GU J Sci 30(2): 133-147 (2017)

Gazi University

Journal of Science

http://dergipark.gov.tr/gujs

An Automatic Formal Model Generation and Verification Method for Railway Interlocking Systems

Muhammed Ali OZ[1,*], Ozgur Turay KAYMAKCI[1]

[1]*Department of Control and Automation Engineering, Faculty of Electrical and Electronics Engineering, Yildiz Technical University, Esenler Istanbul, Turkey*

Article Info

Received: 06/09/2016
Revised: 02/04/2017
Accepted: 30/04/2017

Keywords

Formal verification
Interlocking
Timed arc Petri nets
Railway signalization systems

Abstract

Railway transportation systems incorporate many safety critical systems such as signalization systems. Any possible failure within the scope of these safety critical systems can seriously harm the environment and lead to many life losses. Therefore design, development and the implementation of these sector specific products have been raised to a certain quality with the guidance of sector specific standards like EN 50126. Electronic interlocking system is one of the most important and essential product in railway transportation systems such that it controls railway traffic operation securely and prevents trains from colliding and also derailing. In this context, the developed algorithm must be automatically verified in order to ensure that the system will work totally reliable. In this paper, a new methodology using timed arc Petri nets is introduced in order to formally validate and verify railway interlocking system's correctness and safety. Also in order to reduce the human effort and possible implementation errors, a new software is developed using the programming language C#. The developed program automatically generates the formal models of the interlocking system through a visual interface. Here the safety requirements, which are written in CTL formulation, are verified on TAPAAL. Finally the obtained algorithm and models are implemented on an operational railway station by developed software in order to show the introduced method's effectiveness, accuracy and swiftness.

1. INTRODUCTION

Railway transportation systems gain more and more importance in recent years due to the increasing urbanization. This growing demand also requests a lot of new performance functionalities by time so a lot of new products enter the market and the existing ones are improved in the line of new requirements. In this context European Union (EU) provides a great deal of funding for the rail sector through various programs [1]. Also many projects were funded from a variety of sources in Europe in order to reach the goals of the European Railway Traffic Management System (ERTMS). One of the big challenges in these kind of projects is that the signalization system requires a predefined safety integrity level. In this regard, European Committee for Electrotechnical Standardization (CENELEC) highly recommends the use of formal methods given EN 50126:2001 Table A.16 [2].

Formal methods are mathematical based techniques used to model complex systems. As systems become more complicated, and safety becomes a more crucial issue, the formal modelling approach offers better solutions. While uses of these methods improve system reliability, design time and comprehensibility, they also enable us to verify the system's properties. Furthermore, EN 50128 strongly recommends the utilization of formal methods in the modelling and verification of railway signalization and interlocking systems.

In the literature, there are numerous studies about the interlocking and signalization system design by using formal methods. S. A. Khan et al integrated mobile agent concepts with Petri nets to develop the mobile Petri net (MPN), which supports both mobility and concurrency, and used MPN to model the safety properties of moving block interlocking system along the switch and level crossing [3]. H. Wang et al proposed an innovative topology-based method which addresses the problems of having to rely too much

*Corresponding author, e-mail: maoz@yildiz.edu.tr

on designers' experience and of incurring excessive cost of validation and verification in the development of railway train control systems [4]. A. E. Haxthausen et al explained the techniques for application of bounded model checking, and discusses their advantages in comparison to the alternative approaches [5]. Robert Abo and Laurent Voisin described the process of data validation using Ovada formal tool for railway safety-critical computer-based systems implemented by Systerel [6]. P. James et al proposed a formal method which is a verification step between programming the interlocking and the testing of this program [7]. B. Malakar and B.K. Roy used automation Petri nets (APNs), which is an extension of classical Petri-Nets, to model a railway interlocking and signalization system for a sample railway yard [8]. Pengfei Sun et al. introduces a modelling pattern of the French railway interlocking system, which is a parameterized model respects the French national rules [9]. Xi Wang, Tao Tang and Shuo Liu [10] introduces a model-based methodology for development of CBTC (Communications Based Train Control) interlocking system and they verified the correctness and safety of this interlocking system by means of Prover plug in model checker, which is integrated in SCADE. The authors formed formal models based on safety requirements for a point automation system, which is one part of a signalization system, by using TAPN (Timed Arc Petri Nets) [11]. The introduced algorithms in [11] were implemented in a software tool in order to automatically generate the TAPN models but the verification process is not introduced [12, 13]. Furthermore research on formal approaches to safety verification of railway interlocking system can be found in [14-19]. There are also limited number of studies about the automatic generation and verification of interlocking models. In a study, an algorithm for automatic interlocking table generation and its implementation is introduced. The specially developed software generates an interlocking table by translating a railway topology into an interlocking table [20-21]. However, the obtained interlocking tables in these studies were not verified. Yan Cao et al. introduce a toolset based on Domain Specific Language for Computer Based Interlocking Systems to automatically generate and verify the interlocking table [22].

In this study, the interlocking system was designed automatically by using TAPN, which is highly recommended by EN 50128. The reasons for selecting TAPN are briefly as follows. As it is known that, Petri nets can represent a larger class of languages than the class of regular languages so it provides a possibility to model a wider system family when compared to deterministic finite state automatons. On the other hand there are too many different type of high level Petri nets like colored Petri nets, automation Petri nets, prioritized Petri nets and so on. The most important reason why TAPN is selected among different high level Petri nets is that the time can be added to the model as a parameter and a more realistic modeling dynamics can be acquired. Besides it should be stated that in general the complexity of the models increase when the other high level Petri nets are selected as the modelling tool. In reality more complex models mean more verification and validation times.

In this work, a new software tool was developed by using C# programming language to automatically generate TAPN models of interlocking systems such that these models can be viewed through TAPAAL, which is a tool for editing, simulating, analyzing and verifying TAPN. The verification process is done on the basis of certain safety requirements, which are written in CTL formulation. This study furthers our previous research with simpler and easily programmable TAPN models with lower generation and verification times. Also more complex single line systems are modeled contrary to double track lines which were modeled in our previous research. A modular modelling technique, which is also easy to design and implement into other system, is the main contribution of this paper. Furthermore this paper provides a method on how to automatically produce models using the proposed technique. The verification time of the model is also much shorted and almost instant when compared to other methods in the literature such that the most algorithms and models' verification times ascend exponentially in increasing system complexity.

2. TIMED ARC PETRI NETS

TAPN is defined with a 7-tuple (P, T, IA, OA, Transport, Inhib, Inv) such that P ={P1, P2, P3, …, Pn} is a finite set of places, T={T1, T2, T3, …, Tn} is a finite set of transitions, $IA \subseteq P \times \tau \times T$ is a finite set of input arcs with τ representing the time, $OA \subseteq T \times P$ is a finite set of output arcs, Transport : IA x OA →{true, false} is a function defining transport arcs which are pairs of input and output arcs connected to some transition, Inhib : IA→{true, false} is a function defining inhibitor arcs which do not collide with transport arcs and finally Inv : P→ τ inv is a function assigning age invariants to places.

Here the timed-arc Petri net is an extension of the classic Petri nets. The enabling rule of a TAPN is a little bit different from the classical Petri nets. $T_i \in T$ is enabled if for all input arcs except the inhibitor arcs, there is a token in the input place of the arc with an age satisfying the time interval of the arc. In other words, it is enabled when the age of the tokens in its all input places have reached its time intervals. Also, $T_i \in T$ is enabled if there is no token in the input place of the arc with an age satisfying the time interval of the arc for all inhibitor arcs. A transition $T_i \in T$ may be fired if it is enabled by the marking of its input places and if all time restriction related to its time interval are satisfied. Also for detailed information refer to [24, 25, 26].

3. RAILWAY INTERLOCKING SYSTEM

The safety of railway transportation depends especially on the interlocking system, which is a crucial subsystem of the signalization system. The main purpose of the interlocking system is to guarantee the safety of the vehicles. In this context, the commands coming from TCC (Traffic Control Center) are evaluated and decided according to the states of field equipment like points, signals and track circuits and safe and convenient decisions are taken. The main task of the interlocking system is relevant to route. Generally, in order to set a route for an incoming train, none of the track circuits should be occupied by any railway vehicle, all points on the desired route must be in appropriate position and locked and also all conflicting signals and opposing signals should be locked. Thus, other routes that conflict with the desired route are prohibited to ensure safety and security. After the vehicle completes the route, prohibited railway tracks become available and all the electronic locks are released. In this perspective, interlocking system consists of three parts; points, signals and track circuits.

3.1. Point

A railway point is a mechanical installation enabling trains to maneuver to right or left at a railway junction. When a route is desired, the corresponding points on this desired route are moved laterally to their desired position by the interlocking system itself.

3.2. Signal

Signals are systems that transmit colored light notice, notifying the trains if there is a train on the proceeding tracks and if the train will travel upon diverging points to ensure trains are moving at safe speeds. It is crucial to use signals since the brake distance of railway transportation vehicles is more than that of other transportation vehicles.

3.3. Track Circuits

The positions of the trains have to be known at all times in order to guide the railway traffic in a safe manner. For fixed block railway signalization systems, track circuit is the most frequently used component to detect the absence of a train on the track section.

4. FORMAL MODELING OF INTERLOCKING SYSTEM

Interlocking system in this study was modelled using TAPN, one of the formal modelling methods based on CENELEC EN 50128 standard (Table A.17-Modeling), which was also highly recommended to be used by the relevant standard. The use of TAPN allows us to carry out rigorous analysis and examine the system symbolically. In addition, it is possible to transfer temporal acts into the model better and so more powerful models are obtained. It also enables us to model and design the system on modular basis, which is also recommended by CENELEC EN 50128.

Here in this study, point and signal TAPN models were formed separately. The interaction between the block sections are introduced by a field topology model and the safe movement of a vehicle from signal to signal is also introduced with a route model.

4.1. The TAPN model of a point

The main functionality of the point is modelled by TAPN model with seven places such that these are P={Point_Normal, Region_Busy, Point_Reverse, Point_Move_Normal, Point_ Move _Reverse, RtoN, NtoR}. Here the places respectively indicate the following situations: point is in normal position, occupation of point block section, point is in reverse position, move point to normal position, move point to reverse position, transition state while point transitions from reverse to normal, transition state while point transitions from normal to reverse. If there is a token on “Region_Busy” place, this means that there is a train on the point track section and the point should not move. The corresponding timed-arc petri net model is given at Fig. 1.

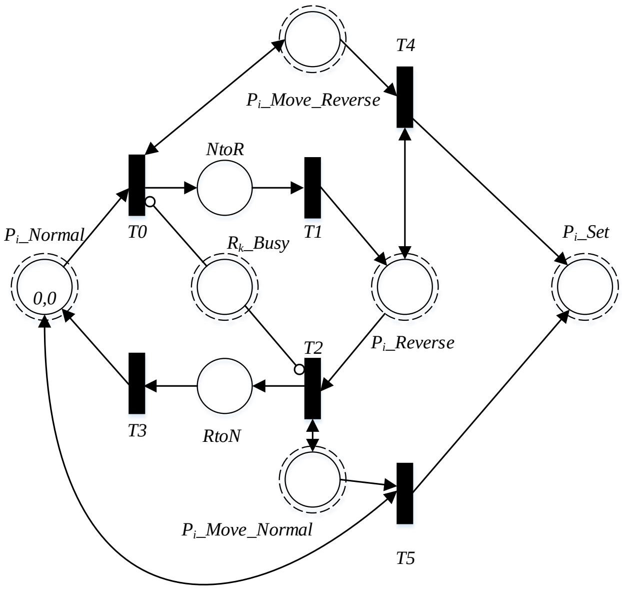

Figure 1. *Point TAPN model*

4.2. The TAPN model of a signal

The general operating principles of the signal model given in Fig. 2 are as follows. The model includes 8 places. These are “Signal_Enable”, “Signal_R”, “Signal_G”, “Signal_Y”, “Signal_GOY”, “Signal_YOY”, “Signal_ROY”, and “System_Idle” places These places express “the signal is enabled”, “the signal indicates red”, “the signal indicates green”, “the signal indicates yellow”, “the signal indicates green and yellow”, “the signal indicates yellow and yellow”, “the signal indicates red and yellow” and “signal is not set” situations respectively. If a diverging point is present on the route, Signal_GOY or Signal_YOY positions will receive a token depending on the indicator of the next signal models. The second yellow signal indicates that the route includes a diverging track section.

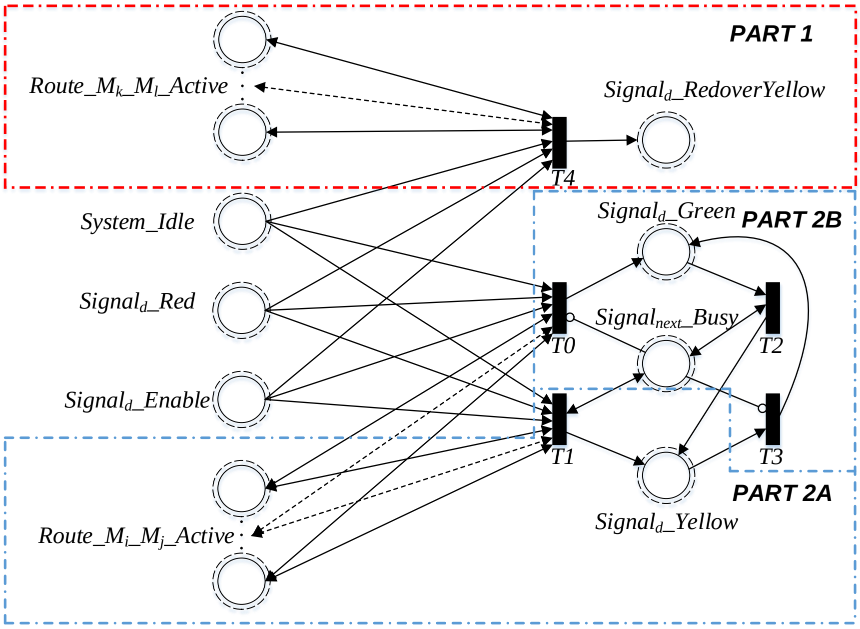

Figure 2. *Non-diverging TAPN model of Signal*

Diverging TAPN model part of signal can be seen in Fig. 3. Lastly if the train is moving to a signalless track section Signal_ROY position will receive a token. System_Idle position is to prevent setting a route before the signal is set. The below signal model is produced for a 4- aspect signal. For a 3- aspect signal a similar model can be easily used but “Signal_GOY”, “Signal_YOY” and “Signal_ROY” places are not present in that model. The relevant timed-arc petri net model formed can be seen at Fig. 2.

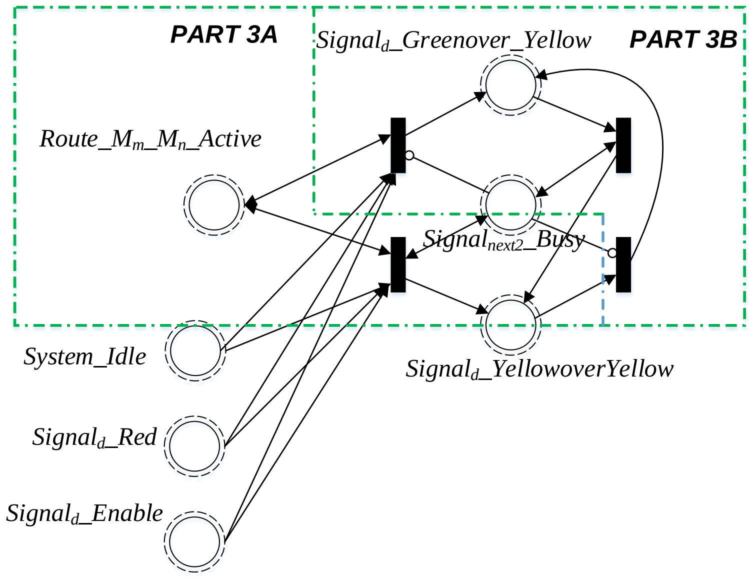

Figure 3. *Diverging Part of the Signal model*

4.3. Route TAPN model

Depending on the need routes, which allow trains to safely access certain block sections, are opened. The route model manages these routes and is composed of intersecting routes and 7 places which are P={System_Idle, Signal_Enable, Point_Move_Normal or Point_Move_Reverse, Point_Set, P0, P1}. When a request to open a route is received if no intersecting routes are open the points along the route are set to their desired positions immediately after the color codes of the signals along the route are set accordingly. With these preparations the route is open and the train can travel safely along the route. The relevant Timed-Arc Petri Net model formed can be seen at Fig. 4.

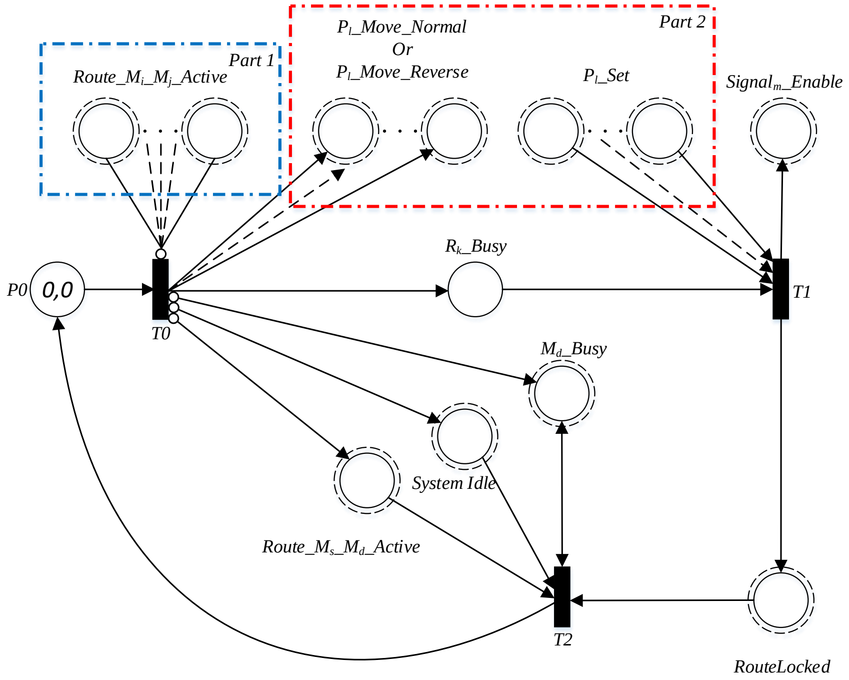

Figure 4. *Diverging Part of the Signal model*

4.4. Field TAPN model

Field model simulates the movement of the trains inside the station. In each field model there is a starting block section and block sections that can be reached from the starting block section. For a detailed representation color codes of the signals, positions of the points and the track circuit which detects the presence of trains are added to this model. The relevant Timed-Arc Petri Net model formed can be seen at Fig. 4.

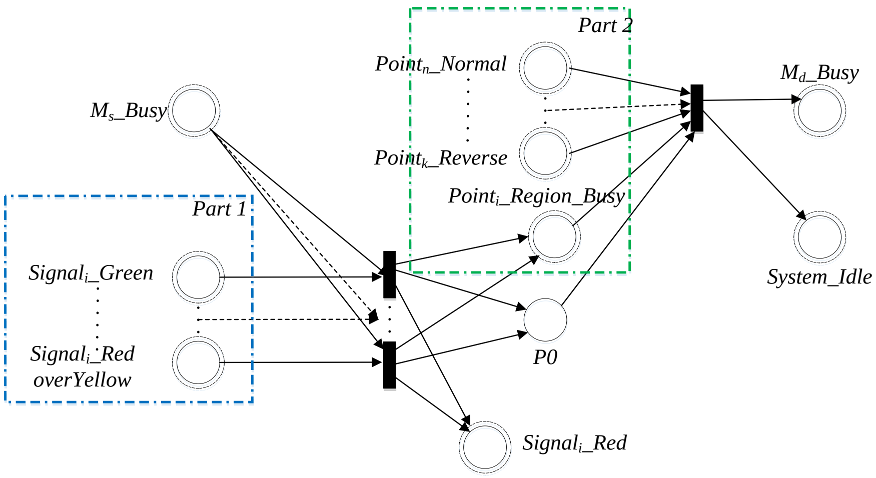

Figure 5. *Field TAPN model*

5. AUTOMATIC INTERLOCKING FORMAL MODEL GENERATION

Automatic formal model generation for interlocking system was performed by a software tool. A graphical user interface provides a simple way for users to draw station topology and eliminates the need for a second tool. Fig. 6 shows the interface with an example station topology.

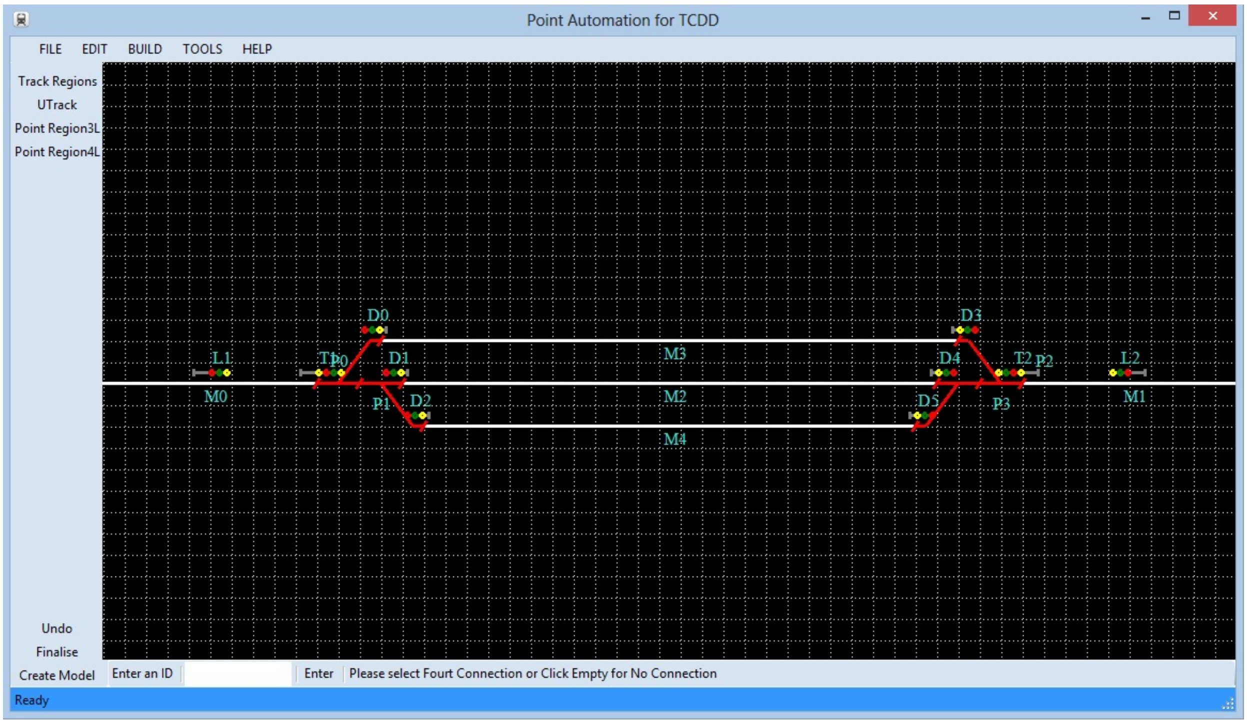

Figure 6. *Snapshot of GUI*

Software part receives the model created inside the user interface and transforms this received messy data into meaningful matrices. A matrix is created for blocks, points and signals containing information on them and information on their connections to each other. Using these matrices another matrix, which contains information on routes, is created as seen at Fig. 7.

As the routes are set from signal to another signal, possible routes can easily be discovered using connection data of each component. The given algorithm in Fig. 7 starts with a block section and reaches to another block section which is the same as starting from a signal and ending at another signal.

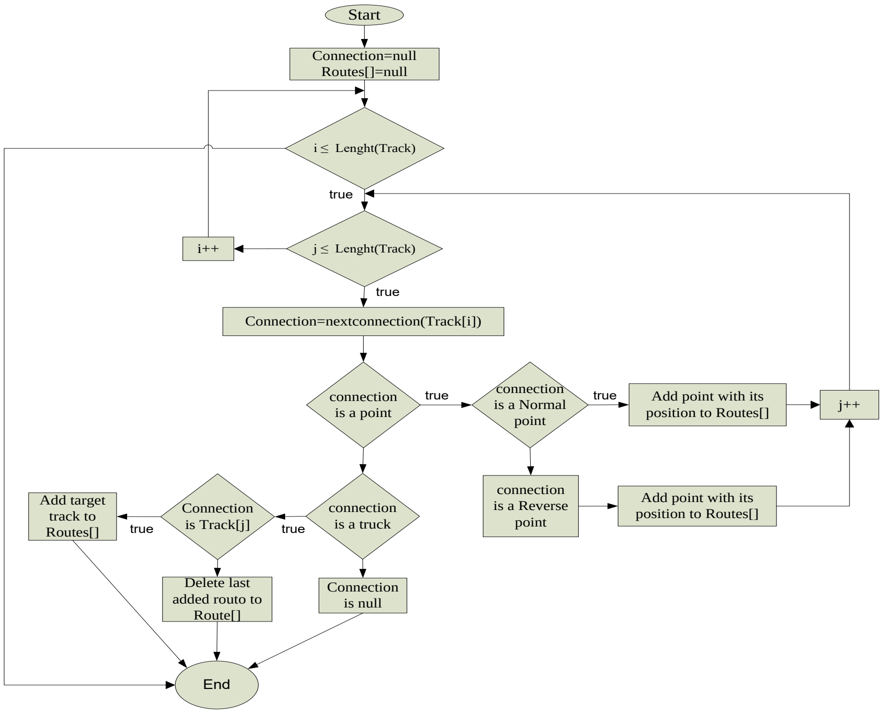

Figure 7. *The Flowchart in order to find all possible routes*

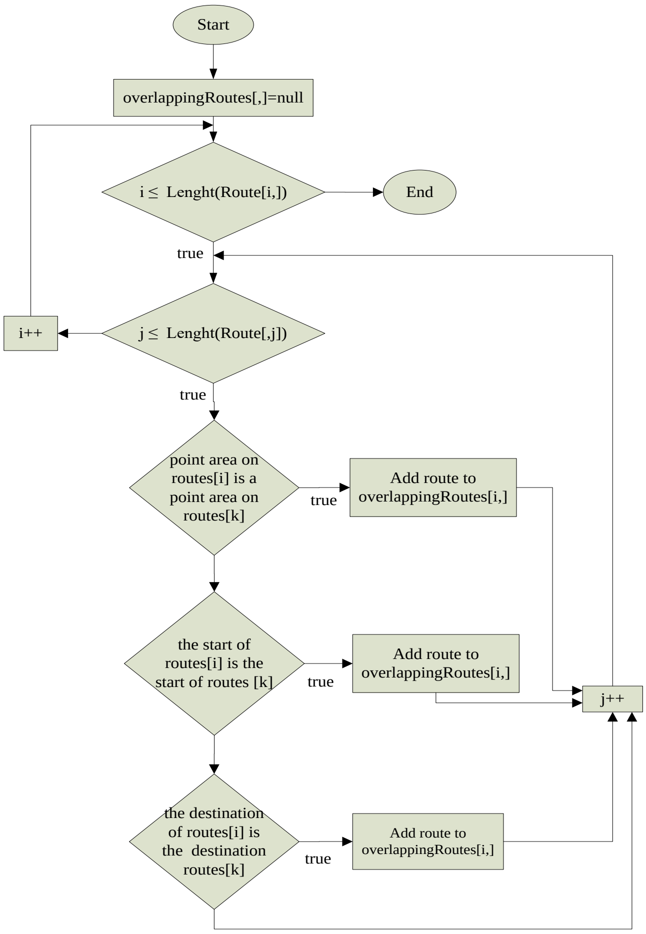

Figure 8. *Conflicting routes flowchart*

Another main constraint comes from the block section topology is the conflicting route set. If two conflicting routes are activated by the interlocking system, a possible collusion can occur. So the conflicting routes are determined within the algorithm given in Fig. 8. Using the sorted data obtained using the algorithms shown above TAPN models can be generated.

Here the generated point TAPN models are same for every point, only the id's of points are changed. The point models for each point are created according to the model given in Fig. 1. The Signal TAPN models are created according to its location on the topology and the introduced signal model in Fig. 2. The corresponding algorithm is given at Fig. 9.

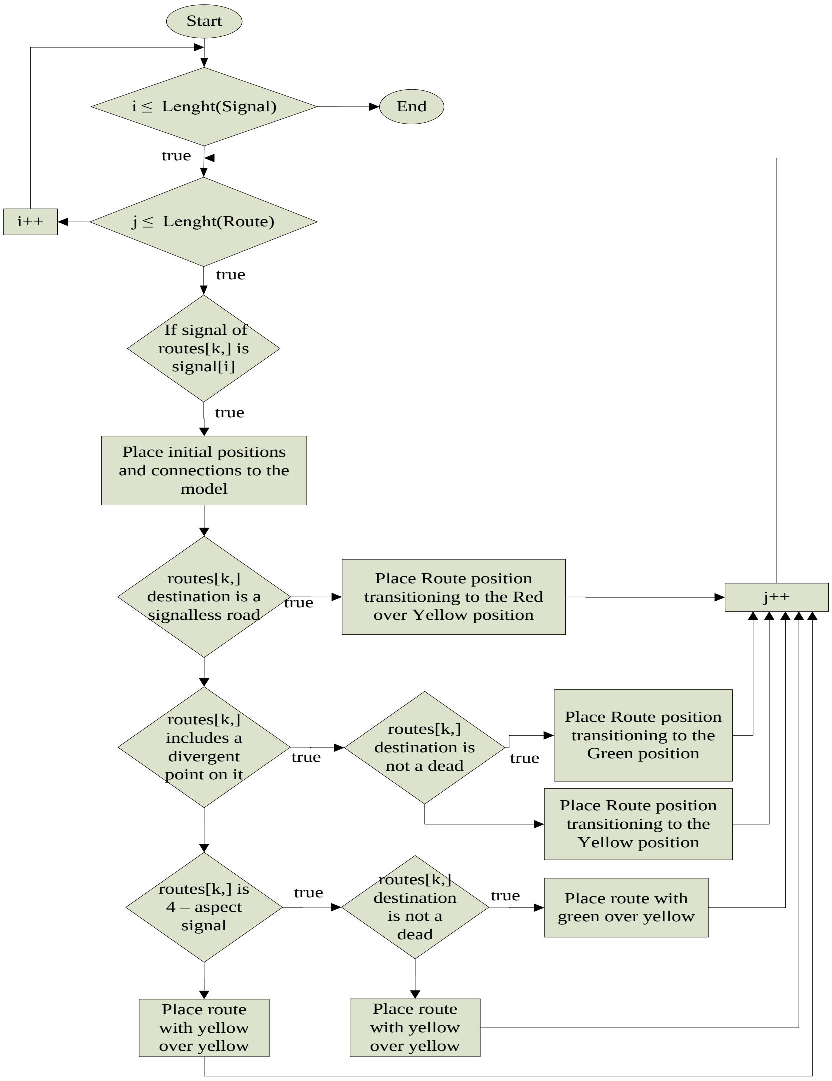

Figure 9. *Signal TAPN model creating flowchart*

Route model for every route possible is created using the route matrix. There are two important issues in forming route model. The first one, conflicting routes should never be opened at the same time. The second one, points should be in correct position on desired route. Route TAPN model can be created according to the algorithm given at Fig. 10a. Topology algorithm deals with the interaction of block sections. The vehicles move with the help of the signal indicator such that these indicators are also added to the model along with point block sections that the vehicle travels. This algorithm is also shown at Fig. 10b.

Once a model is created, the queries are added to the model by using the developed software and an xml file is created. This file can be opened and modified with the help of TAPAAL [25].

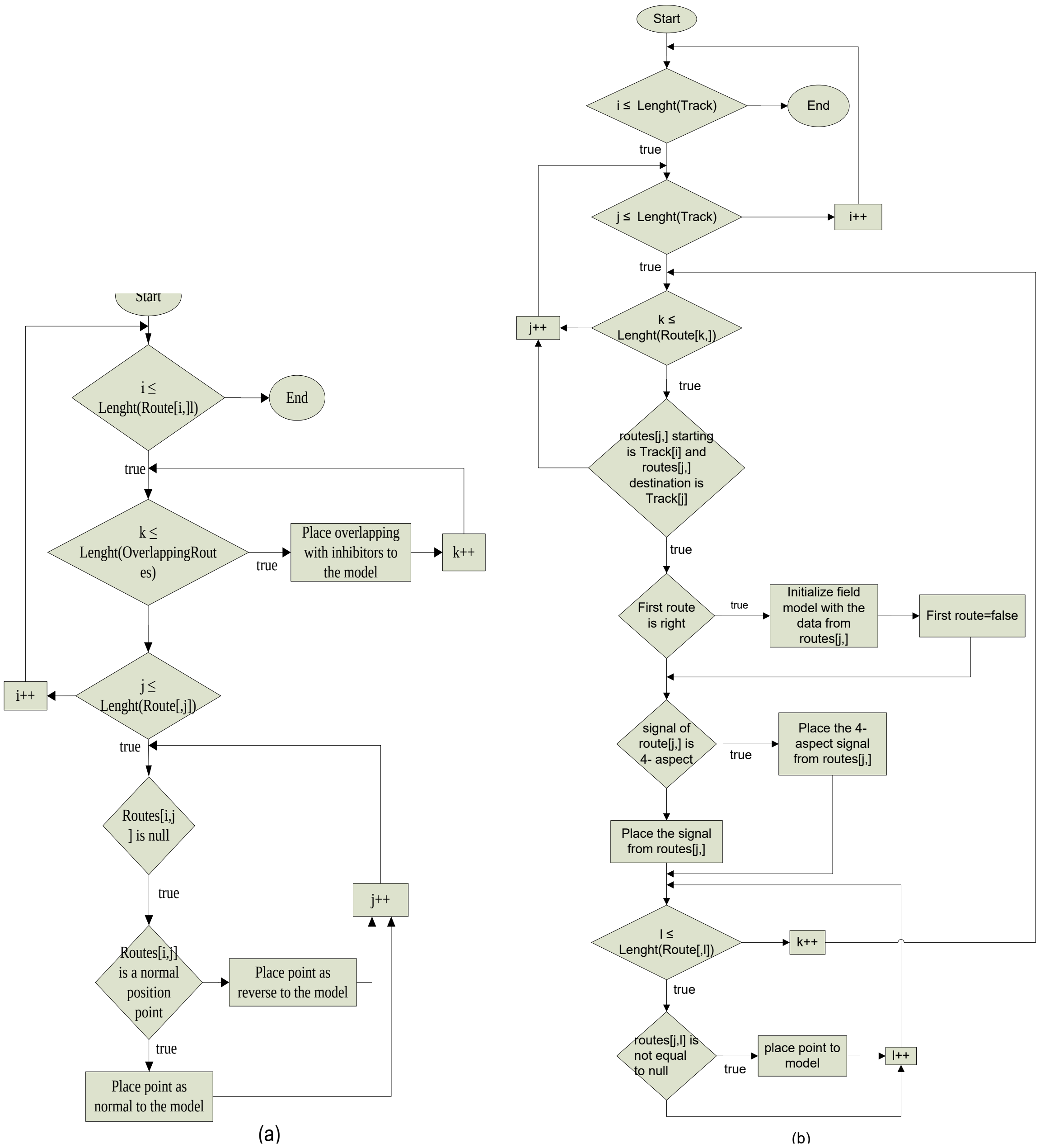

Figure 10. *(a) Route TAPN model creating flowchart* **-** *(b) Topology TAPN model creating flowchar*

6. VERIFICATION OF FORMAL MODELS FOR RAILWAY INTERLOCKING SYSTEM

After the models are created, the adequacy of the model is verified by introducing the safety requirements written in computational tree logic [27]. These safety queries are added to route and field models in automatic modelling process. All queries are checked via TAPAAL discrete verification method based on the breadth first search order in state space.

6.1. Saray station

Saray station, which is operated by Turkish Railway Company, is chosen in order to check out the correctness of the introduced models and algorithms. This station has 4 points P={P0, P1, P2, P3}, 5 track circuits TC={M0, M1, M2, M3, M4} and as well as 10 signals S={L0, L1, T0, T1, D0, D1, D2, D3, D4, D5}. The topology of Saray station is given in Fig. 11.

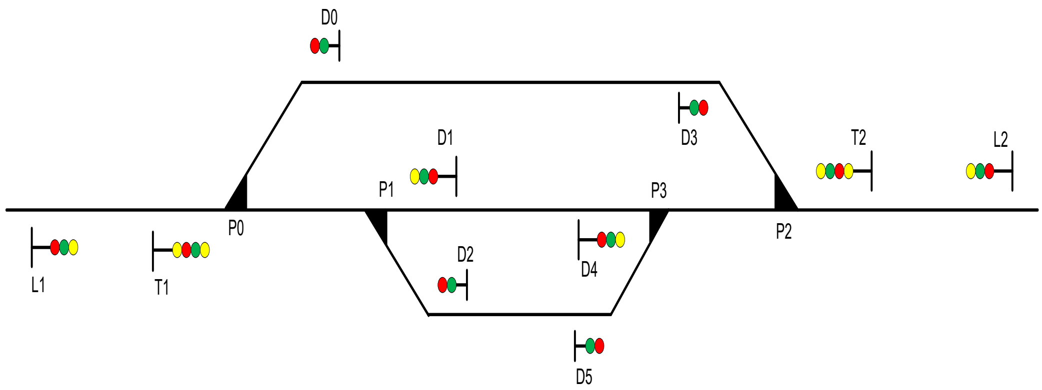

Figure 11. *The topology of TCDD Saray station*

Conflicting routes and occupied block sections must be checked before opening routes to ensure safety. Points that are on the route are translated to their desired positions and the respective signal is set before the route is set. During this process system idle get a token and disables any parallel process until the route is locked. A route TAPN model which created by the developed software is given below in Fig. 12.

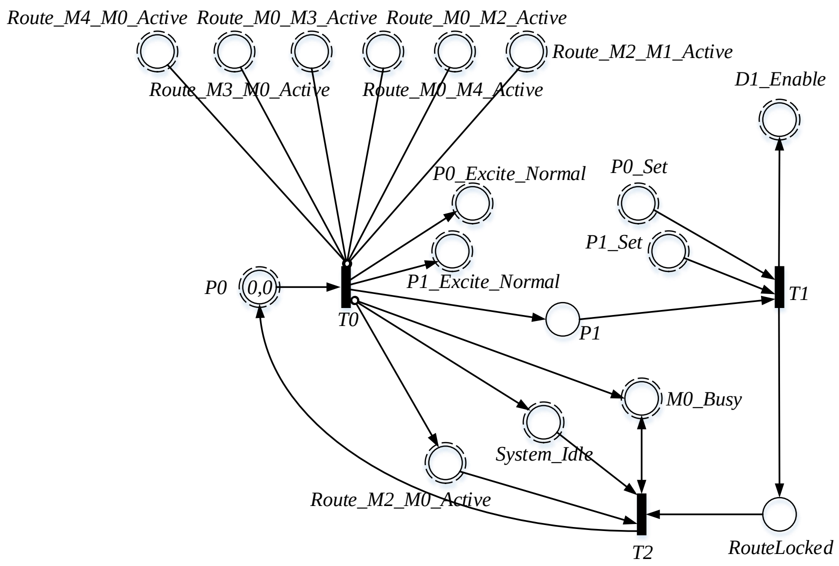

Figure 12. *Route TAPN model from M2 track to M0 track*

Before a route is active, its respective signal must be set. Signals are set to green or yellow depending on the indicator of the next signal only if a signal is a 4-aspect signal then if the route is diverging signal indicator is set as yellow over yellow or green over yellow. A red over yellow indicator is used when the destination is a non-signaled track.

Topology TAPN models are used to simulate a train's movement once a signal is set. The topology TAPN model generated using our software is given in Fig. 13.

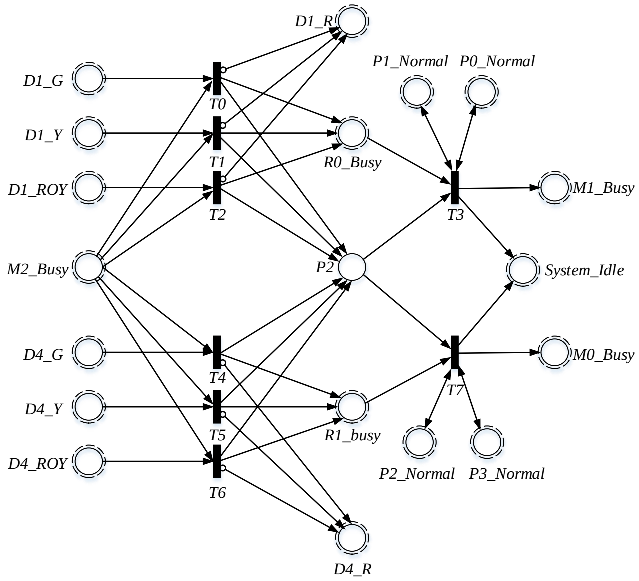

Figure 13. *Field TAPN model from M2 track*

6.2. Verification results for Saray station

Verification process is made based on identified safety requirements(SRs). The some of them can be listed as follows:

SR1: All routes that conflict with the desired route must be prohibited and must never be set. Table 1 shows the analysis of SR1. SR1 is written in CTL formulation as follow:

$$AG\neg\left(RouteM_0\_M_2 \geq 1 \wedge \left[\left(RouteM_1\_M_2 \geq 1\right) \vee \left(RouteM_1\_M_2 \geq 1\right) \vee \ldots\right]\right)$$

Table 1. *Analysis of SR1*

Query	Result	Verification time
Route_M0_M2_SR1	Satisfied	0.016 s
Route_M0_M3_SR1	Satisfied	0.016 s
Route_M0_M4_SR1	Satisfied	0.015 s
Route_M1_M2_SR1	Satisfied	0.015 s
Route_M1_M3_SR1	Satisfied	0.016 s
Route_M1_M4_SR1	Satisfied	0.016 s
Route_M2_M0_SR1	Satisfied	0.016 s
Route_M2_M1_SR1	Satisfied	0.016 s
Route_M3_M0_SR1	Satisfied	0.016 s
Route_M3_M0_SR1	Satisfied	0.016 s
Route_M3_M1_SR1	Satisfied	0.016 s
Route_M4_M0_SR1	Satisfied	0.016 s

SR2: The point must never move, while the train occupies the point. Namely, it should not get any command

for changing of position, while the train passes over the point. Table 2 shows the analysis of SR2. SR2 is written in CTL formulation as follow:

$$AG\neg\left(\left(R_k\_Busy \geq 1\right) \wedge \left(\left(P_k.NtoR \geq 1\right) \vee \left(P_k.RtoN \geq 1\right)\right)\right)$$

Table 2. *Analysis of SR2*

Query	Result	Verification time
Point0_SR2	Satisfied	0.017 s
Point1_SR2	Satisfied	0.015 s
Point2_SR2	Satisfied	0.016 s
Point3_SR2	Satisfied	0.015 s

SR3: Two trains cannot reside on the same track circuit or at the same point region at the same time. The number of trains on a track circuit segment or a point region must not be greater than one. Table 3 shows the analysis of SR3. SR3 is written in CTL formulation as follow:

$$AG\neg\left(R_k\_Busy > 1\right) \vee \left(M_j\_Busy > 1\right)$$

Table 3. *Analysis of SR3*

Query	Result	Verification time
M0_SR3	Satisfied	0.016 s
M1_SR3	Satisfied	0.016 s
M2_SR3	Satisfied	0.016 s
M3_SR3	Satisfied	0.015 s
M4_SR3	Satisfied	0.015 s
R0_SR3	Satisfied	0.016 s
R1_SR3	Satisfied	0.16

7. CONCLUSION

In this study, the modeling and verification of fixed block railway signalization systems is examined and a new automatically modeling and verification formalism is introduced. Here the introduced models are generated by using timed arc Petri nets, which is a highly recommended formal method by CENELEC EN 50128 standard. The developed models include all the primary dynamics of the fixed block railway signalization systems in a plain way such that they can be easily extended by the railway sector experts if needed. Also a software tool is developed by using C# programming language to automatically generate TAPN models for railway signalization interlocking systems by a visual interface. These models can also be viewed through TAPAAL, which is a tool for editing, simulating, analyzing and verifying TAPN. The verification process is done on the basis of certain safety requirements, which are written in CTL formulation. In order to show the effectiveness of the introduced models and algorithms, the results are implemented on Saray station, which is operated by Turkish Railway Company. The results prove the effectiveness and accuracy of the introduced approach such that model production and verification times is very low. Finally it is necessary to state that the developed software reduces the human implementation errors which is very critical when considered safety critical systems.

CONFLICT OF INTEREST

No conflict of interest was declared by the authors

REFERANCES

[1] The European Raıl Industry (UNIFE). (2015). Annual Report. Avenue Louise 221, Bte 11 B – 1050 Brussels.

[2] CENELEC EN 50128, “Railway applications - Communication, Signaling and Processing Systems - Software for Railway Control and Protection Systems (2011).

[3] S. A. Khan, N. A. Zafar, F. Ahmad, and S. Islam, “Extending Petri net to reduce control strategies of railway interlocking system,” Appl. Math. Model., vol. 38, no. 2, pp. 413–424, 2014.

[4] H. Wang, F. Schmid, L. Chen, C. Roberts, and T. Xu, “A topology-based model for railway train control systems,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 819–827, 2013.

[5] Anne E. Haxthausen, Jan Peleska, and Ralf Pinger, “Applied Bounded Model Checking for Interlocking System Designs,” Towards a Formal Methods Body of Knowledge for Railway Control and Safety Systems: FM-RAIL-BOK Workshop 2013.

[6] Robert Abo and Laurent Voison, “Data Formal Validation of Railway Safety-Related Systems: Implementing the OVADO Tool,” Towards a Formal Methods Body of Knowledge for Railway Control and Safety Systems: FM-RAIL-BOK Workshop 2013.

[7] P. James, F. Moller, H. N. Nguyen, M. Roggenbach, S. Schneider, and H. Treharne, “Techniques for modelling and verifying railway interlockings,” Int. J. Softw. Tools Technol. Transf., vol. 16, no. 6, pp. 685–711, 2014.

[8] Malakar, B., & Roy, B. K. Railway fail-safe signalization and interlocking design based on automation Petri Net. International Conference on Information Communication and Embedded Systems (ICICES), 2014, pp. 1-4.

[9] Sun, P., Collart-dutilleul, S., & Bon, P. (2015, June). A model pattern of railway interlocking system by Petri nets. International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 442-449.

[10] X. Wang, T. Tang, S. Liu, Study on modeling and verification of CBTC interlocking system. 5th IET International Conference on Wireless, Mobile and Multimedia Networks (ICWMMN), 2013, pp. 350-354.

[11] I. Sener, O.T. Kaymakçı, I. Ustoğlu, G. Cansever. Specification and formal verification of safety properties in point automation system. Turkish Journal of Electrical and Computer Engineering Sciences, in press, DOI: 10.3906/elk-1311-27.

[12] M.A.N. Oz, I. Sener, O.T. Kaymakçı, I. Ustoğlu, G. Cansever. Topology based automatic TAPN model generation for railway systems. International Journal of Automation, Mechatronics & Robotics 1 (2014), pp. 6-10.

[13] M.A.N. Oz, I. Sener, O.T. Kaymakçı, I. Ustoğlu, G. Cansever. A tool for automatic formal modeling of railway interlocking systems. International Conference on Computer as a Tool (EUROCON), 2015, pp. 1-4.

[14] Andrea Bonacchi, Alessandro Fantechi, Stefano Bacherini, Matteo Tempestini, and Leonardo Cipriani, “Validation of Railway Interlocking Systems by Formal Verification, a Case Study,” Towards a Formal Methods Body of Knowledge for Railway Control and Safety Systems: FM-RAIL-BOK Workshop 2013.

[15] W. Zheng, C. Liang, R. Wang, and W. Kong, "Automated test approach based on all paths covered optimal algorithm and sequence priority selected algorithm," IEEE Trans. Intell. Transp. Syst., vol. 15, no. 6, pp. 2551–2560, 2014..

[16] A. Mekki, M. Ghazel, and A. Toguyéni, "Validation of a New Functional Design of Automatic Protection Systems at Level Crossings with Model-Checking Techniques," Intell. Transp. Syst. IEEE Trans., vol. 13, no. 2, pp. 714–723, 2012.

[17] L. Chen, Z. Shan, T. Tang, and H. Liu, "Performance analysis and verification of safety communication protocol in train control system," Comput. Stand. Interfaces, vol. 33, no. 5, pp. 505–518, 2011.

[18] A.G. Russo, L. Ladenberger. A formal approach to safety verification of railway signaling systems. Reliability and Maintainability Symposium (RAMS), 2012, pp. 1-4.

[19] P. James, F. Moller, H.N. Nguyen, M. Roggenbach, S. Schneider, H. Treharne. On modelling and verifying railway interlockings: Tracking train lengths. Science of Computer Programming, Special Issue on Automated Verification of Critical Systems (AVoCS 2012) 96, 2014, pp. 315–336.

[20] A. Kuzu, O. Songuler, A. Sonat, S. Turk, B. Birol, E. H. Dogruguven. Automatic interlocking table generation from railway topology. IEEE International Conference on Mechatronics, 2011, pp. 64-70.

[21] U. Yildirim, M. S. Durmus, M. T. Soylemez. Automatic interlocking table generation for railway stations using symbolic algebra. 13th IFAC Symposium on Control in Transportation Systems 13 (2012), 171-176.

[22] Y. Cao, T. Xu, T. Tang, H. Wang, L. Zhao. Automatic generation and verification of interlocking tables based on domain specific language for computer based interlocking systems (DSL-CBI). IEEE International Conference on Computer Science and Automation Engineering (CSAE) 2 (2011), 511-515.

[23] L. Jacobsen, M. Jacobsen, M. H. Moller, J. Srba. Verification of timed-arc Petri nets. Lecture Notes in Computer Science 6543 (2011), 46-72.

[24] M. Andersen, H. G. Larsen, J. Srba, M. G. Sørensen, J. H. Taankvist. Verification of liveness properties on closed timed-arc Petri nets. Lecture Notes in Computer Science 7721 (2013), 69-81.

[25] A. David, L. Jacobsen, M. Jacobsen, K. Y. Jørgensen, M. H. Møller, J. Srba. TAPAAL 2.0: integrated development environment for timed-arc Petri nets. Lecture Notes in Computer Science 7214 (2012), 492-497.

[26] L. Jacobsen, M. Jacobsen, M.H. Moller, J. Srba, "Verification of timed arc Petri nets," Lecture Notes in Computer Science, vol. 6543, pages 46-72, 2011.

[27] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In Logic of Programs. Proceedings of Workshop, volume 131 of Lecture Notes in Computer Science, pages 52–71. Springer-Verlag, 1981.