Railway Signal Interlocking Logic Simulation System

Lidong Zhang
Shandong Jiaotong University
Haitang Road 5001
Jinan City of Shandong Province,
China
zhanglidong1102@126.com

ABSTRACT

Railway interlocking systems are apparatuses that prevent conflicting movements of trains through an arrangement of tracks. In this paper, we formulated the main way to design a railway signal interlocking simulation system. To simulate the interlocking logic of railway signal, we first analyzed such devices as signals, track circuit, switches and train routes. Then we designed such classes as signal class, track circuit class, switch class and route class based on object-oriented programming language. By defining the attributes of every class and taking full consideration of the signal relays' types and amounts, we developed the interlocking logic simulation system with C# language. The simulation system is applied on the actual station chart of downward throat and proves it's applicable. The system realized interlocking logic implementation and signal opening functions. Being put into practice, it proves to be worthy of promotion and widely used.

CCS Concepts

Computing methodologies~Systems theory

Keywords

Railway signal; Interlocking logic; Computer simulation

1. INTRODUCTION

The formal description of railway station signal system's interlocking logic is vital not only for the computer interlocking software development, but also for the interlocking software's test. Many scholars have made studies on how to develop or test interlocking simulation system. Xiu H. introduced a new network computer interlocking simulation system which is designed based on B/S mode, ASP.NET and Web 3D technology[1]. Cui N.N., Dong Y., Zhou Y. built simulation platform of a complete Computer Interlocking System by software to meet the interlocking equipment provided by different manufactures[2]. Li Z. , Liu J. and Sun H. proposed an approach to generate test case automatically with the help of SMT Solver after extracting the yard specification written by boolean expressions from interlocking rules and configuration specification[3]. Liu Z.W., Li S.B. , Li Y. developed WebSocket-based interface program as a communication interface to make Automatic Train Supervision (ATS) simulation system compatible with computer interlocking(CI) system[4]. Roanes-Lozano E., Alonso J A. and Hernando A. developed implementations models based on matrix, algebraic, logic and logic–algebraic approaches allowing routes compatibility to be decided in advance[5]. Guan Q., Tao H Q. and Huang B. put forward railway switch failure prediction algorithm based on least squares support vector machine [6]. Cappart Q., Limbree C. and Schaus P. proposed a discrete event simulation approach limiting the verification to a set of likely scenarios, which solved state space explosion problem of interlocking verification[7]. Samootrut P., Lertwatechakul M. and Tongkrairat S. developed an in-house fixed-block train control simulation system to be used as a tool for helping learners to understand train control processes[8].

The remainder of this paper is organized as follows, section 1 presents the state-of-art of railway signal interlocking simulation system, which would bought enough bases and proofs to make our study meaningful. Section 2 studies the railway signaling devices' functions, principle and working state, such signal machine, switch or turnouts, track circuit and route, which were the main simulation objects. Section 3 elaborates how to build the devices class, station yard model, button choice and interlocking logic realization. Section 4 gives an experimental example simulation system and concludes the study. At last, we thank the foundation support.

2. RAILWAY SIGNALING DEVICES

Railway signaling is a safety system used on railways to prevent trains from colliding. Generally speaking, the railway signal devices include signal machines, switches and circuit tracks.

From the viewpoint of device functions, railway signal machines can be divided as home signal or home & shunting signal, starting signal or starting & shunting signal, shunting signal, and distant signal. The shunting signals' type includes single signal location, double signal location, differential signal location, stub-end track signal. The signals have three kinds of states, which are signal at stop, signal at clear, and signal fault. And signal at stop is its default state. For example, the red indication is home signal's default state, and the blue indication is shunting signal's default state.

ICCCV 2019, June 15–18, 2019, Jeju, Republic of Korea

ACM ISBN 978-1-4503-6322-8/19/06…$15.00
https://doi.org/10.1145/3341016.3341017

Home signal Shunting signal

Figure 1. Typical rail signal

A track circuit is a track clear detection device consisting of an electrical circuit fed through the running rails of a section of track.

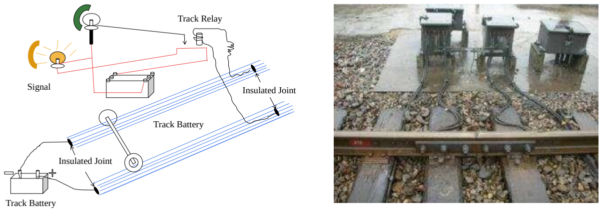

Figure 2. A track circuit

Each section is electrically isolated from others. Generally, an elementary track circuit consists of a source of energy, limiting resistance, rails and rail bonding, ties and ballast, relay series resistance, and a track relay. If a track circuit has one or more switch, we call it a branch section, otherwise turnless section. A track circuit' working state includes adjusting or empty state, occupied state, and track breaking state.

A railroad switch is a mechanical installation enabling trains to be guided from one track to another. A typical railroad switch components include points, frog, guard rail or check rail, switch motor, point lever, joints. Railroad switch working states divide into normal position, reverse position and fault position. And the fault position should be definitely avoided.

Crossover railroad switch Single operation railroad switch

Figure 3. Two kinds of railroad switch

A route is composed of signaling machines, track sections and turnouts to describe the route of train or shunting train in the station. The establishment of the route requires the closure of hostile signals on the road, the idle track section and the correct position of turnout. According to the nature of operation, it can be divided into train route and shunting route, and its state can be separated as locking state and unlocking state according to whether the route is established or not. A shunting route from D3 to D13 is illustrated in Figure.4.

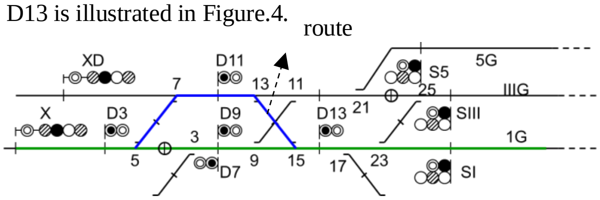

Figure 4 . A shunting route illustration

3. SIGNAL SIMULATION MODEL

3.1 Device Classes

Based on the analysis of the main signal equipment and the programming idea of class in object-oriented, the related classes of signal machine class (CSignal), track section class (CTrack) and turnout class (CSwitch) are designed, and the attributes of each class are set.

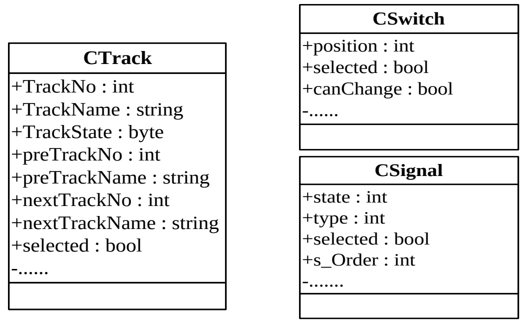

Figure 5. Signaling device classes

CSignal class designs four kinds of attributes, including state attributes for the actual signal, which have three values of integer type, one for opening, zero for closing, and two for failure. Type attributes, set to integer type, use value one to represent inbound signal, value two to represent shunting signal, value three to express outbound and shunting signal. The route selection attribute is set to Boolean type, true value is selected, false value is unchecked. In order to distinguish whether the button corresponding to the signal is the beginning or terminal of the route, the attribute s_Order is designed as an integer variable. When its value is one, it represents the beginning of the route, when its value is two, it represents the terminal of the route, and when its value is three, it represents the route middle button. Part of the signal programming language is as follows,

```
//CSignal Class Instantiation
Signal s_XD = new Signal();//XD signal
Signal s_X = new Signal();//X
//CSignal object initialization
this.s_XD.state = 0;//XD home signal closed
this.s_XD.type = 1;//XD signal type
this.s_XD.selected = false;//XD signal is empty
//X home signal inialization
this.s_X.state = 0;//closed
this.s_X.type = 1;//type
this.s_X.selected = false;//unused
```

The attributes of CTrack class include: TrackNo track segment number is an integer variable; TrackName is a string name variable of track segment; TrackState is three kinds of state of track segment, when it is 0, when it is value one, it is shunt state, and when it is value two, it is broken track. PreTrackNo is the integer variable number of the previous track segment; preTrackName is the string name variable of the previous track segment; nextTrackNo is the integer number variable of the latter track segment; nextTrackName is the string name variable of the latter track segment. Whether or not the track segment is selected is represented by Boolean variable selected, when it is true, it is selected, and when it is false, it is not selected. Part of the track circuit programming language is as follows,

```
// CTrack Class Instantiation
Track t_7DG = new Track();//7DG
```

```
Track t_11_13DG = new Track();//11-13DG
Track t_21DG = new Track();//21DG
Track t_25DG = new Track();//25DG
//CTrack object initialization
 #region Track circuit object inialization
t_7DG.TrackNo = 7;
t_7DG.TrackName = "7DG";
t_7DG.TrackState = 0;//empty
t_7DG.selected = false;//unused
t_7DG.preTrackName = "YXDG";//previous section
t_7DG.nextTrackName = "11-13DG";//following secton
```

CSwitch (Turnout Class) characterizes the class of turnouts. Its attributes include position representation state, which is an integer variable. When its value is 0, it represents positioning, 1 represents inversion, 2 represents four openings; selected attribute is bool type, which indicates whether the equipment currently belongs to the selected route; and canChange attribute is bool type, which indicates that the turnout equipment is unlocked or locked. Part of the switch programming language is as follows,

```
#region CSwitch Instantiation
CSwitch s_5_7 = new CSwitch();//5/7
CSwitch s_9_11 = new CSwitch();//9/11
CSwitch s_13_15 = new CSwitch();//13/15
CSwitch s_21 = new CSwitch();//21
//CSignal object initialization
#region Switch inialization
this.s_5_7.position = 0;//0-normal position
this.s_5_7.selected = false;//unused
this.s_5_7.canChange = true;//unlocking
```

3.2 Station Model

The main elements of the station yard are signaling machine, track circuit section, insulation joint, turnout (including switch machine) and so on. When choosing the elements of the station, the consistency of appearance and equivalence of attributes are fully considered, that is, the track section is represented by a straight line, the signal machine is represented by a circle, and the turnout rail is represented by a short straight line. Taking vs.net programming environment for example, LineShape control can be used to construct route lines and insulation joints model. OvalShape control in the system can be used to construct signal model. The BorderColor property of LineShape control is used to set the adjustment and shunting state of the track section, and the FillColor property of OvalShape control is used to set the display state of the semaphore. The model utilities are shown in Figure 6.

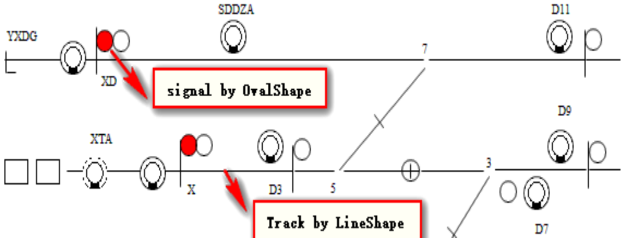

Figure 6. Part of a station model

3.3 Button Choice Logic

Based on the interlocking table, the buttons can be grouped by $B=\{B_i, i=1,2,\ldots,N\}$, here, N is the total number of buttons in the station. The routes can be grouped by $R=\{R_i, i=1,2,\ldots,M\}$,here M is the total routes amount. Each specific button corresponds to the design object of the semaphore class. In the process of route handling, the first-pressed button is the beginning of a route, otherwise as the terminal button. If the total number of routes that the i^{th} button can establish is set as $R_i=\{R_{ik}, k=1,2,\ldots,M_i\}$, M_i is the route amount for i^{th} button. Also, the $R_{ik}=\{B_i, B_k\}$ symbolizes the route start button is B_i ,and the terminal button is B_k . The following logic will define whether the button is a beginning or terminal type,

$$
\begin{array}{ll}
\textit{If} & (\prod_{k=1}^{N_i}(B_k\text{.selected is false}) \text{ is true}) \\
\textit{then} & \{B_i \text{ is a starting button}\} \\
\text{else} & \{B_i \text{ is an end button}\} \\
\textit{end} &
\end{array}
$$

There are also some special uses buttons, which mainly involve manual unlocking, route cancellation and normal unlocking. Route unlocking can be divided into normal results and manual unlocking, while manual unlocking can be divided into manual unlocking and canceling routes, which are applicable to different situations. According to the relationship between manual unlock button circuit and cancel circuit in 6502 electric centralized interlocking circuit, it can be seen that when the manual unlock button is pressed, the excitation of cancel circuit will be triggered; when the cancel button is pressed, the cancel operation will be executed if the route button is pressed again. So, the logic of designing simulation judgment is as follows, for the given button B_i , when it is pressed, whether to cancel or establish the route operation is judged by,

if man-release button is pushed
then the route cancel button is activated
end
when the route button is pushed
if the route cancel button is chosen
then to check cancel condition and execute cancel operation
else to execute build route operation
end

3.4 Interlocking Logic Realization

In 6502 electric centralized interlocking system, the order of interlocking condition determination is as follows, after the route selection is completed by 1^{st}-6^{th} line, the consistency of route selection is checked by line 7 network, and the signal checking relay circuit (XJJ) on 8^{th} line is used to check the possibility of open signal, that is, whether the route is idle, and no hostile route is established, and the turnout position is correct. The 9^{th} line is the section state inspection relay (QJJ) and the section inspection relay (GJJ) circuit, which is used to check the idle section and realize route locking. The 10^{th} line is the self-closing circuit of Section Check Relay (QJJ), which is used to prevent the Section Fault Unlocking mode from mis-unlocking the route. The 11^{th} line is a signal relay (XJ) circuit, which checks that each section of the road is locked, the turnout position is correct, and the hostile route is not built, so that the signal can be opened if the conditions are met. Based on above analysis, combined with turnout operation alone and general locking operation, the interlocking inspection is carried out according to the following logic,

if single operation button or guidance total locking button is pushed down
and route switch positon is wrong
then Exit
else *if* route section is occupied
then Exit
else if hostile signal is open
then Exit
else
build the route and lock it
end

Here is part of a shunting route from D7 to D3 logic judging programming language,

```
#region ShuntingRouteFromD7toD3
if (this.s_D7.selected)//D7-D3
{
//SwitchInterlockingConditionChecking
if ((this.s_1_3.selected || (this.s_1_3.canChange == false &&
1 == this.s_1_3.position))|| (this.s_5_7.selected
||(this.s_5_7.canChange == false && 1 ==
this.s_5_7.position)))
{
    MessageBox.Show("Switch Interlocking Conditons arenot
    Satisfied,the Route cannot be Built!", "Error");
    this.s_D7.selected = false;
    this.s_D3.selected = false;
    return;
}
else{
//TrackCircuitInterlockingConditionChecking
if (t_3DG.selected || t_3DG.TrackState != 0 || (t_5DG.selected
|| t_5DG.TrackState != 0))
{
    MessageBox.Show("Track CiucuitInterlocking Conditons
    arenot Satisfied,the Route cannot be Built!", "Error");
     this.s_D7.selected = false;
    this.s_D3.selected = false;
     return;
}
//HositleSignalConditonChecking
else if ((this.s_D3.state != 0) || (this.s_X.selected))
{
    MessageBox.Show("Signal Interlocking Conditons arenot
    Satisfied,the Route cannot be Built!", "Error");
    this.s_D7.selected = false;
    this.s_D3.selected = false;
    return;
}
else//RouteCanBeBuilt
{
    (to be omitted)…………………………………….
}
```

4. EXAMPLE AND CONCLUSIONS

4.1 Application Example

It is a typical station layout well-known in railway signal specialty in Figure 7. For the station downstream throat, there are three home signals, seven shunting signals, five starting signals, eleven track circuits, two single movement switches, and six pair of switches.

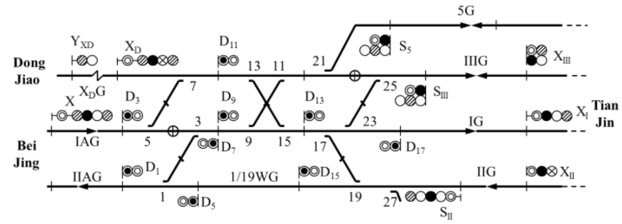

Figure 7. Downstream throat

As we can see that in Figure 8., there are one or two buttons for each signal machine according to their different function. For example, the X signal has two buttons, where one is for train route setting, and the other is for station through route setting. Also, starting signal S_{II} has two buttons. One button on the line is used for train route setting, and the other beside the line is for shunting route handling. Every signal has signal repeater. The repeater can show opening or closing state of its dominant signal.

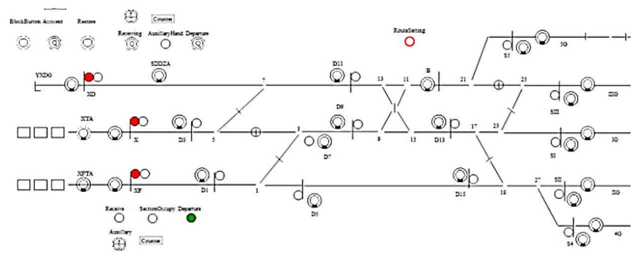

Figure 8. Interlocking simulation system interface

The simulation system realized many functions, such as route handing, route cancelling, signal opening, white track belt showing, diagnostics simulate, and so on. Figure 8 is a shunting route handling process. When the interlocking conditions satisfy, the D3 signal is opening, and 5th and 7th track circuit show white belt, which stands for the route is built successfully.

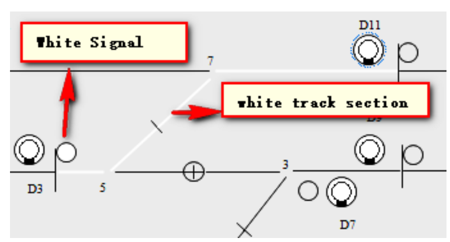

Figure 9. A shunting route and its interface

4.2 Conclusions

In the railway domain, an interlocking is the system ensuring safe train traffic inside a station by controlling its active elements such as the signals or points. Railway station interlocking system is very complex and vitally safe. To better master its interlocking principles, we brought forth the new simulation way with object-oriented programming language, which realized integrated design procedure. Firstly, it made the station yard drawing easier by using controls in the developing environment. Secondly, it fully implemented the complex interlocking logic by designing the class and its attributes' properly. Our simulation system would be release the interlocking logic knowledge study burden. So far, the train control logic is still needed to be improved in order to control the train movement to reach each location at the precisely scheduled time. It may be necessary to apply more accurate function to work with the train model specification such as breaking performance.

5. ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (Grant No. 61773243), Natural Science Foundation of Shandong Province (Grant No.BS2014DX011, ZR2017MF011), and the Climbing Plan Foundation of Shandong Jiaotong University(SDCPF201804).

6. REFERENCES

[1] Xiu H .2016. Research and design of the new network computer interlocking simulation system. In *the 8th International Conference on Measuring Technology & Mechatronics Automation*(Macau, China, March11-12, 2016). IEEE, DOI: http://dx.doi.org/10.1109/ICMTMA.2016.122.

[2] Cui N.N., Dong Y., Zhou Y.2010. Research on simulation platform of computer interlocking system. *Railway Computer Application*,19,12(Apr.2010),57-59.DOI: http://dx.doi.org/10.3969/ j.issn.1005-8451.2010.12.017.

[3] Li Z. , Liu J. , Sun H. , et al. 2017. Automatic test generation of large boolean expressions in computer based interlocking system. In *the Asia-pacific Software Engineering Conference. Computer Society*(The Nanjing, China , Dec. 4-8,2017). IEEE, 513-520. DOI: http://doi.ieeecomputersociety.org/10.1109/APSEC.2017.58.

[4] Liu Z.W., Li S.B. , Li Y. , et al. 2017. Remote ATS simulation system based on webSocket communication protocol. In *International Conference on Intelligent Computation Technology &Automation* (TheChangsha,China,Oct.9-10 2017).IEEE,327-330. DOI: http://doi.ieeecomputersociety.org/10.1109/ICICTA.2017.79.

[5] Roanes-Lozano E, Alonso J A, Hernando A.2014. An approach from answer set programming to decision making in a railway interlocking system. *RACSAM*,108, 2,(Nov.2013),973-987.DOI: http://dx.doi.org/10.1007/s13398-013-0155-1.

[6] Guan Q., Tao H. Q., Huang B..2014.The computer interlocking software system reliability test based on the Monte Carlo. *Applied Mechanics & Materials*, 614,(Sept.2014),397-400. DOI: https://doi.org/10.4028/www.scientific.net/AMM.614.397.

[7] Cappart Q, Limbree C, Schaus P, et al. 2017. Verification of interlocking systems using statistical model checking. In *18th International Symposium on High Assurance Systems Engineering (HASE)*(Singapore,Singapore,Apr.27,2017).IEEE,61-68. DOI: http://dx.doi.org/10.1109/HASE.2017.10 .

[8] Samootrut P, Lertwatechakul M, Tongkrairat S, et al. 2016. On development of train control and signaling simulation. In *the International Conference on Control, Automation and Systems*. (The Gyeongju, South Korea, Oct.16-19,2016), IEEE, 671-674 .DOI: http://dx.doi.org/10.1109/ICCAS.2016.7832391 .