

Train Operation Dispatching Command System Design

Tito E. Mateshi¹, Dening Jiang²

¹Tianjin University of Technology and Education (TUTE). Tianjin, People Republic of China (China)

²Tianjin University of Technology and Education (TUTE), Tianjin, People Republic of China (China)

Abstract: *This paper proposes simulation of train operation dispatching command system design which achieves intelligent identifying, locating, tracking, monitoring and managing railways by using Visual studio Enterprise 2015. The paper will mainly base on Railways system and control center; it will not cover on System on the train itself. This simulation program is mainly based on Chinese Train Control System(CTCS), and we use Beijing - Shanghai route as its case study. We make used of special features of Visual studio to achieve train control. Features like timer control, PictureBox, Label, Textbox, etc. to achieve train movements, tracking, locating, timing, signaling and monitoring which help us in controlling the trains along the lines.*

Keywords: Railway System, Block section, Dispatching Command, Train Control System

1. Introduction

Train control is the process by which the movement of rail transit vehicles is regulated for the purposes of safety and efficiency. The process is carried out by a combination of elements some men and some machines located on the train, along the track, in stations, and at remote central facilities. These elements interact to form a command and control system.

In this paper we are going to see how we achieve **Train detection** (that is monitoring of the railway track to determine trains locations), **Train separation** (insure that trains on the same track maintain a safe following distance so as to avoid collisions), **Route interlocking** (preventing trains on crossing or branching routes from making collision), **Station stopping** (insuring the train stops within specified area in a station), **Train starting** (initiating train departure from a station by giving a line clear signal), **Train Supervision**, and **Train dispatching** (that is controlling train departures from terminals or waypoints in accordance with the track status) by using visual studio enterprise 2015.

As this paper is mainly based on CTCS let see a brief overview of it, Chinese Railway Train Control System (CTCS) has Learned from Europe Train Control System (ETCS) development and foreign experience in high-speed rail control system, combining with Chinese railway transportation characteristics, and following the principle of unified planning the whole railway, Chinese Ministry of Railway (MoR) build a Chinese Train Control System (CTCS).The target of CTCS is to improve safety function and Efficiency of transport, satisfy interchange operationregulating the system design and system classification to adapt to the demand of development. CTCS is divided into 5 levels depending on the devices and technology used.

For safety and efficient operation of a railway system, it is very important to know the locations of trains at all times.

Visual Basic 2015 is Microsoft'atest incarnation of the enormously popular Visual Basic language, and it's fundamentally different from the versions that came before it. Visual Basic is more powerful and more capable than ever before, and its features and functionality are on par with "higher-level" languages such as C++.

In this work, we have used Visual Basic toolbox to build an interface with which station master's and other authorized railways personnel can interact with the program, and we have written codes in strategic event handlers to make our program perform things that we want them to perform.

We have used Visual Basic PictureBox features to display pictures and signaling equipment need by our project, timer function has been used to provide timing of the trains, that is movement of the train according to priority given, designing speed of each trains, It also helped us designing signaling of the dispatching system. Labels and command features has helped us giving names to identify different features of the dispatching system and ability to interact with the system by giving commands we want the system to execute respectively.

We have used Visual Basic special features like multiple forms and form Interaction, coordinates and positions to enable us linking and tracking position of the trains along the lines, that is from one station to another and from one block section to the next.

2. Abbreviations

ETCS – European Train Control system
CTCS – Chinese Train Control system
MoR – Ministry of Railway
VB – Visual Basic
TDCS – Train Dispatching Command System

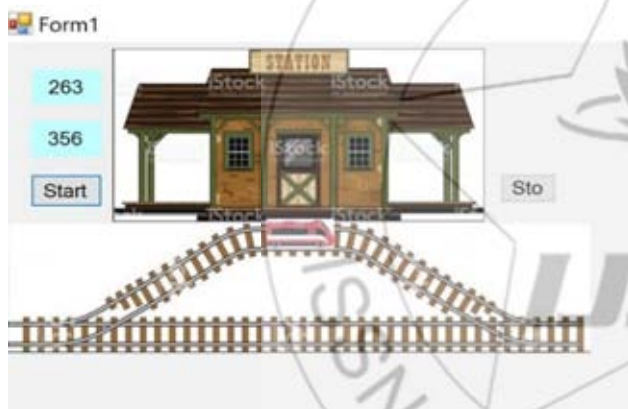
3. A Walk through a Train System

Train system consist of many parts from Train Stations, interlocking, signaling, block section, central control,

vehicles, to yards and shops. All of these makes a successful train system.

Train Stations; The passenger’s first point of contact with a train system is the station. The most prominent features- vending and fare collection facilities. It’s also a place where trains crossing. Normally stations have a station master who oversee trains operation at the station. He is responsible for shunting and issuing of line clear. This is only when the block section is clear, that is there is no train occupying the block section at the time of issuing line clear. It is a place where by the trains with higher priority get the chance to overtake the ones with lower priority by taking all the trains with lower priority out of the main line and allow only the ones with higher priority to proceeds first.

In our simulation program this has been achieved by using visual basic controls, and events features. We used PictureBox, Commandbuttons and Timer control to design train stations. Here timer is used to provide train movement and also for timings. That is to set train priority. The picture 1 shows a layout of our station. From the picture, it can be observed there is a train with less priority resting out of the main line to pave ways for the trains with higher priority to proceed first.



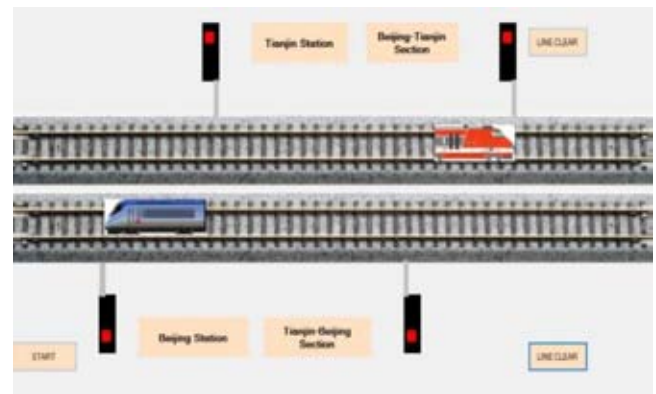
Picture 1

Signaling; this forms the communication part between the drivers and station masters, through these signaling a driver receive status of the lines and guidelines of whether to proceed, reduce speed or to stop the train. Signaling must be very accurate as any mistake will lead to accidents and delays. It’s very important for the drivers to always follow directions given by the signaling equipment’s for the safety operation of railways.

In our simulation program, we achieve signaling feature by using PictureBox, timer and label features of the visual basic to create wayside signaling. The three colors Green, Yellow and Red are the most commonly used signaling color in CTCS which tells the train drivers whether to stop, slow down or proceed ahead. Red means Stop, Yellow slow down and Green means proceed ahead, that is if the block section is clear.

From the picture 2 it can be seen that when the train is occupying a block section other trains are not allowed to

access the same block section at that time hence the light should be turned Red. And soon after the train has cleared from that section the light should turn Green to allow other trains to access the block section.



Picture 2

Block Section; this is the section between two interlocking or stations. To avoid collision no trains are allowed to access the block section at the same time. They can only be allowed to access the block section after the other train has already cleared the block section. Block section clearance forms the security party of the trains. It prevents collision of the trains. Upon reaching block sections drivers are required to obey the directions given by station masters through signaling equipment.

In our project, we have used labels to show block section names of each section along the line which is from Beijing to Shanghai. Also by using coordinates and position features of Visual basic we were able to track position of the trains at each section along both lines, that is from Beijing to Shanghai and from Shanghai to Beijing.

Also by using multiple forms and form interaction we were able to link all of the blocking sections and in turn it help us forming a complete train operation dispatching command system. This can be seen from the picture 3.



Picture 3

Central Control; this is the main control center where by all stations are linked together. From this control center, you can

access all information of the lines and vehicle positions. You can view activities done at any station of your choice. In our project, we have used labels to display different functions, command buttons has been used to offer command features and we also use form interaction feature of VB to link all the forms and their commands. This can be seen from the picture 4. From the picture, we can see trains schedule and stations control. The two help us tracking position of the trains and their schedules.

If you want to see what is happening or movement of trains at particular section or station you click view button of the corresponding station. The design allows a person from control station to issues line clear for both lines by just clicking the corresponding line he/she wish to give the line clear command.

Where by line 1 means the line that starts from Beijing to Shanghai Station and line 2 means the line that starts from Shanghai to Beijing station.

The control panel also offers day, date and real time of the day, it also gives the running time (Duration) of the trains.



Figure 4

4. The Timer Object

A Timer simply generates events at regular intervals. It can be customized using the Properties Window. When the program runs nothing is seen, the timer just ticks. It is an entirely invisible control. A timer event occurs automatically after the amount of time specified in the interval property has passed. These events are handled by the Timer's Tick event handler.

Timer interval is specified in milliseconds, that is 1000 milliseconds equal one second. You can switch a Timer ON and OFF using its Enabled property. When this is true the Timer causes events at the specified interval and when it is false the Timer does not operate. Also, the timer can be enabled by a command code `Timer1.start()` and disabled by a command code `Timer1.stop()` during the execution of the program as it can be shown on Picture 5.

The timer object is the heart of this project, so much depends on timer object from movement of the trains, signaling, to form Interactions.



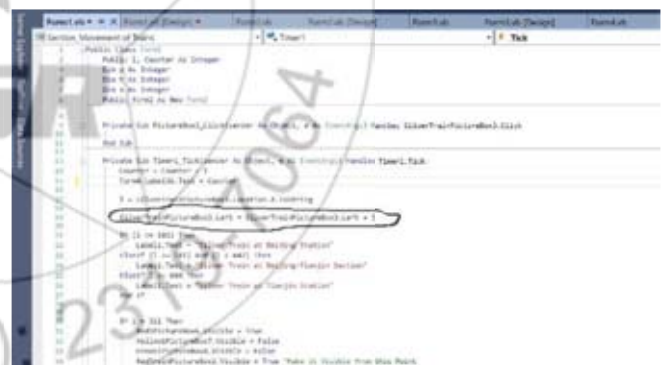
Picture 5

5. Moving PictureBox

A shape or graphic can be moved within its container, which is a form or PictureBox. The move method is used at run time to move a PictureBox object to a new location. By using properties of PictureBox such as **left**, **top**, **width** and **height**. Where by left property changes the position of the left edge of the object and the left edge of the form, top property changes the position of the top edge of the object and top edge of the form, width property changes the width of the object and the Height property changes the height of the object.

By using `PictureBox1.left = PictureBox1.left + 1` code under timer event derived from above properties we were able to attain PictureBox1 movement.

Picture 6 below shows how we attained movement of Silver Colored Train.



Picture 6

6. Position of PictureBox

We have used PictureBox property to display and control movement of trains, we also use the same to track position of the train along the sections. This has been made possible by combined both timer and PictureBox features. These coordinates can be named according to our wish. For our case, we name them according to the names of the stations and block sections we wanted to display in our project. These coordinates can also be used to activate other events as it can be deemed necessary by the programmer.

For example, to display position/coordinates of the moving PictureBox1 we used the following command under the timer object.

PictureBox1.left=PictureBox1.left+5
Label1.Text="X=" +PictureBox1.location.X.To string
Label2.Text=" Y=" +PictureBox1.location.Y.To string

Picture 7 shows these codes as they have been used in our simulation program.

```

10 Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
11     p = PictureBox1.Location.Location + 5
12
13     IF (p <= 290) Then
14         Label1.Text = "Setting south station"
15     ElseIf (p >= 290) And (p <= 400) Then
16         Label2.Text = "Setting train in section"
17     ElseIf (p <= 400) Then
18         Label1.Text = "Setting north station"
19     End IF
20
21     IF p = 330 Then
22         PictureBox1.Visible = True
23         PictureBox7.Visible = False
24         PictureBox8.Visible = False
25     ElseIf p = 400 Then
26         PictureBox1.Visible = True
27         PictureBox2.Visible = True
28         PictureBox3.Visible = True
29         PictureBox4.Visible = False
30     ElseIf p = 450 Then
31         PictureBox1.Visible = False
32         PictureBox8.Visible = False
33         PictureBox9.Visible = True
34         Timer1.Stop()
35     ElseIf p = 500 Then
36         PictureBox1.Visible = True
37         PictureBox10.Visible = False
38         PictureBox11.Visible = False
39         PictureBox12.Visible = True
40     End IF
41 End Sub

```

Picture 7

7. Multiple Forms and Form Interactions

VB 2015 has special features which enable multiple forms interactions, these features help us transferring information and data from one form to another. It helps us to link all individually designed forms to operate as one system which it enables us monitoring the railway tracks to determine trains locations, insure that trains on the same track maintain a safe following distance so as to avoid collisions, preventing trains on crossing or branching routes from making collision, Insuring trains stops within specified area in a station, and Initiating train departure from a station by giving a line clear signal. Picture 8 shows how we have used these features to link and transfer data from one form to another.

```

100 PictureBox1.Visible = True
101 PictureBox2.Visible = False
102 PictureBox3.Visible = False
103
104 ElseIf p = 450 Then
105     PictureBox1.Visible = False
106     PictureBox4.Visible = False
107
108 PictureBox11.Visible = True
109 PictureBox12.Visible = True
110 PictureBox13.Visible = True
111 PictureBox14.Visible = True
112 PictureBox15.Visible = True
113 PictureBox16.Visible = True
114 PictureBox17.Visible = True
115 PictureBox18.Visible = True
116 PictureBox19.Visible = True
117 PictureBox20.Visible = True
118 PictureBox21.Visible = True
119 PictureBox22.Visible = True
120 PictureBox23.Visible = True
121 PictureBox24.Visible = True
122 PictureBox25.Visible = True
123 PictureBox26.Visible = True
124 PictureBox27.Visible = True
125 PictureBox28.Visible = True
126 PictureBox29.Visible = True
127 PictureBox30.Visible = True
128 PictureBox31.Visible = True
129 PictureBox32.Visible = True
130 PictureBox33.Visible = True
131 PictureBox34.Visible = True
132 PictureBox35.Visible = True
133 PictureBox36.Visible = True
134 PictureBox37.Visible = True
135 PictureBox38.Visible = True
136 PictureBox39.Visible = True
137 PictureBox40.Visible = True
138 PictureBox41.Visible = True
139 PictureBox42.Visible = True
140 PictureBox43.Visible = True
141 PictureBox44.Visible = True
142 PictureBox45.Visible = True
143 PictureBox46.Visible = True
144 PictureBox47.Visible = True
145 PictureBox48.Visible = True
146 PictureBox49.Visible = True
147 PictureBox50.Visible = True
148 PictureBox51.Visible = True
149 PictureBox52.Visible = True
150 PictureBox53.Visible = True
151 PictureBox54.Visible = True
152 PictureBox55.Visible = True
153 PictureBox56.Visible = True
154 PictureBox57.Visible = True
155 PictureBox58.Visible = True
156 PictureBox59.Visible = True
157 PictureBox60.Visible = True
158 PictureBox61.Visible = True
159 PictureBox62.Visible = True
160 PictureBox63.Visible = True
161 PictureBox64.Visible = True
162 PictureBox65.Visible = True
163 PictureBox66.Visible = True
164 PictureBox67.Visible = True
165 PictureBox68.Visible = True
166 PictureBox69.Visible = True
167 PictureBox70.Visible = True
168 PictureBox71.Visible = True
169 PictureBox72.Visible = True
170 PictureBox73.Visible = True
171 PictureBox74.Visible = True
172 PictureBox75.Visible = True
173 PictureBox76.Visible = True
174 PictureBox77.Visible = True
175 PictureBox78.Visible = True
176 PictureBox79.Visible = True
177 PictureBox80.Visible = True
178 PictureBox81.Visible = True
179 PictureBox82.Visible = True
180 PictureBox83.Visible = True
181 PictureBox84.Visible = True
182 PictureBox85.Visible = True
183 PictureBox86.Visible = True
184 PictureBox87.Visible = True
185 PictureBox88.Visible = True
186 PictureBox89.Visible = True
187 PictureBox90.Visible = True
188 PictureBox91.Visible = True
189 PictureBox92.Visible = True
190 PictureBox93.Visible = True
191 PictureBox94.Visible = True
192 PictureBox95.Visible = True
193 PictureBox96.Visible = True
194 PictureBox97.Visible = True
195 PictureBox98.Visible = True
196 PictureBox99.Visible = True
200 End Sub

```

Picture 8

8. Applications of Train Operational Dispatching Command System Design

This simulation program is very useful for demonstration of how railways system works. It can be used as teaching guide for students who wish to learn railways transportation system.

It can also be used to study railways networks and risk analysis along the lines.

The system can be linked to Arduino circuit and implement the physical operation of the system. This can be done with the help of sensor for sensing movements and locations of the trains, relays for opening and closing of the interlocking, LEDs for wayside signaling, LCDs for displaying stations, timings, date and locations and GSM receiver for station communications, switches and push buttons for issuing commands like line clear, Start and Stop.

9. Conclusion

We have seen above how we were able to simulate train operation dispatching command system design, the work will be very useful for anyone who has interest in understanding railways system. Also to all those who has got interest in Visual basic and simulation programs.

Furthermore, this simulation work provides a cheaper tool for teaching and simulating railways transport system for all people who wish to acquire knowledge of railways transport as in many countries it is strictly forbidden for unauthorized railways personnel to access railways system.

References

- [1] David I. Schneider “An introduction to programming using visual basic 2010, Eighth Edition”
- [2] B. Ning “Advanced Train Control Systems”
- [3] B. Ning, T. Tang, K. Qiu, C. Gao & Q. Wang “CTCS—Chinese Train Control System”
- [4] Michael Halvorson “Microsoft Visual Basic 2010”
- [5] Thomas Fotherby “Visual Traffic Simulation”
- [6] James foxall “Visual basic 2015 in 24 hours”
- [7] CHEPTSOV M.N, TSYKHMISTRO S.I., BOINIK A.B & BAKHAL I.G “NETWORK OF EUROPEAN/ASIAN RAIL RESEARCH CAPACITIES (SIGNALLING SYSTEMS)”
- [8] 佟立本 “Introduction to Railway (Sixth Edition, 铁道概论 (第六版))”