Turkish Journal of Electrical Engineering and Computer Sciences

Volume 24 | Number 3 Article 49

1-1-2016

Specification and formal verification of safety properties in a point automation system

İBRAHİM ŞENER

ÖZGÜR TURAY KAYMAKÇI

İLKER ÜSTOĞLU

GALİP CANSEVER

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 1384 – 1396

doi:10.3906/elk-1311-27

Specification and formal verification of safety properties in a point automation system

İbrahim ŞENER, Özgür Turay KAYMAKÇI, İlker ÜSTOĞLU, Galip CANSEVER*
Department of Control and Automation Engineering, Faculty of Electrical & Electronics Engineering,
Yıldız Technical University, İstanbul, Turkey

Received: 04.11.2013 • **Accepted/Published Online:** 07.03.2014 • **Final Version:** 23.03.2016

Abstract: Railroad transportation systems are an area that poses the threat of causing huge risk for both the environment and people if an error emerges during operation. For this reason, designing and developing relevant products in this area is challenging. What is more, methods to be utilized for the purposes of minimizing risk susceptibility are to be specified by international standards. While relevant standards strongly recommend that some methods be utilized based on the desired safety integrity level during the development phase, some methods are not recommended to be utilized. CENELEC 50128 strongly recommends the utilization of timed-arc Petri nets during system modeling and the utilization of formal proof methods during the verification and test phases of the command and control structure developed. In this study, a control structure related to the safety of the point automation system, which has a critical significance for tram lines, was designed through timed-arc Petri nets by taking the relevant standard as the reference. The verification was performed through computational tree logic, which is one of the formal proof methods. The timed-arc Petri nets model has been used for the first time in this area in this study. Within this context, the structure was developed by taking the point automation system at the 50. Yıl Station on the T4 Topkapı-Habibler line, operated by İstanbul Ulaşım A.Ş., as the reference. Moreover, safety requirements for the automation of the points were identified and denoted mathematically while their safety functions were designed.

Key words: Point automation, safety, formal verification, temporal logic, timed-arc Petri net

1. Introduction

Transportation has become the most important concern for people these days. Railroad systems have come to the forefront in many countries for the solution of traffic and transportation problems in big cities. Accordingly, the problem has been alleviated. Although the railroad transportation systems remained underdeveloped until the 1990s, they have great significance nowadays. Investments in railroad transportation systems to alleviate traffic congestion seen on roads of big cities have been on the rise.

Railroad systems are one of the most important means of transportation both for passenger transportation and freight shipment. Railroad systems enjoy a great number of advantages over other means of transportation. One of the most important advantages is that railroad system transportation has become safer, more economic, faster, and environmentally friendly thanks to developing technology. Such advantages prove that railroad systems enjoy superiority to other modes of transportation for the time being. Safety and reliability issues become more significant for railroad transportation systems than for roads when the length, weight, and

*Correspondence: cansever@yildiz.edu.tr

passenger capacities of the trains are taken into consideration. Point automation systems guide the movement of the vehicles on the tramlines for the safe conducting of vehicles. For this reason, control and automation of the points should be performed in a safe manner, particularly near the stations, so that casualties and material losses can be prevented.

Use of formal methods in modeling and verifying signalization and interlocking systems in far more complicated systems, like railroads, where safety and reliability are of crucial value is strongly recommended by CENELEC 50128. There exists a great number of studies in the literature regarding the designing of signalization and interlocking systems using formal methods. In one such study, Winter [1] formed a formal model for a railroad interlocking system by using communicating sequential processes, which is a formal modeling language. The author categorized the model into two separate parts, which were the interlocking model and the signaling model. She also checked the functional specifications of the formal model against the signaling principles of the formal model, using the failures-divergences refinement model checking tool. By converting safety properties defined by ladder logic into propositional logic, Kanso et al. [2] conducted a study where they verified the signalization principles written in ladder logic format. Thus, they managed to verify the safety properties automatically thanks to the software they developed through verification strategy. Russo and Ladenberger [3] proposed a new tool, named VeRaSIS, with a formal method base that verified the rail topology and movement conditions of the trains. This tool enables the graphical simulation of railroad specifications. In this way, the movement properties of the trains can be automatically produced and verified. Jo et al. [4] proposed an eclectic approach to incorporate Z (Zed) formal language and 'Statemate MAGNUM', which is a formal method tool, using Statechart. They also applied the proposed method to safety-critical railway signaling systems for the formal requirement specification and analyzed the specification.

As the verification of the system designed can be made by formal methods, Petri nets have become a formal modeling tool used frequently for railroad systems [5–8]. Ahmad and Khan [9] modeled the multiple crossing region near Vancouver station through arc-constant colored Petri nets and identified the safety requirements for the trains so that they could have safe passing from the region. They also verified such requirements by coverability tree method. Hei et al. [10] proposed the use of a distributed interlocking system instead of the traditional interlocking structure they had modeled using Petri nets in their study. They performed the verification of the proposed system by using the coverability tree method. Giua and Seatzu [11] modeled each component like the point, track circuit, and station separately. They modeled a railroad network using Petri nets and designed a supervisory controller in the model formed so that the trains could complete the travel safely on the railroad network.

The point automation system of the 50. Yıl station on the T4 Topkapı-Habibler line operated by İstanbul Ulaşım A.Ş. was designed by using timed-arc Petri nets in this study unlike the other formal methods following the widening of Petri nets. This enabled the modeling and simulation of a time-bound system to act in a more realistic way. As a result, the movement of points and that of trains in the station occurred within a certain framework of time. In addition, temporal acts could be transferred into the model better, which enabled the empowering of the system modeling. It was not possible to form models that were powerful enough to reflect the system in previous studies since temporal movements in the system were not transferred into the model due to the fact that timed-arc Petri nets had not been used.

Studies performed earlier considered the system as a whole rather than focusing on the points, which is one of the most important building blocks of railroad systems [12–14]. This study, however, dwells on the automation and control of the points. It is important that safety requirements necessary for a safe journey be

identified and based on a mathematical basis while conducting the automation of the points. In this study the safety requirements necessary for the automation of points are identified and their safety functions are designed by denoting such requirements mathematically. Another important issue is to test whether the models formed to ensure the accurate and safe conduct of the point automation system fulfill the identified safety requirements or not. Therefore, the TAPAAL editor was used to verify the existence of anticipated safety requirements for the relevant functions. The verification of the identified safety requirements was made automatically [EF (there exist some reachable markings that satisfy), EG (there exists a trace on which every marking satisfies), AF (on all traces there is eventually a marking that satisfies), and AG (all reachable markings satisfy)] and was written based on the computational tree logic formulation, which is a subcategory of temporal logic.

2. Railway point automation and its subcomponents

As one of the fundamental blocks of railroad systems, points enable the trains to maneuver to the right or left. They also play a crucial role in ensuring a safer and speedier journey on the rails. For this reason, conducting the checks of points is as significant as their production and installation into the system [15]. Efficiency and speed of a railroad is highly influenced by the number and form of the points. Reliability of a railroad is also directly related to the automation and checking of these points. In double track tramlines, performing the automation safely in places where points are located, rather than monitoring the entire line, is a method acknowledged today, which is generally due to cost factors. Nowadays, fail-safe command and control of the points is conducted on tramlines, particularly at stations where points are concentrated. Other components of the railway in the station also play a significant role in the conducting of point automation at a station. This section gives a brief introduction of each component.

2.1. Point

A railway point is a mechanical tool that is usually controlled with an electrical motor that lets the trains be guided from one track to another at a railway intersection according to the desired route. A point can be settled in two positions, named as normal and diverging. There two positions determined at the mounting stage and they cannot be changed further. When a route is desired to be formed, the corresponding point is moved laterally from one position to another by the interlocking system itself.

2.2. Signals

As the brake distance of railway transportation vehicles is more than that of other vehicles, it becomes necessary to use signals, which enables a safe area between the trains. Signals are systems that transmit colored lights, notifying the trains regarding the proceeding of the trains until the next signal. Trams operate within a system where vehicles with low speed run based on double track lines. At this point, notifications of the signals provide information more so on whether the destination line is available or not as well as the mode of movement (whether the vehicle will turn toward another direction or go straight forward to the destination) and if the route involves a place with the point rather than providing information regarding the speed.

2.3. Track circuits

It is important to know which point the trains are at so that railway traffic can be managed safely. A track circuit is an electrical circuit used to detect whether the route is available or occupied by a railroad vehicle. The relevant mechanism works by using the rails in one part of the road as conductors and short-circuiting the rails

by the train wheels. Using track circuits has certain pros and cons. The most significant advantage is that it becomes possible to detect problems like track cracks thanks to track circuits. The disadvantages, on the other hand, are the interaction that could be brought about by traction currents and problems regarding impedance bonds.

3. Determination of point automation safety requirements

It is an accepted fact that trains can be easily affected by any disorder in the railway traffic. The visibility ranges are usually not adequate enough to let the locomotive drivers stop the trains; furthermore, stop distances of a train can vary within a large interval based on total mass. For this reason, railway signaling systems are developed to control railway traffic securely, fundamentally to prevent trains from colliding and derailing as well. Within this context, with tramlines having double tracks, the conduction of the vehicles is generally performed via point automation systems. For a safe journey it is important to ensure that the safety requirements are identified formally. In addition, it should be assured whether the control structure achieved as a result of system modeling fulfills the necessary requirements or not. It is important to bear in mind that fatal accidents are bound to happen if there remains an unfulfilled requirement.

The point automation system in this study was modeled by using timed-arc Petri nets, one of the formal methods based on the CENELEC EN 50128 standard, which was also highly recommended to be used for the relevant standard. The use of formal methods enabled the safety requirements to be denoted and defined better. Moreover, verifying the accuracy of the requirements becomes easier as each step will be performed within the context of an identified rule. Another advantage of using formal methods is that such methods enable one to do the following: examine the system symbolically, ensure its accuracy, or prove that the safety requirements are accurate in all states. For all these reasons, formal methods are recommended strongly in the CENELEC EN 50128 standard. Although the use of mathematical logic is a unifying theme across the discipline of formal methods, there exists one perfect formal method. Each application might need different methods of modeling, which are communicating sequential processes, calculus of communicating systems, higher order logic, language of temporal ordering specification, OBJ, temporal logic, the Vienna development method, the Z method, the B method, and model checking (D.28 formal methods) [16]. In this study, computational tree logic, which is a subcategory of temporal logic, was used and it was verified that the relevant safety requirements had been ensured. Safety requirements (SRs) to be fulfilled by the system in point automation projects can be listed as follows:

SR1. The point should either be in its normal position or in a diverging position as it cannot remain in the same position concurrently.

SR2. For a point to be locked, the point should either be in its normal position or in a diverging position. Otherwise, point position error should be noted and the route should not be opened.

SR3. The point should not be moving while the train occupies any point, which means that while the train is on its way over the point, it should not get any point engine command or move.

SR4. Signals should be locked into green, yellow, and red lights, referring to normal direction, side direction, and stopping direction, respectively. The train should start moving when the signal notifies about the proceeding direction, and the signal should give red notification once the train occupies the first track circuit.

SR5. When the route selected is locked and opened, the points on the route should also be locked in the relevant position and there should be no proceeding until the route is free.

SR6. Points should first be locked based on the route chosen. Then relevant signal notification should be given when the route is locked.

The safety requirements mentioned above are depicted as follows, where all the points in the field are represented as $P = \{p_1,\ p_2,\ p_3,\ \ldots,\ p_i\}$, all track circuits as $TC = \{tc_1,\ tc_2,\ tc_3,\ \ldots,\ tc_j\}$, signals as $S = \{s_1,\ s_2,\ s_3,\ \ldots..,\ s_k\}$, all probable routes by $R = \{r_1,\ r_2,\ r_3,\ \ldots..,\ r_m\}$, and trains by TR= $\{tr_1,\ tr_2,\ tr_3,\ \ldots..,\ tr_n\}$.

I. $F : P \rightarrow$ Normal $\vee\ P \rightarrow$ Diverging

$\forall p \in P,\ \ F(p)$

F $\rightarrow p_k$ point is either in normal position or in diverging position.

II. $P : R\,X\,P \rightarrow Partof\,(r,p) \rightarrow Pointlocked$

$\forall r \in R,\ \forall p \in P,\ \ P(rp)$

$(r_k\,,\,p_k) \rightarrow$ Pointlocked $\in P \Longrightarrow (p_k) \rightarrow$ Normal $\vee\ (p_k) \rightarrow$ Diverging $\in F$

P $\rightarrow r_k p_k$ point is locked on the route specified. The locked p_k point is either in normal position or in diverging position.

III. $O : P\,X\,TR \rightarrow Occupied$

$\forall p \in P,\ \forall tr \in TR,\ \ O(p\,,\,tr)$

$(p_k\,,\,tr_k) \rightarrow$ Occupied $\in O \Longrightarrow (r_k\,,\,p_k) \rightarrow$ Pointlocked $\in P$

O $\rightarrow p_k$ point is occupied by tr_k train. The occupied point is locked.

IV. $K : S \rightarrow$ Green $\vee\ S \rightarrow$ Yellow $\vee\ S \rightarrow$ Red

$\forall s \in S,\ \ K(s)$

$(p_k\,,\,tr_k) \rightarrow$ Occupied $\in O \Longrightarrow (s_k) \rightarrow Red \in K$

K $\rightarrow s_k$ signal notifies green, yellow, or red.

V. $R : R \rightarrow$ Routelocked

$\forall r \in R,\ \ R(r)$

$(r_k) \rightarrow$ Routelocked $\in R \Longrightarrow (r_k\,,\,p_k) \rightarrow$ Pointlocked $\in P$

R $\rightarrow r_k$ route is locked. Points on the route are also locked once the route is locked.

VI. $S : R\,X\,S \rightarrow$ Partof $(r,s) \rightarrow$ Signallocked

$\forall r \in R,\ \forall s \in S,\ \ S(rs)$

$(r_k) \rightarrow Routelocked \in R \Longrightarrow (r_k\,,\,s_k) \rightarrow Signallocked \in S$

$(r_k\,,\,s_k) \rightarrow Signallocked \in S \Longrightarrow s_k \rightarrow Green\ \vee\ s_k \rightarrow Yellow \in K$

$S \rightarrow s_k$ signal is locked. The route should be locked for the locking of the signal.

The locked signal indicates either green or yellow.

4. Timed-arc Petri nets

The timed-arc Petri net is defined with a 7-tuple $TAPN = \{P, T, IA, OA, Transport, Inhib, Inv\}$, where P is a finite set of places, T is a finite set of transitions, $IA \subseteq P\,x\,T\,x\,T$ is a finite set of input arcs, $OA \subseteq T \times P$ is a finite set of output arcs, $Transport : IA \times OA \rightarrow \{truefalse$ is a function defining transport arcs that are pairs of input and output arcs connected to some transition, $Inhib : IA \rightarrow \{truefalse$ is a function defining inhibitor arcs that do not collide with transport arcs, and $Inv : P \rightarrow T^{inv}$ is a function assigning age invariants to places. Here the preset of a transition $t \in T$ is defined as $^{\circ}t = [ERR : md : MbegChr = 0x007B, MendChr = 0x007C, nParams = 1]\,(p,\, I,\, t) \in IA$. Similarly, the postset of a transition t is defined as $t^{\circ} = [ERR : md : MbegChr = 0x007B, MendChr = 0x007C, nParams = 1]\,(t, p) \in OA$. Similar to a basic Petri net a marking M on N is a function $M : P \rightarrow B\,(R \geq 0)$ where for every place $p \in P$ and every token $x \in M(p)$ we have $x \in Inv(p)$. Thus, the set of all markings over N is denoted by $M(N)$. A marked timed-arc Petri net is a pair $(N,\, M_0)$ where N is a timed-arc Petri net and M_0 is an initial marking on N where all tokens have the age 0.

The enabling rule of a timed-arc Petri net is a little bit different from the basic Petri net. $\mathrm{t} \in \mathrm{T}$ is enabled in a marking M by tokens $\mathrm{In}[ERR : md : MbegChr = 0x007B, MendChr = 0x007C, nParams = 1]\,\mathrm{p} \in {}^{\circ}\mathrm{t}\} \subseteq \mathrm{M}$ and $\mathrm{Out} = [ERR : md : MbegChr = 0x007B, MendChr = 0x007C, nParams = 1]\,\mathrm{p}' \in \mathrm{t}^{\circ}\}$ if $\forall\,(\mathrm{p}, \mathrm{I}, \mathrm{t}) \in \mathrm{IA}.\ \neg\mathrm{Inhib} \quad ((\mathrm{p}, \mathrm{I}, \mathrm{t})) \Longrightarrow \mathrm{x} \in \mathrm{I}$ and $\forall\,(\mathrm{p}, \mathrm{I}, \mathrm{t}) \in \mathrm{IA}.\ \mathrm{Inhib} \quad ((\mathrm{p}, \mathrm{I}, \mathrm{t})) \Longrightarrow \neg\exists \mathrm{x} \in \mathrm{M}\,(\mathrm{p})\,.\,\mathrm{x} \in \mathrm{I}$ and $\forall\,(\mathrm{p}, \mathrm{I}, \mathrm{t})$ IA. $\forall\left(\mathrm{t}, \mathrm{p}^{'}\right)$ OA.

Transport $((\mathrm{p}, \mathrm{I}, \mathrm{t}))\,\mathrm{Inhib}\left((\mathrm{p}, \mathrm{I}, \mathrm{t})\,,\left(\mathrm{t}, \mathrm{p}^{'}\right)\right) \Longrightarrow \left(\mathrm{x_p} = \mathrm{x_{p'}}\right) \wedge \left(\mathrm{x_p} \in \mathrm{Inv}\left(\mathrm{p}^{'}\right)\right)$ and $\forall\left(\mathrm{t}, \mathrm{p}^{'}\right) \in \mathrm{OA}.$ $(\neg(\exists\alpha \in IA.\, Transport(\alpha(t, \mathrm{p}^{'}))) \Longrightarrow \mathrm{x_{p'}} = 0$ conditions hold. The firing rule t is enabled in the marking M by tokens In and Out and then it can fire and produce a marking $\mathrm{M_0}$ defined as $\mathrm{M}^{'} = (\mathrm{M} \setminus \mathrm{In}) \cup \mathrm{Out}$ where M is a marking on N and $\mathrm{t} \in \mathrm{T}$ is a transition. The time delay $\mathrm{d} \in \mathrm{R} \geq$ is allowed in M if $(\mathrm{x} + \mathrm{d}) \in \mathrm{Inv}\,(\mathrm{p})$ for all $\forall \mathrm{p} \in \mathrm{P}$ and $\forall \mathrm{x} \in \mathrm{M(p)}$. For detailed information about timed-arc Petri nets, refer to [17,18].

5. Timed-arc Petri net modeling of points and subcomponents

CENELEC EN 50128 Table A.4-Software Design & Imp. requires the use of a modular approach. Modeling and design of the system was conducted on a modular basis considering the subcomponents to stick to the relevant requirement. For this reason, separate timed-arc Petri net models were formed for points and signals. After that, the models were connected and the timed-arc Petri net model belonging to the point automation system was achieved. No model was formed for the track circuit. It was integrated into the system as the field model.

5.1. Point timed-arc Petri net model

All points are assumed to be in the normal position at the initial stage. There are five places and four transitions. Places are for $\forall_p \in P$, point model P = {P_Enable, RCM, Point_N, Point_R, RtoN, NtoR} . Transitions are T = {T0, T1, T2, T3} . The places respectively indicate the following situations: the point is enabled, the occupancy of the point, the point is in normal position, the point is in diverging position, the point is passing from diverging position to normal position and the point is passing from normal position to diverging position. If there is a token of age 0.0 in the place RCM (point track circuit), it means it is occupied; if not, it means the point is not occupied. Likewise, if there is a token in the P_Enable section, it means the point position can be changed, and if not it means the position cannot be changed. For the point to change position, it should not be

locked for any route in the enable status and there should be no tokens in the RCM section, which means the point is not occupied by a train. When enabled, the point moves towards a diverging position. It is required to achieve diverging position by completing its movement (within a [max1, max2] interval) within a certain time period. In case it does not achieve a diverging position within a certain time period, this will be identified as point position error and the intended route will not be opened. The same rule applies for moving from a diverging position to a normal position. The relevant timed-arc Petri net model formed can be seen in Figure 1.

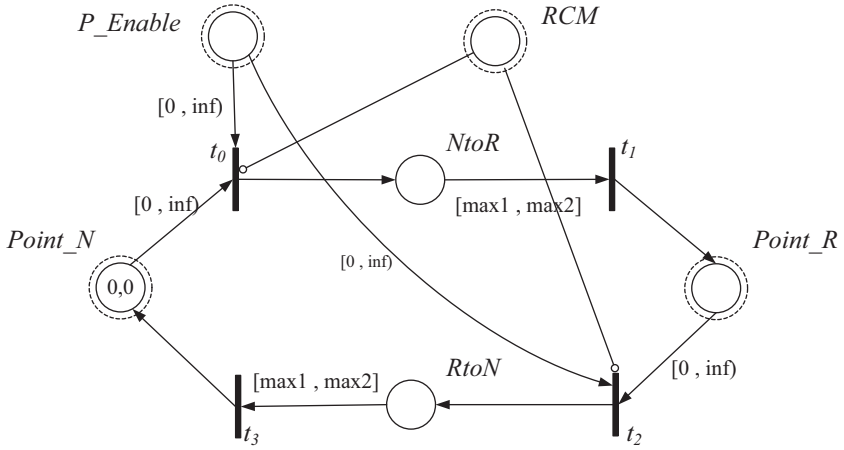

Figure 1. Point timed-arc Petri net model.

5.2. Signal timed-arc Petri net model

Signals indicate red at the initial stage. There are four places and two transitions: places are $\forall s \in S$, signal model P = {Signal_Enable, Signal_red, Signal_green, TrEntM} , and transitions are T = {T0, T1} . The places respectively indicate the following situations: the signal is enabled, the signal indicates red, the signal indicates green and the train occupies the first track circuit following the signal. After the points on the route to be opened reach the relevant position, the signal is enabled and a green notification is transmitted to the train to allow it to pass. As the train passes the signal and occupies the first track circuit, the signal indicates red once again. The timed-arc Petri net model formed for the signal can be seen in Figure 2.

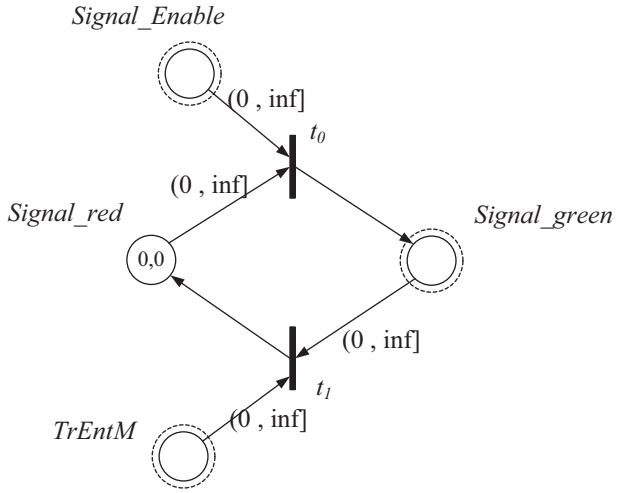

Figure 2. Signal timed-arc Petri net model.

6. Point Automation System of 50. Yıl Station

Below can be seen the track scheme of the 50. Yıl Station on the T4 Topkapı-Habibler line operated by İstanbul Ulaşım A.Ş. chosen as a model. The station has five points, five signals, and ten track circuits.

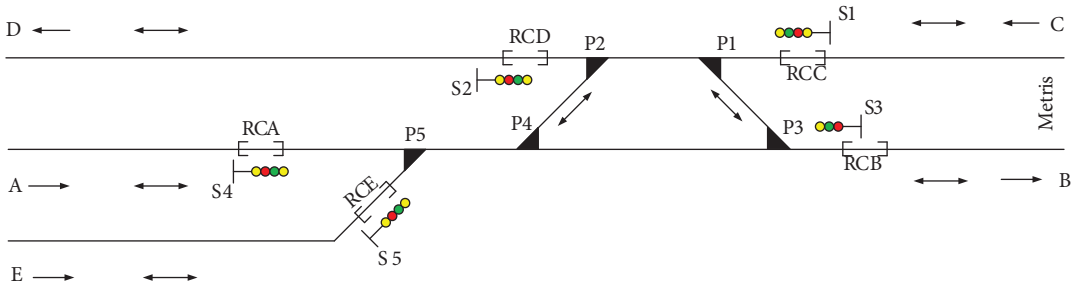

Figure 3. The 50. Yıl Station track scheme.

Sets to represent the following items at 50. Yıl Station, whose track scheme is presented in Figure 3, were defined: five points $P = \{p_1,\ p_2,\ p_3,\ p_4,\ p_5\}$ and ten track circuits $TC = \{RCA,\ RCB,\ RCC,\ RCD,\ RCE,\ RCM1,\ RCM2,\ RCM3,\ RCM4,\ RCM5\}$, the first of five indicating the entering and departing of the station and the last five indicating the occupancy of the points, as well as five signals, $S = \{s_1,\ s_2,\ s_3,\ s_4,\ s_5\}$. In addition to these sets, other sets were also defined, such as the set $TR = \{tr_1,\ tr_2,\ tr_3, \ldots\ldots tr_n\}$ to represent the trains and the route set $R = \{r_1,\ r_2,\ r_3,\ r_4,\ r_5,\ r_6,\ r_7\}$ that can be opened for these trains.

The routes identified can be opened for the trains on the condition that the track circuits are not occupied and the train proceeding on the second route to be opened should not be facing the train proceeding on the first route. Based on this, separate timed-arc Petri net models were formed for each route. As an example, the r_1 route timed-arc Petri net model can be seen in Figure 4, formed for a train that will be proceeding on the CD route.

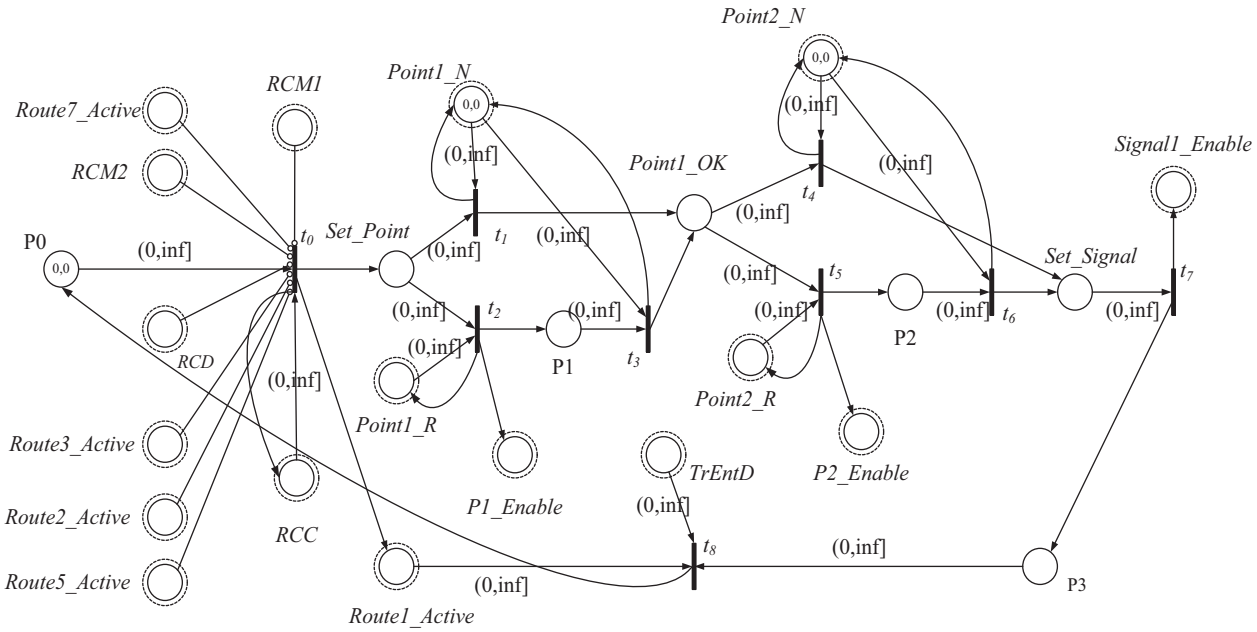

Figure 4. r_1 route timed-arc Petri net model.

Based on the model formed, the r_1 route can be opened provided that tc_1 (RCC) and tc_2 (RCD) track circuits are unoccupied and the relevant points (Point1 and Point2) are not occupied, either. The former condition should be fulfilled earlier than the latter. In addition, any of the r_2, r_3, r_5, or r_7 routes should be opened; they should not be locked. The points (Point1_N and Point2_N) on the route are placed in appropriate positions in the right order once the route is chosen. As a next step, Signal1 is enabled and a green notification is transmitted. At that point the train starts moving. Any route that might clash with the route of the train, from C (the entrance point of the train into the station) to station D (where the train leaves the station), is not allowed to be opened. The same situation applies for all the other routes. A new route can be opened on the condition that the track circuits on that route are unoccupied and the points are not occupied, either. It is also required that any other route has not been opened. The Table represents the points, their relevant positions based on the routes to be opened, and which track circuits are controlled.

Table. Track circuit, point, and point position by route.

Entrance into the station	Route	Controlled point and its position	Track circuit controlled
C	r_1	P1 → N P2 → N	RCC, RCD RCM1, RCM2
	r_2	P1 → N P2 → R P4 → R P5 → N	RCC, RCA RCM1, RCM2 RCM4, RCM5
	r_3	P1 → N P2 → R P4 → R P5 → R	RCC, RCE RCM1, RCM2 RCM4, RCM5
A	r_4	P3 → N P4 → N P5 → N	RCA, RCB RCM3, RCM4 RCM5
	r_5	P1 → N P2 → R P4 → R P5 → N	RCA, RCC RCM1, RCM2 RCM4, RCM5
E	r_6	P3 → N P4 → N P5 → R	RCE, RCB RCM3, RCM4 RCM5
	r_7	P1 → N P2 → R P4 → R P5 → R	RCE, RCA RCM1, RCM2 RCM4, RCM5

Track circuits working based on the occupancy principle constantly provide the feed on where the trains are. This is a condition required for a safe journey. The $TC_x = \{tc_1, tc_2, tc_3\}$ set denotes the track circuit set occupied by the trains during their entrance into the station for the Metris station, whereas the $TC_y = \{tc_1, tc_2, tc_3, tc_4, tc_5\}$ set depicts the track circuit set occupied by the trains while they are leaving the station. With $x, y = \{1, 2, 3, \ldots n\}$ ve $x \neq y$, it is assumed that the train remains at the station as long as it does not pass from a second track circuit based on the route opened after it passes a track circuit. In Figure 5 can be seen the track circuit timed-arc Petri net model, which indicates the actions of the trains that enter

the station from C. Similarly, timed-arc Petri net models for trains entering the station from A and E were also formed.

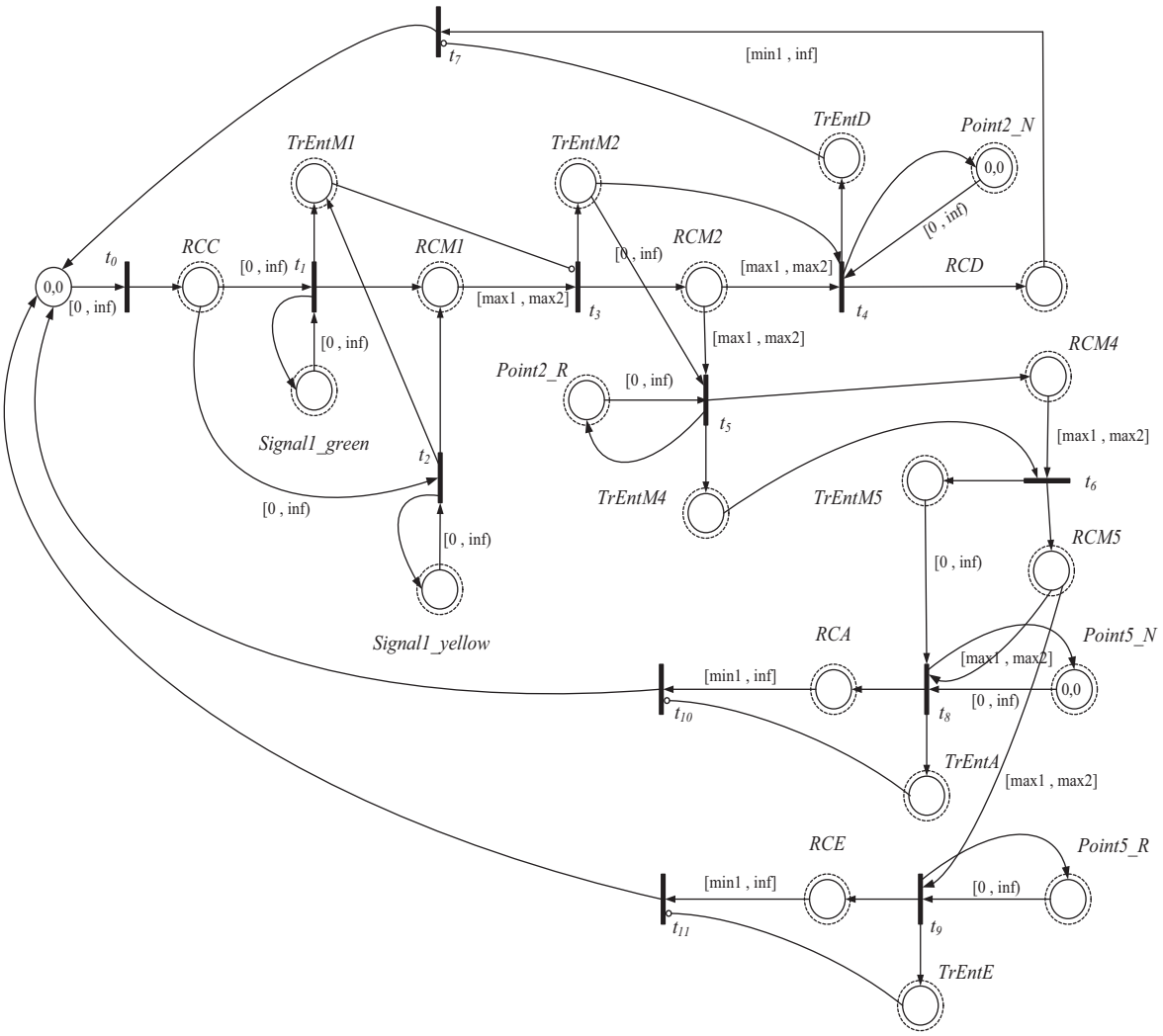

Figure 5. Track circuit timed-arc Petri Net model for trains entering the station from C.

As specified in the previous section and based on the model formed, the trains entering the station from C can leave the station from D, A, or E depending on the route to be chosen. The train occupies the RCC track circuit initially. Then it proceeds on the route opened, occupying one of the track circuits, which are RCD, RCA, or RCE, and leaves the station.

7. Verification of the point automation system of 50. Yıl Station

It is of great importance to verify and prove that the formed system models fulfill the identified safety requirements so that a safe journey can be ensured on railway systems. To verify the accuracy of the safety requirements identified in the point automation system designed for the 50. Yıl Station, the TAPAAL editor was used. The editor allows the modeling, simulation, and verification of the systems through timed-arc Petri nets. The verification of the identified safety requirements was made automatically as (EF, EG, AF, AG) and

written based on the computational tree logic formulation, which is a subcategory of temporal logic. Thus, it is possible to determine whether the formulae verify the formed model or not as a result of the verification procedure.

It is examined whether the queries, which were written in the verification process, fulfill the identified safety requirements or not by considering all reachable markings (AG) in the timed-arc Petri nets model of the system. The second query, written in SR5, is verified by considering some reachable markings (EF). In cases where the route r_1 is not locked, points can be in their normal position. All queries are checked via the TAPAAL discrete verification method based on the breadth-first search order in state space. As the coverability tree is too large, it is not given in this study.

SR1: The point should either be in its normal position or in a diverging position as it cannot remain in the same position concurrently.

For $\forall p \in P$,

$AG \neg (Normal(p_k) \wedge Reverse(p_k))$

$\equiv AG \neg (Point_k\_N \geq 1 \wedge Point_k\_R \geq 1)$

The property is satisfied.

The points can either be in their normal position or in a diverging position. They cannot remain in the same position concurrently.

SR2: For a point to be locked, the point should either be in its normal position or in a diverging position.

For $\forall p \in P$,

$AG(Pointlocked(r_k, p_k) \wedge (Normal(p_k) \vee Reverse(p_k)))$

$\equiv AG(P_k\_Enable = 0 \wedge (Point_k\_N \geq 1 \vee Point_k\_R \geq 1))$

$AG \neg (Pointlocked(r_k, p_k) \wedge (Normal(p_k) \wedge Reverse(p_k)))$

$\equiv AG \neg (P_k\_Enable = 0 \wedge (Point_k\_N \geq 1 \wedge Point_k\_R \geq 1))$

The property is satisfied.

For any point related to the route chosen to be locked, the point should either be in its normal position or in a diverging position. The point is locked either in a normal position or in a diverging position as it cannot remain in the same position concurrently. Otherwise, the point is not locked.

SR3: The point should not be moving while the train occupies any point, which means that while the train is on its way over the point, it should not get any point engine command or move.

For $\forall p \in P$,

$AG(Occupied(p_k, tr_k) \wedge Pointlocked(r_k p_k))$

$\equiv AG \neg (RCM_k \geq 1 \wedge (P_k NtoR \geq 1 \vee P_k RtoN \geq 1))$

The property is satisfied.

If the point is occupied by a train, the status of the point will never get into the modes of P_k.NtoR or P_k.RtoN, which represent the movement of the point used in the model, which goes respectively from normal to diverging and from diverging to normal. Thus, the point does not move toward a normal or diverging position while a train is passing over it.

SR4: The signal should be locked into green, yellow, and red light, referring to normal direction, side direction, and stopping direction, respectively. The train should start moving when the signal notifies about the proceeding direction, and the signal should give a red notification again once the train occupies the first track circuit.

For $\forall r \in R$, $\forall p \in P$ and $\forall s \in S$,

$AG\,(Green\,(s_k) \vee Yellow\,(s_k) \vee Red(s_k))$

$\equiv AG(Signal1\_green \geq 1 \vee Signal1\_yellow \geq 1 \vee Signal1\_red \geq 1))$

$AG\neg(Green\,(s_1) \wedge (Occupied(p_1 tr_k) \wedge Occupied(p_2, tr_k))$

$\equiv AG\neg(Signal1\_green \geq 1 \wedge (RCM1 \geq 1 \wedge RCM2 \geq 1))$

The property is satisfied.

Once the train starts moving on route r_1 and occupies Point1 and Point2, respectively, Signal1 does not give green notification. It turns red.

SR5: When the route selected is locked and opened, the points on the route should also be locked in the relevant position and there should be no proceeding until the route is free.

For $\forall r \in R$ and $\forall p \in P$,

$AG\neg(Routelocked\,(r_1) \wedge (Reverse(p_1) \vee Reverse\,(p_2)))$

$\equiv AG\neg(Route1locked \geq 1 \wedge (Point1\_R \geq 1 \vee Point2\_R \geq 1))$

$EF(Routelocked\,(r_1) \wedge (Normal(p_1) \wedge Normal\,(p_2)))$

$\equiv EF(Route1locked \geq 1 \wedge (Point1\_N \geq 1 \wedge Point2\_N \geq 1))$

The property is satisfied.

In order for route r_1 to be locked, Point1 and Point2 should definitely be in diverging positions when all accessible modes are considered in the timed-arc Petri net model. The route can be locked provided that both of the points are in diverging positions. Otherwise, route r_1 will not be opened or locked. In cases where the route is not locked, points can be in their normal position.

SR6: Points should first be locked based on the route chosen. Then relevant signal notification should be given when the route is locked.

For $\forall r \in R$, $\forall p \in P$ and $\forall s \in S$,

$EF\,(Routelocked\,(r_1) \wedge (Normal(p_1) \wedge Normal\,(p_2)))$

$\equiv EF(Route1locked \geq 1 \wedge (Point1\_N \geq 1 \wedge Point2\_N \geq 1))$

$AG\,(Routelocked\,(r_1) \wedge Green\,(s_1))$

$\equiv AG(Route1locked \geq 1 \wedge Signal1\_green \geq 1)$

The property is satisfied.

In order for route r_1 to be locked, Point1 and Point2 are locked in normal positions. Then green notification is given.

8. Conclusion

The point automation system of 50. Yıl Station, chosen as the model, on the T4 Topkapı-Habibler line and operated by İstanbul Ulaşım A.Ş., was successfully modeled and designed by using timed-arc Petri nets based on the CENELEC EN 50128 standard. The aim was that the trains would complete their transitions on the specified line safely. Safety requirements were identified so that the control of the point automation on railroad transportation systems could be performed in a safe manner. Correspondingly, the model of the system was formed based on such safety requirements. Eventually it was verified and proven through temporal logic, one of the formal methods recommended by the CENELEC EN 50128 standard, that timed-arc Petri net models fulfilled the identified safety requirements.

References

[1] Winter K. Model checking railway interlocking systems. Aust Comp S 2002; 24: 303–310.

[2] Kanso K, Moller F, Setzer A. Automated verification of signalling principles in railway interlocking systems. Electronic Notes in Theoretical Computer Science 2009; 250: 19–31.

[3] Russo AG, Ladenberger L. A formal approach to safety verification of railway signaling systems. In: Reliability and Maintainability Symposium; 23–26 January 2012; Reno, NV, USA. New York, NY, USA: IEEE. pp. 1–4.

[4] Jo HJ, Hwang JG, Yoon YK. Formal requirements specification in safety-critical railway signaling system. In: Transmission & Distribution Conference & Exposition: Asia and Pacific; 26–30 October 2009; Seoul, Korea. New York, NY, USA: IEEE. pp. 1–4.

[5] Piotrowicz M, Slusarczyk K, Napieralski A. A coloured Petri nets based solution for the generalized railway crossing problem. In: 14th International Conference on Mixed Design of Integrated Circuits and Systems; 21–23 June 2007; Ciechocinek, Poland. New York, NY, USA: IEEE. pp. 657–660.

[6] Fanti MP, Giua A, Seatzu C. Monitor design for colored Petri nets: an application to deadlock prevention in railway networks. Control Eng Pract 2006; 14: 1231–1247.

[7] Cheng YH, Yang LA. A fuzzy Petri nets approach for railway traf?c control in case of abnormality: evidence from Taiwan railway system. Expert Syst Appl 2009; 36: 8040–8048.

[8] Khana SA, Zafar NA, Ahmad F, Islam S. Extending Petri net to reduce control strategies of railway interlocking system. App Math Model 2014; 38: 413–424.

[9] Ahmad F, Khan S. Specification and verification of safety properties along a crossing region in a railway network control. App Math Model 2013; 37: 5162–5170.

[10] Hei X, Takahashi S, Nakamura H. Distributed interlocking system and its safety verification. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, Vol. 2; 2006; Dalian, China. New York, NY, USA: IEEE. pp. 8612–8615.

[11] Giua A, Seatzu C. Modeling and supervisory control of railway networks using Petri nets. IEEE T Autom Sci Eng 2008; 5: 431–445.

[12] Okan MR, Durmuş MS, Özmal K, Akçil L, Üstoğlu İ, Kaymakçı ÖT. Signaling system solution for urban railways: Esenler railway depot. In: IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications; 2013.

[13] Mutlu İ, Yıldırım U, Durmuş MS, Söylemez MT. Automatic interlocking table generation for non-ideal railway yards. In: IFAC Workshop on Advances in Control and Automation Theory for Transportation Applications; 2013.

[14] Lozano E, Hernando A, Alonso JA, Laita LM. A logic approach to decision taking in a railway interlocking system using Maple. Math Comput Simulat 2011; 82: 15–28.

[15] Kaymakçı ÖT, Üstoğlu İ, Anık VG. A local modular supervisory controller for a real railway station. In: 5th International System Safety Conference; 18–20 October 2010; Manchester, UK. New York, NY, USA: IEEE. pp. 1–6.

[16] CENELEC EN 50128. Railway Applications - Communication, Signalling and Processing Systems - Software for Railway Control and Protection Systems. East Greenwich, RI, USA: Vector Software, 2011.

[17] Jacobsen L, Jacobsen M, Moller MH, Srba J. Verification of timed-arc Petri nets. Lect Notes Comp Sci 2011; 6543: 46–72.

[18] Rakkay H., Boucheneb H., Roux OH. Time arc Petri nets and their analysis. In: 9th International Conference on Application of Concurrency to System Design; 1–3 July 2009; Augsburg, Germany. New York, NY, USA: IEEE. pp. 138–147.