Modelling Railway Interlocking Tables Using Coloured Petri Nets*

Somsak Vanit-Anunchai

School of Telecommunication Engineering
Institute of Engineering, Suranaree University of Technology
Muang, Nakhon Ratchasima 30000, Thailand
somsav@sut.ac.th

Abstract. Interlocking tables are the functional specification defining the routes, on which the passage of the train is allowed. Associated with the route, the states and actions of all related signalling equipment are also specified. This paper formally models the interlocking tables using Coloured Petri Nets (CPN). The CPN model comprises two parts: *Signaling Layout* and *Interlocking Control*. The *Signaling Layout* part is used to simulate the passage of the train. It stores geographic information of the signalling layout in *tokens*. The *Interlocking Control* part models actions of the controller according to the functions specified in the interlocking tables. The arc inscriptions in the model represent the content of the interlocking tables. Following our modelling approach we can reuse the same CPN net structure to model any new or modified interlocking system regardless of its size. Experimental results are presented to provide increased confidence in the model correctness.

Keywords: Control Tables, Railway Signalling Systems, State space analysis, XML, XSLT.

1 Introduction

Background. Currently the State Railway of Thailand (SRT) has been undertaking several railway signalling projects involving either improvement of the existing signalling systems or expansion of the existing railway lines. During the whole process of designing, installing and testing the signalling system, "Interlocking Tables" or "Control Tables" play a vital role. The control table is a tabular representation specifying how the trains move together with the required states and actions of all related equipment. This important document also acts as an agreement between the railway administrators and the contractors. Many signalling contractors have software tools for editing, generating and verifying the control tables. Usually the control table generated by a software tool is bound up with a specific railway company. But SRT has its own operating regulations, requirements and signalling principles that control tables need to comply with. Thus after the control tables are designed and checked by the contractors, they

* Supported by National Research Council of Thailand Grant no. PorKor/2551-153.

D. Clarke and G. Agha (Eds.): COORDINATION 2010, LNCS 6116, pp. 137–151, 2010.

need to be rechecked by SRT's signal engineers. Now SRT signal engineers manually inspect the submitted control tables without any software tools. Thus the checking process is very slow, labour intensive and prone to errors. In order to assist their inspection, detect and rectify errors rapidly, we propose to formally model and analyze the control tables using Coloured Petri Nets (CPNs) [9]. Because SRT's railway signalling project involves hundreds of interlocking systems, we wish to seek out an approach to rapidly build and analyse the CPN model of the control tables especially for a very large interlocking system.

Related Work. In [5], Fokkink and Hollingshead divide the railway signalling system into three layers: infrastructure, interlocking and logistics layers. The infrastructure layer involves objects or equipment used in the yard. The work in this category, for instance [2, 10], ties closely with manufacturer's products. The logistics layer involves human operation and train scheduling which aims at efficiency and absence of deadlocks. It involves the operation of the whole railway network (e.g., [6, 8, 11]) thus the state space explosion problem is often encountered. The interlocking layer provides the interface between logistics and infrastructure layers. It prevents accidents caused by human errors or equipment failure. The work in this category models the interlocking tables and verifies them against the signalling principles. For example [5,14] use theorem provers and [15] uses a model checker to verify interlocking tables.

Hansen [7] presents a VDM model of a railway interlocking system, and validates it through simulation using Meta Language (ML). The work of [7] focuses on the principles and concepts of Danish systems rather than a generic interlocking system. In [7], it is also pointed out that interlocking systems from other countries may be different from the interlocking described in that paper. Winter et al. [13] propose to create two formal models during the design process. One is the formal model of signalling principles called the principle model. The other is the formal model of the functional specification for a specific track-layout called the interlocking model. The control tables are translated into an interlocking model and then checked against the principle model. In [13] CSP (Communicating Sequential Processes) is used as a modelling language but in [15] it is observed that the CSP models of the interlocking system and the signalling principle are difficult to understand and validate. Thus [15] uses ASM (Abstract State Machine) notation to model the semantics of control tables. The ASM model is then automatically transformed to NuSMV code [3] while the safety properties are modeled in CTL (Computational Tree Logic).

Petri Nets, including CPNs, have been used extensively to model railway systems. Most researchers focus on train scheduling and performance measures. Without modelling signalling equipment, [11] uses Interval Timed Coloured Petri Nets (ITCPN) to model train movement through railway stations and analyses throughput and waiting times of trains using the Modified Transition System Reduction Technique (MTSRT). Similar to [11], Hagalisletto et al. [6] use CPNs to model Oslo subway and analyse the train schedule but their refined model includes signalling equipment such as track circuits and points. Durmus and

Soylemez [4] use an extension of Petri Nets, Automation Petri Nets (APN), to design a simple railway yard. The APN model is then translated into a ladder diagram and Code generated for a programable logic controller. Even though [11, 6, 4] and our work use Petri Nets, our application is different from them. [11, 6] are in the logistics layer which aims to analyse the train scheduling. [4] is in the infrastructure layer which involves code generation. Our work is in the interlocking layer which is similar to [1]. Basten [1] simulates and analyses a railway interlocking specification using ExSpect which is a software tool based on high level Petri Nets. However formal verification of railway interlockings is not possible because the interlockings are too complex for the technology at that time.

Choosing Petri Nets. Designing and testing a large railway signalling systems is a complicated tasks involving a lot of details. Our counterpart, SRT signal engineers, suggest to build, maintain and modify the formal models of railway signalling themselves. According to our experience most of the formal techniques previously discussed are too difficult for the signal engineers to comprehend. On the contrary, Coloured Petri Nets (CPNs) provide a graphical notation with hierarchical structuring facilities and the inclusion of a rich set of data types providing a high level of user friendliness. CPNs are also well suited to formalising interlocking tables. In this paper we investigate the feasibility of using CPN Tools by signal engineers.

Contribution. In [12] we modelled and analysed a single track railway station using CPNs. This paper extends that work to consider a more complex double track station. The contribution of this paper is three fold. Firstly, in [12] we developed a static model where the CPN structure reflected the signaling layout of the railway station. In this paper we encode the signalling layout into *tokens*, and automatically generate these *tokens* from the drawing file[1]. This allows our CPN model to easily handle signalling layouts of other stations. Secondly, we propose to standardize the format of control tables using XML and use XSLT to transform the *content* of the control table to ML functions called from the CPN model. By generating the ML functions, we can reuse the CPN net structure for different interlocking tables. As a result, once signal engineers have an understanding of the CPN net structure and methodology, they can apply it to arbitrary railway stations and interlocking systems. Finally, we perform formal analysis, which has so far revealed several errors in the submitted interlocking table from the contractor.

The rest of this paper is organised as follows. Section 2 briefly explains the concept of railway signalling system and control tables. Section 3 defines the scope of work by discussing the assumptions, the modelling approach and the model structure. The CPN model of Panthong control table is explained in Section 4. Section 5 describes our analysis techniques and results. Conclusions and future work are presented in Section 6.

[1] Proprietary software is used to edit the drawing.

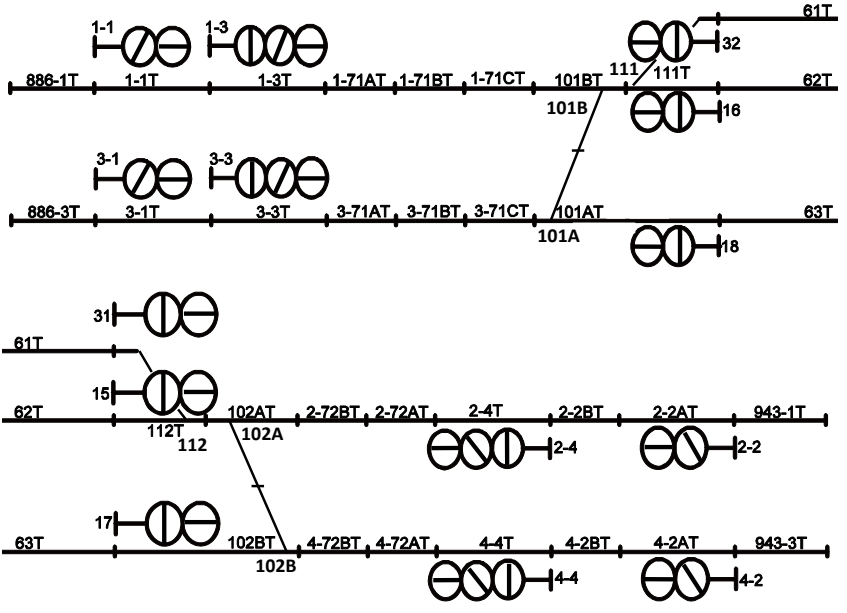

Fig. 1. Signalling layout of the Panthong Station (double track)

2 Railway Signalling Systems and Control Tables

2.1 Signalling Systems

In general the railway lines are divided into *sections*. To avoid collision, only one train is allowed in one *section* at a time. The train can enter or leave a *section* when the driver receives authorization from a signal man via a signal indicator. Before the signal man issues the authorization, he needs to ensure that no object blocks the passage of the train. The *section* between two railway stations, which involves two signal men, is called "*block section*". To prevent human error, which often leads to collisions, the strict operation on a *block section* is controlled by equipment called "Block Instruments". Figure 1 shows the signalling layout of a double track station named "Panthong". The signalling layout comprises a collection of railway tracks and signalling equipment such as track circuits, points and signals. (e.g., signal no.1-3 and signal no.2-4). Each piece of signalling equipment has an identification number and holds a certain state as follows.

Track Circuits. A track circuit is an electrical device used to detect the presence of a train. A track circuit (e.g., 61T, 1-3T) is either *clear* indicating no train on the track or *occupied* indicating the possible presence of a train[2].

[2] When the track circuit fails, its state is occupied even if there is no train.

Warner signals. A warner signal (e.g., 1-1, 2-2, 3-1,4-2) has two aspects: *yellow* or *green*. It informs drivers about the status of the next signal.

Home signals. A home signal (e.g., 1-3, 2-4, 3-3, 4-4) has three aspects: *red*, *yellow* or *green*. It displays *red* when the train is forbidden to enter the *station area*. It displays *yellow* giving the driver the authority to move the train into the *station area* and prepare to stop at the next signal. It displays *green* giving the driver the authority to move the train passing the *station* and enter the next *block section*.

Starter signals. A starter (e.g., 15, 16, 17, 18, 31, 32) has two aspects: *red* or *green*. It displays *red* when forbidding the train to enter the *block section*. It displays *green* when giving the driver the authority to move the train into the *block section*.

Point. A point (e.g., 101A, 101B, 111, 112, 102A, 102B) or railway switch or turnout is a mechanical installation used to guide a train from one track to another. A point usually has a straight through track called "main-line" and a diverging track called loop line. A point is right-hand when a moving train from a joint track diverges to the right of the straight track. Similarly a left-hand point has the diverging track on the opposite side of a right-hand point. When a point diverges the train, it is in reverse position. When a point lets the train move straight through, it is in normal position.

2.2 Control Tables

A collection of track circuits along the reserved *section* is called a "*route*". An entry signal shall be clear to let the train enter the route. Although the request to clear the entry signal is issued by the signal man, the route entry permission is decided by the interlocking system using safety rules and control methods specified in the agreed control tables. Tables 1 and 2 are the (partial) control tables for Panthong station, of which the signalling layout is shown in Fig. 1. Data in the first column, "From", is the route identifications which are labeled by the entry signal: 1-3(1); 1-3(2); 3-3(1); 3-3(2); 3-3(3); 2-4(1); 2-4(2); 4-4(1); 4-4(2); 4-4(3); 15(1); 15(2); 16(1); 16(2); 31(1);31(2); 32(1);32(2); 17 and 18. Due to space limitation we show only 2 routes in Tables 1 and 2. Each row in the tables represents the requirement how to set and release each route. For example, route 1-3(2) comprises the track circuits 1-3T, 1-71AT, 1-71BT,1-71CT,101BT, 111T, 62T, 112T and requires that the points 101, 111 and 112 are in normal position. Routes 1-3(1) and 1-3(2) specify that behind signal 1-3 two routes are possible. Similar rule applies to routes 3-3; 2-4; and 4-4. The column "Requires Route Normal" shows conflict routes. A route cannot be set if any conflict routes have been set and not yet released. For route 1-3(2) the conflict routes are 1-3(1), 16(1), 16(2), 32(1), 32(2), 3-3(1), 3-3(2), 2-4(1), 2-4(2), 4-4(1) and 4-4(2). The exit (starter) signal of this route is 15, and if home signal 1-3 shows green, then starter signal 15 shows green.

Table 1. A control table for Panthong station (part 1:Route locking)

ROUTE		INTERLOCKING			CONTROLS		
		REQUIRES	SET & LOCKS POINTS		ASPECT	SIGNAL AHEAD	REQUIRES TC CLEAR
FROM	TO	ROUTE NORMAL	NORMAL	REVERSE			
1-3(1)	31	16(1), 16(2), 32(1), 32(2), 3-3(1), 3-3(2), 1-3(2), 2-4(1), 2-4(2)	101	111, 112	Y	31 AT R#	1-3T, 1-71AT, 1-71BT, 1-71CT, 101BT, 111T, 61T, 112T
1-3(2)	15	16(1), 16(2), 32(1), 32(2), 3-3(1), 3-3(2), 1-3(1), 2-4(1), 2-4(2) , 4-4(1) , 4-4(2)	101,111, 112		Y G	15 AT R# 15 AT G#	1-3T, 1-71CT, 1-71BT, 1-71AT, 101BT, 111T, 62T, 112T

Table 2. A control table for Panthong station (part 2:Approach locking)

ROUTE		CONTROL						
		APPROACH LOCKED WHEN SIGNAL CLEARED AND		ROUTE RELEASED BY				Notes
				TC	TC OCC	TC OCC	OR EMERCENCY RELEASE AFTER	AND / OR REMARKS
From	TO	TC OCC	OR TIME	CLEAR	& CLEAR			
1-3(1)	31	1-1T	120 sec	1-3T,1-71AT , 1-71BT, 1-71CT, 101BT	111T	61T	240 sec	DOWN BLOCK 1 NOT SET
1-3(2)	15	1-1T	120 sec	1-3T,1-71AT , 1-71BT, 1-71CT, 101BT	111T	62T	240 sec	DOWN BLOCK 1 NOT SET

Different Interlocking systems from different manufacturers may have different control methods. However there are four basic control methods, explained below, which are widely accepted and used among railway companies.

Route locking. Route setting involves a collection of adjacent track circuits, points and signals. A route can be set and reserved for a passage of a train along this route. To assure the safety, firstly, the interlocking system verifies that the route does not conflict with other routes previously set. Secondly, the points along the route are locked in the correct positions. If the related points are not in the correct positions, the controller will attempt to set and lock them in the correct positions. Thirdly, the track circuits along the required route are all clear or unoccupied so that nothing obstructs the passage of the train. Then the entry signal can be cleared (showing yellow or green).

Approach locking. After a route is set; the point is locked; and the entry signal is cleared, if the track circuit in front of (approaching) the entry signal is occupied, then the signal man cannot cancel the route and the entry signal by the normal procedure. Approach locking prevents the train driver from the sudden change of signal aspect from green or yellow to red. Column 3 in Table 2,

"APPROACH LOCKED WHEN SIGNAL CLEARED & TC OCC", presents locking when a route is set and the approach track circuit is occupied. For example, route 1-3(2) will be approach locked if the route is set and track 1-1T is occupied.

Route released. After the passage of the train, the reserved route is released automatically. Column "Route Released by" in Table 2 presents route released mechanism for the signalling layout in Fig. 1. Route 1-3(2) will be released when the track circuits 1-3T, 1-71AT, 1-71BT, 1-71CT, 101BT are clear; the track circuit 111T is occupied and then clear; and the track circuit 62T is occupied.

Flank protection. The equipment within the surrounding area of the reserved route that may cause an accident shall be protected even if no train is expected to pass such a signal or such points. For example points should be in such positions that they do not give immediate access to the route: for example route 1-3(2), the track circuit 61T, which is not in the route 1-3(2), shall be unoccupied; if it is occupied, the object on the track circuit 61T should stand still. This condition is implied when the track 61T is occupied for longer than 1 minute.

3 CPN Model of the Panthong's Control Table - Overview

Coloured Petri Nets (CPNs) [9] are a graphical modelling language for design, verification and analysis of distributed, concurrent and complex systems. CPNs include hierarchical constructs that allow modular specifications to be created. CPN Tools [9] is a software tool used to create, maintain, simulate and analyse CPNs. We use CPN Tools to create and analyse our railway signaling model using state space analysis.

3.1 Modelling Scope and Assumptions

To reduce the complexity of the model as well as avoid the state explosion problem when analysing railway networks [15,6], we need to make the following assumptions regarding train movement and signalling operations:

1. We assume that a train has no length and it occupies one track at a time. The train moves in only one direction. Train shunting is not considered.
2. We assume the trains are running at the same speed.
3. Our model does not include the auxiliary signals such as Call-on, Shunting and Junction indicators.
4. Our model does not include timers. However we use time stamps when modelling the trains moving along the track. This implies that the train must not move through a track circuit so fast that the interlocking cannot detect the presence of the train.
5. Our model does not consider equipment failure.
6. Our model does not include level crossings.

7. Our model includes high level abstraction of block systems but we do not model their operations in detail.
8. Our model does not include flank protections.
9. The train drivers strictly obey the signals.

3.2 Modelling Approach and Model Structure

Signalling layout and control tables are two important documents that are used as references during design and installation of any railway signalling systems. Corresponding to these two documents, in [12] we divided the CPN model into two parts: *Signalling Layout* and *Interlocking Control*.

Signalling Layout. We proposed in [12] to use CPN diagram to mimic the signalling layout so that the train movement can be simulated. Basically our CPN model simulates three kinds of train movements: Train movement between two consecutive track circuits; Trains passing a signal; and Trains passing a point. Despite the fact that the top level CPN model of the signalling layout is easy to read and understand, we encountered two problems while modeled a large station. Firstly, it took about 2-3 days to edit the new CPN model of the signalling layout of a large station. Secondly, where points and signals are located nearby each other, the second level CPN diagrams modelling these equipments are too complex. To solve these two problems, this paper proposes to represent the signalling layout by *tokens* with a complex data structure. Because the geographic information is encoded in the *tokens*, the CPN diagram is not changed when signaling layout is modified or rebuilt. To prevent human error we used C++ to generate a text file containing a list of the *tokens* directly from a drawing file of the signalling layout.

Interlocking Control. The *Interlocking Control* part models point setting, route locking, signal clearing and route release functions as specified in the control table and described in Section 2.2. Unlike [14] that does not include the functionality of approach locking (to avoid the state explosion problem), our CPN model does include the approach locking function. Even though the control table of each railway station has different contents, the functions—router locking, approach locking, route release, and flank protection—are essentially the same. To create a generic interlocking model, we extract the content of the control table and code them into ML functions which are used in arc inscriptions. To model control tables of other railway stations we simply change the content of the ML functions while using the same CPN models of the *Interlocking Control* part.

Next we create these ML functions automatically as illustrated in Fig. 2. In previous projects contractors submitted the control table files in Microsoft-EXCEL format to SRT; we encourage SRT to maintain the control table in XML format instead. As shown in Fig. 2 the control table in Microsoft-EXCEL is transformed to XML. Then it is transformed to ML functions using Extensible Stylesheet Language Transformations (XSLT). All operations are done using Microsoft-Excel and Microsoft-Word version 7.

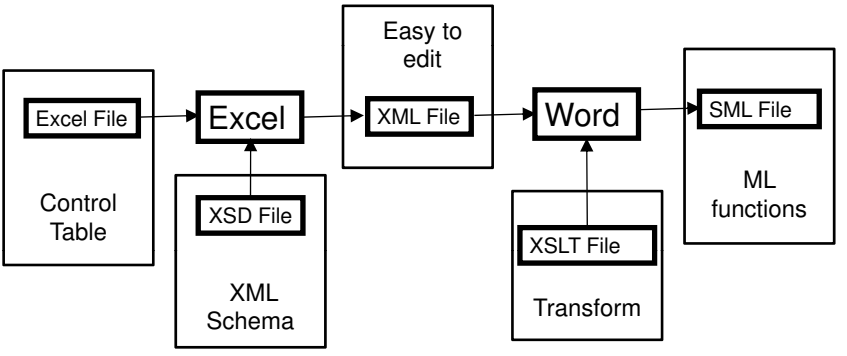

Fig. 2. Transformation of the control table to ML functions using XSLT

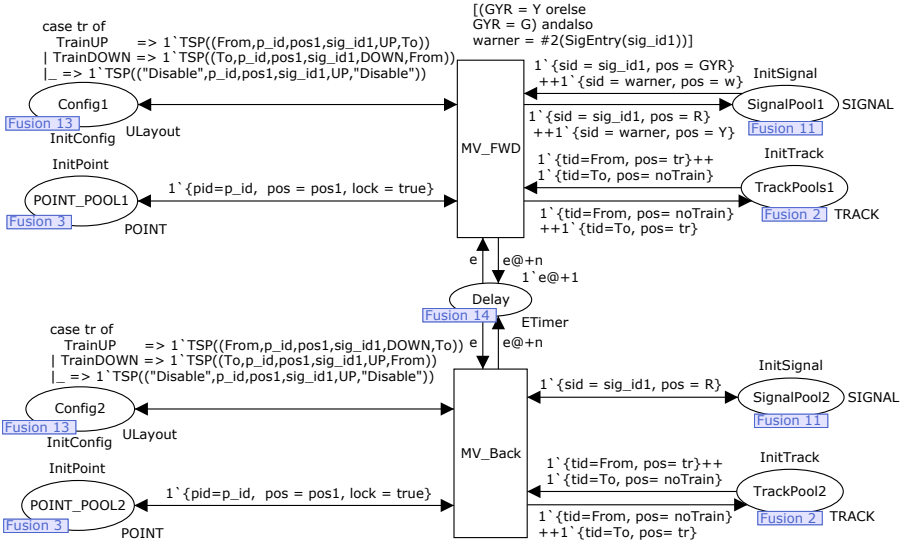

Fig. 3. CPN model: MoveXSignalPoint page

4 The CPN Model of a Signalling System

As discussed in Section 3, our CPN model comprises two parts: the *Signalling Layout* and the *Interlocking Control*. Due to space limitation we explain only the CPN model of *Signalling Layout*. The CPN model of *Signalling Layout* actually models the train movement comprising 6 transitions; and 5 fusion places. Due to space limitation we choose to explain only two transitions shown in Fig. 3. Fusion places `Config1` and `Config2`, typed by `ULayout`, store geographic information of signalling layout in their tokens. Each token basically contains identification numbers of two adjacent track circuits.

Listing 1.1. Declarations

```
1  colset E = with e;
2  colset ETimer = E timed;
3  colset NR = with Normal | Reverse;
4  colset TC_ID = STRING;
5  colset P_ID = STRING;
6  colset SIG_ID = STRING;
7  colset ConfigT2T = product TC_ID*TC_ID;
8  colset ConfigTPT = product TC_ID*P_ID*NR*TC_ID;
9  colset ConfigTST = product TC_ID*SIG_ID*UPDOWN*TC_ID;
10 colset ConfigTSP = product TC_ID*P_ID*NR*SIG_ID
11                           *UPDOWN*TC_ID;
12 colset ULayout = union T2T:ConfigT2T + TST:ConfigTST
13                       +TPT:ConfigTPT + TSP:ConfigTSP;
14 var pos1:NR;
```

Listing 1.2. Declarations

```
1  colset TD = with noTrain | TrainUP | TrainDOWN;
2  var tr:TD;
3  colset TRACK = record tid:STRING * pos:TD;
4  colset SIGNAL = record sid:STRING * pos:SIG;
5  var p_id,tc_id1,tc_id2,sig_id1,sig_id2:STRING;
6  colset ROUTE = STRING;
7  colset ROUTExSIG_ID = product ROUTE * STRING;
8  colset POINT = record pid:STRING * pos:NR * lock:BOOL;
9  var point:POINT;
10 colset BLOCK_POS = with COMING | NORMAL | GOING;
11 var CNG:BLOCK_POS;
12 colset BLOCK = record bid:STRING * pos:BLOCK_POS;
13 var x:BOOL;
```

We classify the tokens into four categories, as follows:

1) Typed by `ConfigT2T` (line 8 of listing 1.1) : one track circuit connects to the adjacent one;

2) Typed by `ConfigTPT` (line 9): a track circuit (either the main line or the loop line) connects to the point track. The position of the point is required to identify which track circuit is connected to the point track;

3) Typed by `ConfigTST` (line 10): a signal is located between two adjacent tracks;

4) Typed by `ConfigTSP` (line 11): a signal is located between a track circuit (either the main line or the loop line) and a point track.

Thus `ULayout` (line 13 of listing 1.1) is defined as the union of the above four colour sets. Actually Fig. 3 is the CPN diagram modelling the fourth category of the train movement. Transition `MV_FWD` models when the train moves facing the signal (e.g., 32) toward the point track (e.g., 111T). Transition `MV_Back` models when the train moves from the point track (e.g., 111T) facing the back of the signal (e.g., 32).

The train movement requires three pieces of information about the state of equipment, namely, the presence of the trains, the signal cleared, and the point locked in a correct position. Three fusion places are used to store these states of equipment: `TrackPool` (typed by `TRACK` - line 4 of listing 1.2); `SignalPool` (typed by `SIGNAL` - line 5); `PointPool` (typed by `POINT` - line 9). `TRACK` is defined

as a record of track identification and train description. `SIGNAL` is defined as a record of signal identification and its aspect (green, yellow or red). `POINT` is defined as record of point identification, its position (Normal or Reverse) and locking status.

5 Analysis

5.1 Desired Property

A basic safety property that railway signalling shall provide is to prevent train collision. In Fig. 3 moving a train requires a token with `noTrain` in the designated track circuit. Each track circuit can contain only one train. Our modelling decision causes two effects. First, two trains in the same track circuit are not allowed. Second, trains cannot move pass each other. We conclude that two trains have a chance of collision if they are on two consecutive track circuits.

To get more confidence about the correctness of our CPN model and the control table, the CPN model is analysed using state space method in CPN Tools. The investigation of the generated state spaces is conducted on a AMD9650 computer with 2.30 GHz and 3.5 GB of RAM. After generating each state space, we use ML query functions searching the entire state space for the markings that have trains in two consecutive track circuits.

5.2 Initial Configurations

Despite the fact that we can analyze various scenarios by changing the initial markings, due to space limitation, we select to discuss only four cases with the initial configurations shown in Table 3. The initial configurations are:

Case A is when four trains are coming from the north and south directions and three trains are on the platform tracks. We set the route request commands for all routes. This is the deadlock case because no train can enter or leave the platform tracks.

Case B is when four trains are coming from the north and south directions and two trains are leaving the platform tracks. We set the route request commands for all routes. Case B-1 and B-2 are similar to Case B but the number of route request commands are fewer in order to reduce the state space sizes.

Table 3. Initial configurations of track circuits and route request commands

Case	886-1T	886-2T	61T	62T	63T	943-1T	943-3T	Route Request Commands
A	TrainUP	TrainUP	TrainUP	TrainDOWN	TrainUP	TrainDOWN	TrainDOWN	All Routes
B	TrainUP	TrainUP	noTrain	TrainDOWN	TrainUP	TrainDOWN	TrainDOWN	All Routes
B-1	TrainUP	TrainUP	noTrain	TrainUP	TrainUP	TrainDOWN	TrainDOWN	All Incoming Routes and 15(1),15(2),31(1),31(2),17
B-2	TrainUP	TrainUP	noTrain	TrainDOWN	TrainDOWN	TrainDOWN	TrainDOWN	All Incoming Routes and 16(1),16(2),32(1),32(2),18

Table 4. Summary of state space results

Case	Nodes	Arcs	Time hh:mm:ss	Terminal Markings
A	36	84	00:01:01	1
B	261,522	1,189,280	11:28:44	57
B-1	9,059	30,954	01:18:24	9
B-2	8,981	27,831	01:17:23	9

Table 5. Terminal Markings of Case B-1.

Route Used 1-3(1)	BlockDown1 GOING	886-1T noTrain	1-1T noTrain	61T TrainUP	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 COMING	886-3T noTrain	3-1T TrainUP		63T TrainUP	4-2BT TrainDOWN	943-3T noTrain	BlockUP4 COMING
Route Used 3-3(1)	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainUP	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 GOING	886-3T noTrain	3-1T noTrain		63T TrainUP	4-2BT TrainDOWN	943-3T noTrain	BlockUP4 COMING
Route Used 2-4(1) 31(1)G	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainDOWN	62T TrainUP	2-2BT noTrain	943-1T noTrain	BlockUP2 GOING
	BlockDown3 COMING	886-3T noTrain	3-1T TrainUP		63T TrainUP	4-2BT TrainDOWN	943-3T noTrain	BlockUP4 COMING
Route Used 4-4(1) 31(2)G	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainDOWN	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 COMING	886-3T noTrain	3-1T TrainUP		63T TrainUP	4-2BT noTrain	943-3T noTrain	BlockUP4 GOING
Route Used 2-4(1) 15(1) 1-3(2)	BlockDown1 GOING	886-1T noTrain	1-1T noTrain	61T TrainDOWN	62T TrainUP	2-2BT noTrain	943-1T TrainUP	BlockUP2 GOING
	BlockDown2 COMING	886-3T noTrain	3-1T TrainUP		63T TrainUP	4-2BT TrainDOWN	943-3T noTrain	BlockUP4 COMING
Route Used 2-4(1) 15(1) 3-3(2)	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainDOWN	62T TrainUP	2-2BT noTrain	943-1T TrainUP	BlockUP2 GOING
	BlockDown3 GOING	886-3T noTrain	3-1T noTrain		63T TrainUP	4-2BT TrainDOWN	943-3T noTrain	BlockUP4 COMING
Route Used 4-4(1) 15(2) 1-3(2)	BlockDown1 GOING	886-1T noTrain	1-1T noTrain	61T TrainDOWN	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 COMING	886-3T noTrain	3-1T TrainUP		63T TrainUP	4-2BT noTrain	943-3T TrainUP	BlockUP4 GOING
Route Used 4-4(1) 15(2) 3-3(2)	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainDOWN	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 GOING	886-3T noTrain	3-1T noTrain		63T TrainUP	4-2BT noTrain	943-3T TrainUP	BlockUP4 GOING
Route Used 4-4(1) 17 3-3(3)	BlockDown1 COMING	886-1T noTrain	1-1T TrainUP	61T TrainDOWN	62T TrainUP	2-2BT TrainDOWN	943-1T noTrain	BlockUP2 COMING
	BlockDown3 GOING	886-3T noTrain	3-1T noTrain		63T TrainUP	4-2BT noTrain	943-3T TrainUP	BlockUP4 GOING

In all initial markings, other track circuits are unoccupied; all points are in `Normal` position and unlocked. All signals are in normal states. Blocks in every directions are initially in the `Incoming` state. One block request command for each outgoing direction is set for the departure train. The block request command cannot be executed unless the block state returns to `Normal`.

5.3 Analysis Results

Tables 4 shows the analysis results: state space sizes; execute time; and the number of terminal markings. Actually we choose to focus on these cases because the number of terminal markings is so few that can be inspected manually. In particular while we were inspecting the terminal markings of case B-1 and B-2, we found several errors in the control table designed by the contractor. These errors were rectified and reported to the contractor. After the errors were rectified, we exhaustively searched the entire state spaces for the train collision condition as discussed in Section 5.1. So far we have not found the train collision in any cases.

A terminal marking in Case A is occurred when the route request commands cannot be executed because required track circuits are not clear. Incoming trains are moved and stopped in the front of the home signals. For Case B even though we are able to manually investigate all 57 terminal markings, we cannot show them here. Due to space limitation we can only show the detail of nine terminal markings of case B-1 (Table 5) and explain only two terminal markings as follows.

a) In the third markings in Table 5, when the first route request command 2-4(1) is set, the train moves from 943-1T to 61T. BlockUP2 is returned to `Normal` and then set to `GOING`. The second request command 31(1) is set for the train on 61T going toward north but the train on 61T (`TrainDOWN`) plans to go toward south instead.

b) In the last markings in Table 5, when the first route request command 4-4(1) is set, the train moves from 943-3T to 61T. BlockUP4 is returned to `Normal` and then set to `GOING`. The second train moves from 63T to 943-3T via route 17 and the third train moves from 886-3T to 63T via route 3-3(3). BlockDOWN3 is returned to `Normal` and then set to `GOING`.

6 Conclusions

This paper has outlined an approach for developing a CPN model of SRT's railway signalling system. The CPN model comprises two parts: *Signalling Layout* and *Interlocking*. Geographic information how each piece of equipment connects to each other is stored in the *tokens*. Thus the CPN net structure of the *Signalling Layout* part does not depend on the signalling plan. Similarly the *Interlocking* part does not depend on the signalling plan as well. It has the contents of the control tables encoded in the ML functions. Thus we can use the same net structure to model any interlocking systems regardless of the size of the interlocking. We also discuss the analysis results to demonstrate the applicability of our approach. Despite prior expectations, several errors in the control tables were discovered during analysis.

There are two lines of future work we would like to pursue. Firstly, we had encountered the state space explosion problem while we were attempting to verify the interlocking table of a large station. Thus we wish to seek out a systematic approach to tackle this problem. Secondly, we would like to relax the modelling assumptions and refine the model.

Acknowledgments. The author is thankful to the anonymous reviewers and also to MohammadReza Mousavi and Steve Gordon. Their constructive feedback has helped the author improve the quality of this paper.

References

1. Basten, T., Bol, R., Voorhoeve, M.: Simulating and Analyzing Railway Interlockings in ExSpec. IEEE Parallel and Distributed Technology, Systems and Applications 3(3), 50–62 (1995)
2. Chevilat, C., Carrington, D., Strooper, P., Süß, J.G., Wildman, L.: Model-Based Generation of Interlocking Controller Software from Control Tables. In: Schieferdecker, I., Hartman, A. (eds.) ECMDA-FA 2008. LNCS, vol. 5095, pp. 349–360. Springer, Heidelberg (2008)
3. Cimatti, A., Clarke, E.E., Giunchiglia, F., Roveri, M.: NuSMV: A new symbolic model verifier. In: Halbwachs, N., Peled, D.A. (eds.) CAV 1999. LNCS, vol. 1633, pp. 495–499. Springer, Heidelberg (1999)
4. Durmus, M.S., Soylemez, M.T.: Railway Signalization and Interlocking Design via Automation Petri Nets. In: Proceedings of the 7th Asian Control Conference, Hong Kong, August 27-29, 2009, pp. 1558–1563 (2009)
5. Fokkink, W.J., Hollingshead, P.R.: Verification of Interlockings: from Control Tables to Ladder Logic Diagrams. In: Proceedings of the 3rd Workshop on Formal Methods for Industrial Critical Systems (FMICS 1998), Amsterdam, May 1998, pp. 171–185. Stichting Mathematisch Centrum (1998)
6. Hagalisletto, A.M., Bjørk, J., Yu, I.C., Enger, P.: Constructing and Refining Large-Scale Railway Models Represented by Petri Nets. IEEE Transactions on Systems, Man, and Cybernetics, Part C 37(4), 444–460 (2007)
7. Hansen, K.M.: Formalizing Railway Interlocking Systems. In: Nordic Seminar on Dependable Computing Systems, Department of Computer Science, Technical University of Denmark, pp. 83–94 (1994)
8. Janczura, C.W.: Modelling and Analysis of Railway Network Control Logic using Coloured Petri Nets. PhD thesis, School of Mathematics and Institute for Telecommunications Research, University of South Australia, Adelaide, Australia (August 1998)
9. Jensen, K., Kristensen, L.M., Wells, L.: Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. International Journal on Software Tools for Technology Transfer 9(3-4), 213–254 (2007)
10. Svendsen, A., Olsen, G.K., Endresen, J., Moen, T., Carlson, E., Alme, K., Haugen, Ø.: The Future of Train Signaling. In: Czarnecki, K., Ober, I., Bruel, J.-M., Uhl, A., Völter, M. (eds.) MODELS 2008. LNCS, vol. 5301, pp. 128–142. Springer, Heidelberg (2008)
11. van der Aalst, W.M.P., Odijk, M.A.: Analysis of Railway Stations by Means of Interval Timed Coloured Petri Nets. Real-Time Systems 9(3), 1–23 (1995)
12. Vanit-Anunchai, S.: Verification of Railway Interlocking Tables using Coloured Petri Nets. In: The Tenth Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, DAIMI PB 590, Department of Computer Science, University of Aarhus, October 19-21, pp. 139–158 (2009)

13. Winter, K.: Model Checking Railway Interlocking Systems. In: Oudshoorn, M. (ed.) Proceeding of the 25th Australasian Computer Science Conference (ACSC 2002), Melbourne, Australia, vol. 4, pp. 303–310. Australian Computer Society (2002)
14. Winter, K., Johnston, W., Robinson, P., Strooper, P., van den Berg, L.: Tool Support for Checking Railway Interlocking Designs. In: Cant, T. (ed.) Proceeding of the 10th Australian Workshop on Safety Related Programmable Systems (SCS 2005), Sydney, Australia, vol. 55, pp. 101–107. Australian Computer Society (2005)
15. Winter, K., Robinson, N.: Modelling Large Railway Interlockings and Model Checking Small Ones. In: Oudshoorn, M. (ed.) Proceeding of the 26th Australasian Cumputer Science Conference (ACSC 2003), Adelaide, Australia, vol. 16, pp. 309–316. Australian Computer Society (2003)