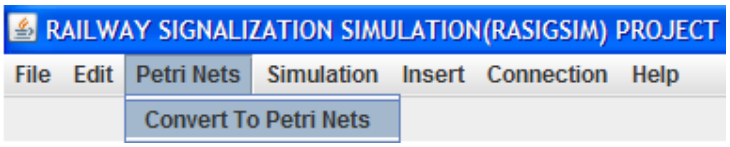

Figure 5.8 : Convert To Petri Nets Menu Item

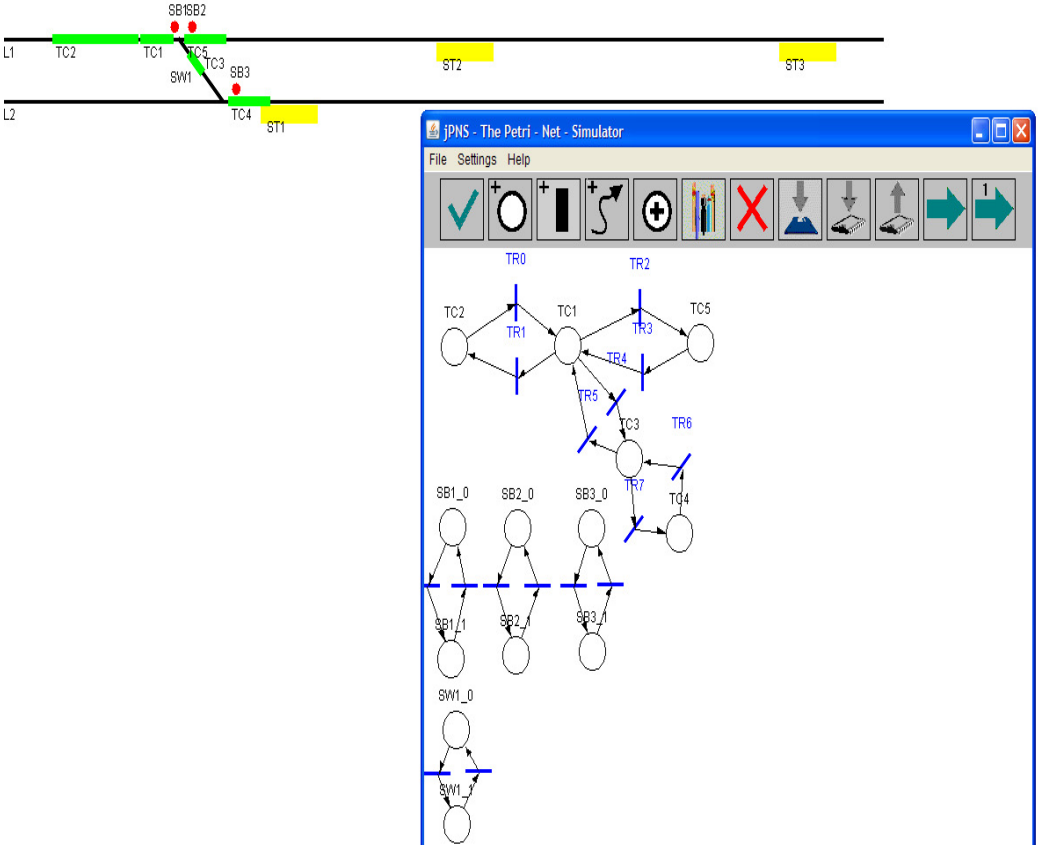

Figure 5.9 : Transformation of interlocking system model to basic Petri Nets.

6. CONCLUSION AND RECOMMENDATIONS

In this graduation project, a Simulation and SCADA Program was developed for Railway Interlocking Systems. A feature that provides transformation from interlocking system model to basic Petri Nets and STL PLC program, was added to the project with other features. The user interface and usage of the program was kept as simple and as user-friendly as possible. Also program' s database and object oriented design were explained in detail. By adding contents of the source files to the user manual, it was made clear for the developers to comprehend the program easily. Also this project enables to be able to be simulated a lot of undergraduation and graduation projects. In addition to these; database installation, the user manual of the software and the development documentation(javadoc) are added to the software CD...

REFERENCES

[1] Url-1 <*http://www.nycsubway.org/tech/signals/interlocking.html*>, accessed at 15.03.2009.
[2] Url-2 <*http://www.scadalink.com/scada.htm* >, accessed at 25.03.2009.
[3] Url-3 <*http://www.petrinets.info/docs/proposal.html* >, accessed at 03.04.2009.
[4] Url-4
<http://www.opcfoundation.org/Default.aspx/01_about/01_whatis.asp?MID=AboutOPC >, accessed at 03.04.2009.
[5] Url-5 <*http://www.uytes.com.tr/simulasyon/index.html*>, accessed at 01.04.2009.
[6] Url-6 <*http://www.absoluteastronomy.com/topics/Interlocking* >, accessed at 15.04.2009.
[7] ***JEasyOPC Manual***

APPENDICES

APPENDIX A.1 : Program Menus

APPENDIX A.1

PROGRAM MENUS

File Menu - New Project Menu Item

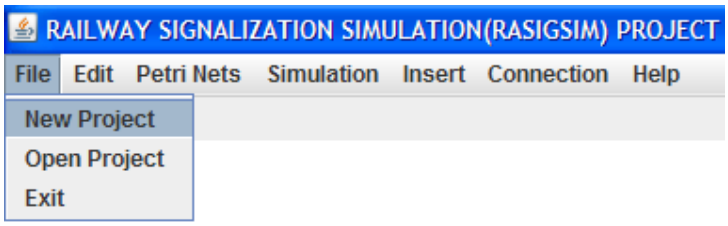

Figure A.1.1 : New Project Menu Item

New Project item is selected via New Project under File Menu as shown Figure A.1.1 and New Project Dialog is seen as shown Figure A.1.2. After Project Name is written and Date is selected, New project can be added by clicking Save Button.

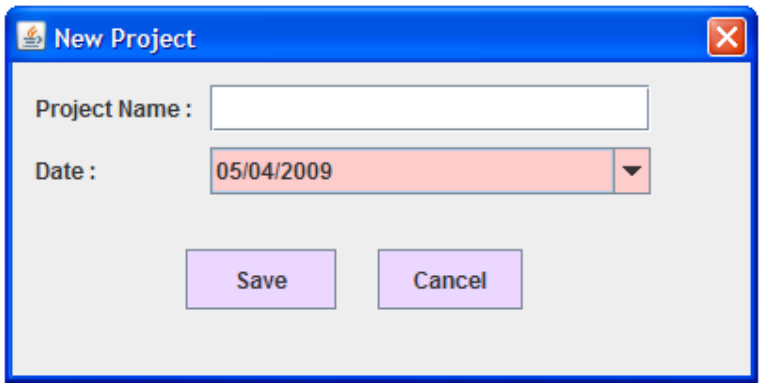

Figure A.1.2 : New Project Screen

File Menu - Open Project Menu Item

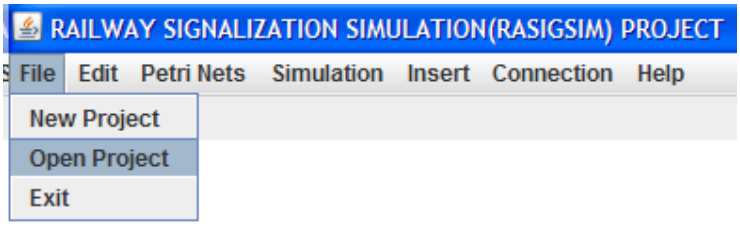

Figure A.1.3 : Open Project Menu Item

6.1.1

Open Project Item is selected via Open Project under File Menu as shown Figure A.1.3 and Open Project Dialog is seen as shown Figure A.1.4. After Project is selected, selected project is opened by clicking Open Button.

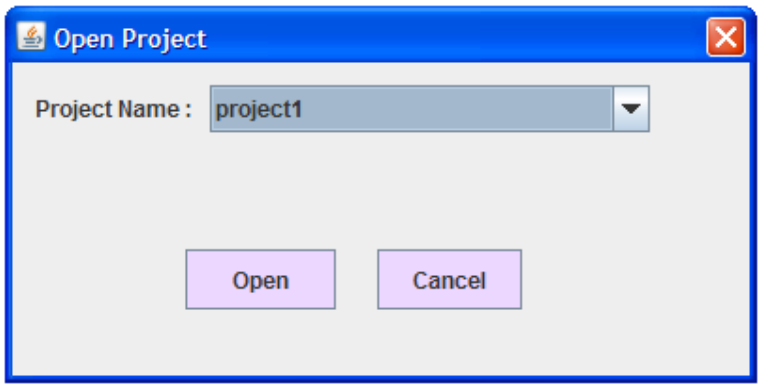

Figure A.1.4 : Open Project Screen

File Menu – Exit Menu Item

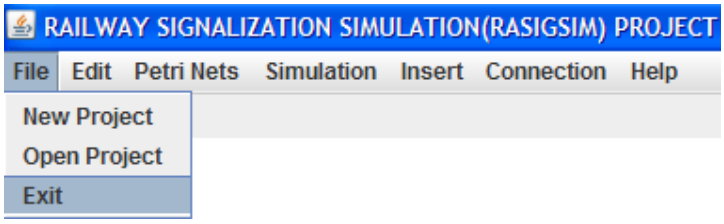

Figure A.1.5 : Exit Menu Item

Exit Item is selected via Exit under File Menu as shown Figure A.1.5 and Program can be closed by clicking Exit Item.

Petri Nets Menu – Convert To Petri Nets Menu Item

Petri Nets Item is selected via Convert To Petri Nets under Petri Nets Menu as shown Figure 5.8 and JPNS Editor is seen as shown Figure 5.9. After a project is opened, its basic Petri nets can be seen by selecting Convert To Petri Nets Item.

Simulation Menu – Simulate Menu Item

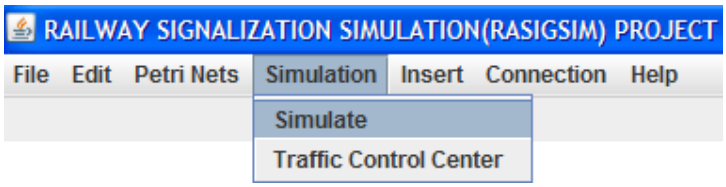

Figure A.1.6 : Simulate Menu Item

Simulate Item is selected via Simulate under Simulation Menu as shown Figure A.1.6 and Simulation Panel is seen as shown Figure 5.6. Progress measure of train must be between 0 and 50 so it is produced in this range via Simulation Panel. If Progress Length is 0, trains will not forward.

Simulation Menu - Traffic Control Center Menu Item

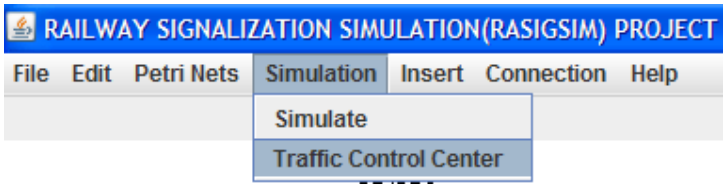

Figure A.1.7 : Traffic Control Center Menu Item

Traffic Control Center Item is selected via Traffic Control Center under Simulation Menu as shown Figure A.1.7 and Traffic Control Center Panel is seen as shown Figure 5.7. If A train encounter with a signal box when it moves, A warning occurs

on the Main Panel as “Train1 is waiting at Red Signal Box”. In this situation, Traffic Control Center(TCC) sets in and moves the train via, Traffic Control Center Item under Simulation Menu.

Insert Menu – Line Menu Item

Line Item is selected via Configuration Parameters under Insert Menu as shown Figure 5.2 and Line Dialog is seen as shown Figure A.1.8. Line components characterize railway lines which are used to move passengers or goods by trains. When a line is added, Line Name, Line Start Point and Line End Point are required as shown Schema X.

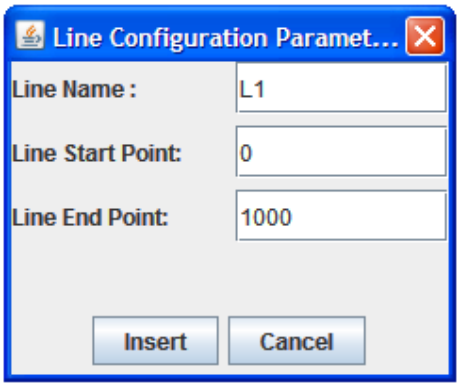

Figure A.1.8 : Line Parameter Screen

Insert Menu – Switch Menu Item

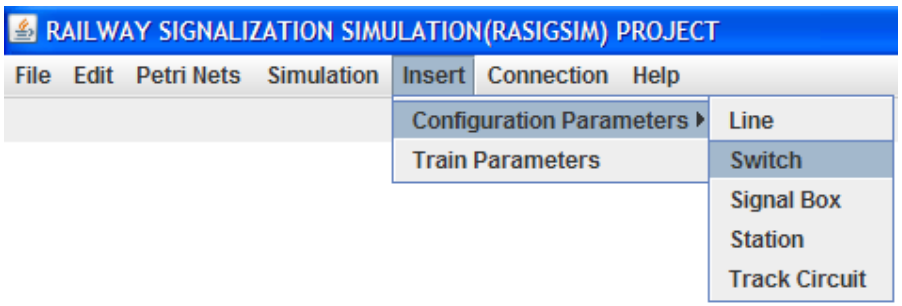

Figure A.1.9 : Switch Parameter Menu Item

Switch Item is selected via Configuration Parameters under Insert Menu as shown Figure A.1.9 and Switch Dialog is seen as shown Figure A.1.10. Transition from a line to another line is achieved with Switch components. When a switch is added, Switch Name, First Line, Second Line, First Line Point and Second Line Point are required as shown Schema X. First Line and Second Line are explained that switch is located between which lines. First Line Point and Second Line Point are explained that switch is located from which point on First Line to which point on Second Line.

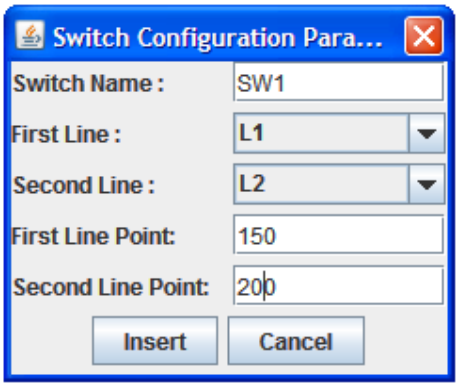

Figure A.1.10 : Switch Parameter Screen

Insert Menu – Signal Box Menu Item

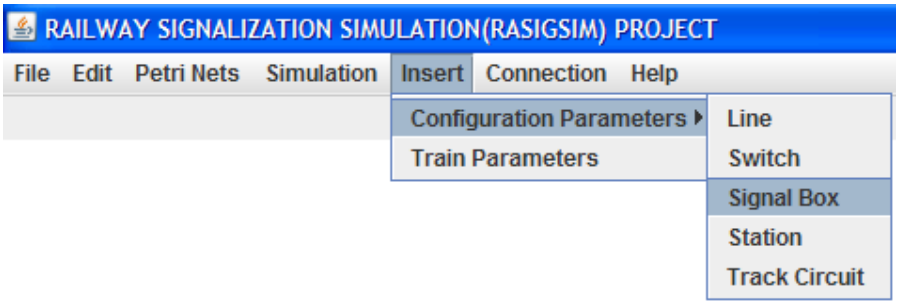

Figure A.1.11 : Signal Box Parameter Menu Item

Signal Box Item is selected via Configuration Parameters under Insert Menu as shown Figure A.1.11 and Signal Box Dialog is seen as shown Figure A.1.12. The movements of Trains are controlled by Signal Box components. When a Signal Box is added, SignalBox Name, Line, Direction, Track Circuit and Point are required as shown Schema X. Line is explained that signal box is located which line. Direction is

specified that signal box is used for which direction. Track Circuit is selected in order to be integrated with signal box. Point is explained that signal box is located to which point on the line.

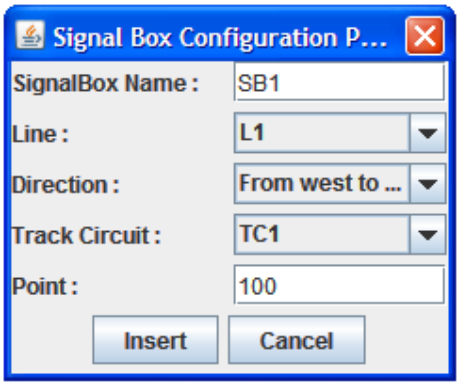

Figure A.1.12 : Signal Box Parameter Screen

Insert Menu – Track Circuit Menu Item

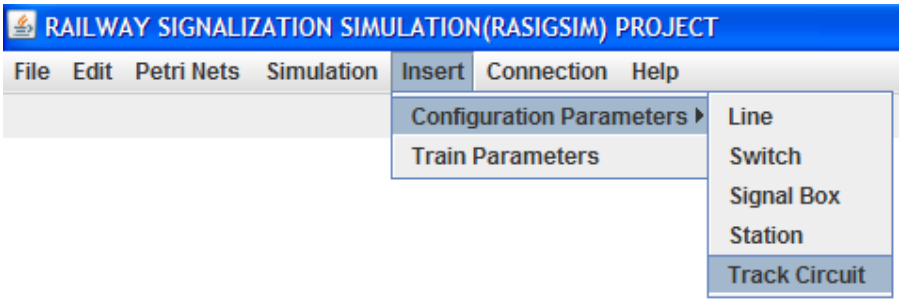

Figure A.1.13 : Track Circuit Parameter Menu Item

Track Circuit Item is selected via Configuration Parameters under Insert Menu as shown Figure A.1.13 and Track Circuit Dialog is seen as shown Figure A.1.14. The movements of Trains are tracked by Track Circuit components. When a Track Circuit is added, Track Circuit Name, Line or Switch, Start Point and End Point are required as shown Schema X. Line and Switch are explained that track circuit is located on which line or switch. Start Point and End Point are explained that track circuit is located from which point on line or switch to which point on line or switch.

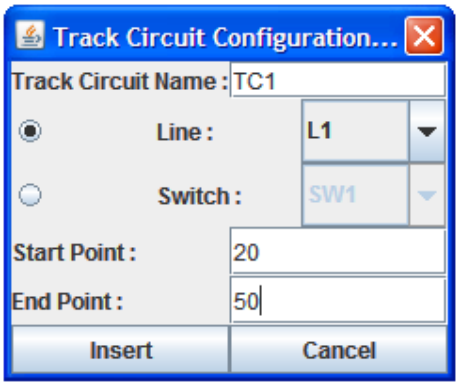

Figure A.1.14 : Track Circuit Parameter Screen

Insert Menu – Station Menu Item

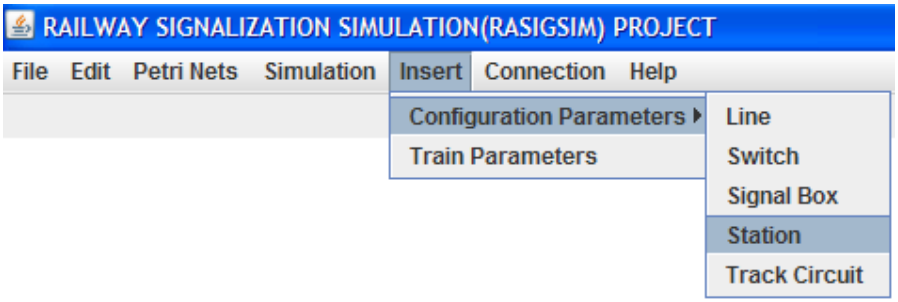

Figure A.1.15 : Station Parameter Menu Item

Station Item is selected via Configuration Parameters under Insert Menu as shown Figure A.1.15 and Station Dialog is seen as shown Figure A.1.16. Station can be added via Station Configuration Parameter Frame. When a station is added, Station Name, Line and Point are required as shown Schema X. Line is explained that station is located on which line. Point is explained that station is located on which point on the line.

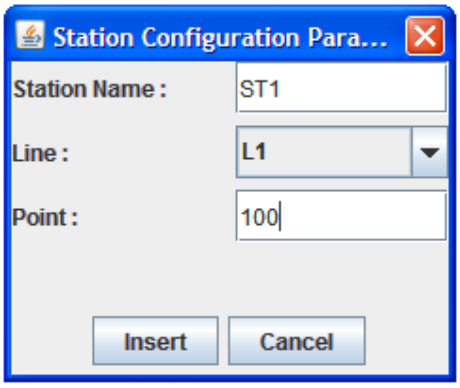

Figure A.1.16 : Station Parameter Screen

Insert Menu – Train Menu Item

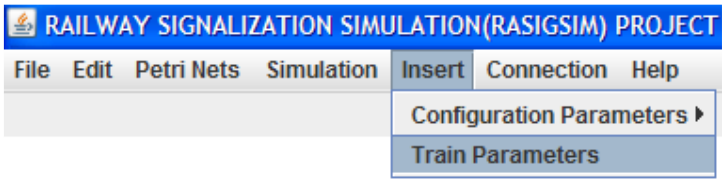

Figure A.1.17 : Train Parameter Menu Item

Train Item is selected via Configuration Parameters under Insert Menu as shown Figure A.1.17 and Train Dialog is seen as shown Figure A.1.18. Trains can be added via Train Parameters Frame. When a Train is added, Train Name, Line, Direction, Head Point and Length of Train are required as shown Schema X. Line is explained that train is located which line is. Direction is specified that train is used which direction is. Head Point is explained that head point of train is started from which point on the line. Length of train is specified that Length of train on the line.

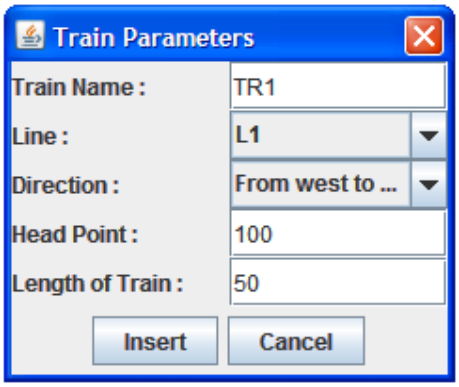

Figure A.1.18 : Train Parameter Screen

Insert Menu – Connection Menu Item

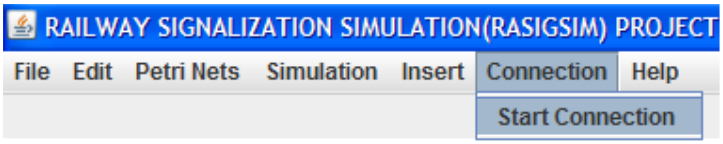

Figure A.1.19 : Start Connection Menu Item

Start Connection Item is selected via Start Connection under Connection Menu as shown Figure A.1.19 and initial values are gotten from PLC via Start Connection Item under Connection Menu. Thus, Simulation and PLC has been synchronized.

Help Menu – Help Menu Item

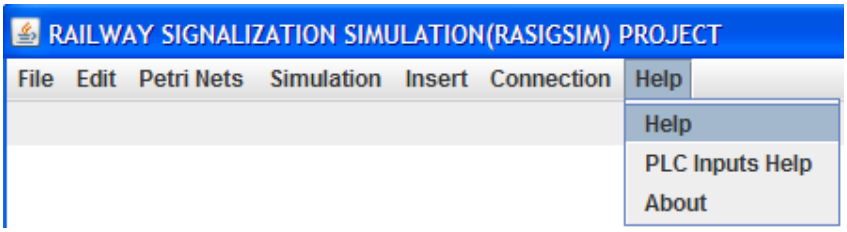

Figure A.1.20 : Help Menu Item

Help Item is selected via Help under Help Menu as shown Figure A.1.20.

Help Menu – PLC Inputs Help Menu Item

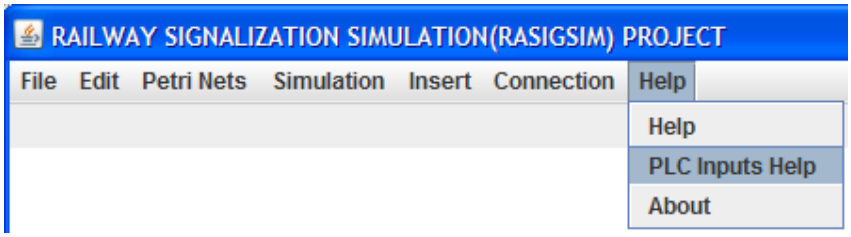

Figure A.1.21 : PLC Inputs Help Menu Item

PLC Inputs Help Item is selected via PLC Inputs Help under Help Menu as shown Figure A.1.21. The Names of Inputs of Interlocking PLC Software must be specified according to these names PLC Inputs Help File. After selecting PLC Inputs Help, PLC Input Names file is seen as shown Figure 5.12.

CV

Candidate's full name:

Ömer Eren Avşaroğulları

Place and date of birth:

Gaziantep - 10.01.1980

Permanent Address:

-

Universities and Colleges attended:

Istanbul Technical University, Control and Automation Engineering – M. Sc. Degree

Karadeniz Technical University, Elec. and Electronic Engineering – B. Sc. Degree

Publications:

-