Application of Functional Safety on Railways Part I: Modelling & Design

Mustafa Seçkin Durmuş
Istanbul Technical University
Istanbul, Turkey
durmusmu@itu.edu.tr

Uğur Yıldırım
Istanbul Technical University
Istanbul, Turkey
yildirimu@itu.edu.tr

Mehmet Turan Söylemez
Istanbul Technical University
Istanbul, Turkey
soylemezm@itu.edu.tr

***Abstract*—*By the rapid development of railway systems, the need of reliable signalization and interlocking systems increases day by day. Satisfying the requirements of standards like CENELEC (European Committee for Electrotechnical Standardization) ensures designers to achieve reliable system models. One of the main issues of designing part is to use formal methods while modelling these systems. Using Petri-Nets is the most popular method in the literature that provides designers easy error-tracking and a visual approach. In this study, modelling of a sample railway yard is achieved for signalization and interlocking design that considers required standards.*

Keywords; Functional Safety, Petri-Nets, Railway Interlocking and Signalization Design.

I. INTRODUCTION

Early railway systems were based mainly on railway guards who showed red flag or lamp to warn the driver of the train about the obstruction in front of him. Very soon after railways began to develop, many accidents occurred because of human failures (caused by railway guards or even by the drivers themselves) or nonhuman failures (caused by mechanical systems). As a result, the first interlocking installation was constructed at Bricklayers Arms, South Eastern Railway in 1843 [1]. Nowadays, train speed and densities are increasing day after day and as a result of this, reliable signalization and interlocking systems have to be developed and established to achieve safety of freight and passenger transportation on railway lines where a little error sometimes result in fatal accidents [2].

Functional safety requirements related with railway applications are described in *EN 50126* (where RAMS – Reliability, Availability, Maintenance and Safety analysis are described), *EN 50128* (defines software development requirements for railway applications) and *EN 50129* (requirement of hardware devices that are used in railway applications). Besides, the European standard *EN 61508* is developed as an umbrella standard for functional safety requirements of all kinds of electronic and programmable devices [3], [4]. *EN 50128* highly recommend to use semi-formal methods (like automata or Petri Nets) for developing safety related software for railway applications to satisfy the required Safety Integrity Levels (SIL).

The concept of *system* is described in the literature as *a combination of components that act together to perform a function not possible with any of the individual parts* [5]. In addition to description of system concept, some systems use *events* to change its states where an event is identified as a specific action taken (e.g. pressing a button) or a spontaneous occurrence (broken down of a car suddenly). A Discrete Event System (DES) is a system which changes its state by events that happen asynchronously [6].

Petri Nets [7] and Automation Petri Nets (which is an extended type of PNs that also deals with both sensors and actuators) [8] are one of the most widely used DES modelling tools in the literature because of their easiness [9], [10]. Interlocking and signalization design can also be achieved using APN models [11-15].

In this study, the requirements of functional safety stated in the *CENELEC EN 61508* and *EN 50128* documents are realized by using APNs. Definitions of PNs and APNs are given in Section 2. In Section 3, components of railway signalization are described. Modelling part is given in Section 4 and conclusive remarks are given in Section 5.

II. PETRI NETS AND AUTOMATION PETRI NETS

A. Petri Nets (PNs)

PNs [7] are graphical and also a mathematical DES modelling tool that has a wide range of application areas like railways [16], [17], industrial applications [18], control of flexible manufacturing systems [19]. A PN can be defined as follows:

$$PN = (P, T, Pre, Post, M_0) \quad (1)$$

- P : $\{p_1, p_2, \ldots, p_n\}$, finite set of places.
- T : $\{t_1, t_2, \ldots, t_n\}$, finite set of transitions.
- Pre : $(PxT) \rightarrow N$, directed ordinary arcs from places to transitions (N is a set of nonnegative integers).
- Post: $(TxP) \rightarrow N$, directed ordinary arcs from transitions to places.
- M_0 : $P \rightarrow N$, initial marking

B. *Automation Petri Nets (APNs)*

An extension of the Petri Net definition is done by adding four terms to ordinary PNs, and known as Automation Petri Nets (APN) [8], [10]. An APN is defined as follows:

$$APN = (P, T, Pre, Post, In, En, \chi, Q, M_0) \quad (2)$$

- In : (PxT) → N, inhibitor arcs from places to transitions.
- En : (PxT) → N, enabling arcs from places to transitions.
- χ : $\{\chi_1, \chi_2, \ldots, \chi_m\}$, firing conditions associated with the transitions.

Q : $\{q_1, q_2, \ldots, q_n\}$, finite set of actions that might be assigned to the places.

In addition to ordinary directed arcs (→) defined in PNs, inhibitor arcs (—◦) and enabling arcs (—▷) are added in APNs structure. These newly added arcs are ineffective on number of tokens in places but enable or inhibit transitions. More information on APNs can be found in [8] and [10].

III. SIGNALIZATION COMPONENTS

Traffic Command Center (TCC) is a place where all train traffic is controlled and route reservations are made by dispatchers. The commands of dispatcher, which are mostly route reservation requests, are sent to the interlocking system, where they are checked considering the current state of the railway yard. The interlocking allows only commands that do not endanger the safety of the system, and hence is usually considered as the most crucial part of the signalization system.

When, a route reservation required by the dispatcher is possible, interlocking adjusts all equipments related with that route reservation (colours of signal lights, position of switches and gate barriers etc.).

Track circuits (TCs) are simple electrical equipments which are used to detect trains on railways. Both DC and AC TCs are used in Turkish railways. On some regions axel counters are also used to detect trains. These equipments inform interlocking and also TCC about the absence or presence of the train [1], [20].

Similar to the road traffic, *colour signal lights* are used to inform the driver of the trains if the next railway block is occupied or not. Meanings of colour signals used in Turkish Railways can be found in [13].

Switches are used on railways to allow the trains to change lines. Switches are also controlled by the interlocking system and have to be on *normal* or *reverse* position. If the switch is on neither normal nor reverse position or the position indicators show both positions at the same time, it is assumed that the switch is on a faulty state.

Level crossings are intersection points where road traffic and rail traffic intersects [1]. They have three states: active, which means barrier is closed to allow train traffic, inactive, which means barrier is open to allow road traffic, and faulty, which means crossing is being opened or closed, or a fault has occurred [21].

IV. MODELLING OF SIGNALIZATION COMPONENTS

Basically, interlocking is decision-making software that mostly uses prohibitions on railway blocks. For example, in order to model a route reservation the designer have to consider too many conditions like position of switches, colours of signal lights that enables the driver to move on that route and also the colours of other signal lights that disables entrance of a reserved route, errors that might happen on a switch or a track circuit and locations of the trains on the railway yard. Some of the recommendations of EN50128 for designing safety critical software can be found in Figure 1.

An example railway yard and its related interlocking table are given in figure 2 and table 1, respectively. The design steps used in developing software can be seen in Figure 3.

Table A.4 – Software Design and Implementation (clause 10)

TECHNIQUE/MEASURE		Ref	SWS IL0	SWS IL1	SWS IL2	SWS IL3	SWS IL4
1.	Formal Methods including for example CCS, CSP, HOL, LOTOS, OBJ, Temporal Logic, VDM, Z and B	B.30	-	R	R	HR	HR
2.	Semi-Formal Methods	D.7	R	HR	HR	HR	HR
3.	Structured. Methodology including for example JSD, MASCOT, SADT, SDL, SSADM and Yourdon.	B.60	R	HR	HR	HR	HR

Figure 1. Part of EN50128 Software Design and Implementation Table (HR – Highly Recommended).

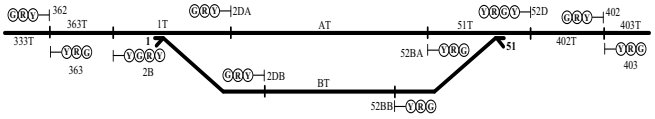

Figure 2. An example railway yard.

TABLE I. INTERLOCKING TABLE

Route Selection	Controlled Signal		Colour Signal Lights	Switch 1 Position
363T – AT	2B	Green	52BA (**Y**) or (**G**)	Normal
		Yellow	52BA (**R**)	
363T – BT		Yellow-Green	52BB (**Y**) or (**G**)	Opposite
		Yellow-Yellow	52BB (**R**)	

(Y-Yellow, G-Green and R-Red)

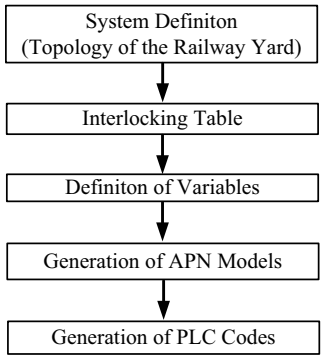

Figure 3. Development steps for interlocking software.

In this paper, we explained the generation of models, the last step of the design in figure 3 which is about software development can be found in [22].

While constituting an APN model for a switch the designer has to consider actual position of switch, incoming commands from TCC, possible switch errors like indication error or data accordance error. If no feedback is receiving from normal or reverse position sensors this means there is no indication and the switch is remained in the middle part of the track which causes indication error. Likewise, if both sensors are sending the same data (switch is on both normal and reverse position) this causes data accordance error. An example model of a switch is defined in Figure 4, 5, 6, 7 and 8. The definitions of places and some transitions are given on table 2.

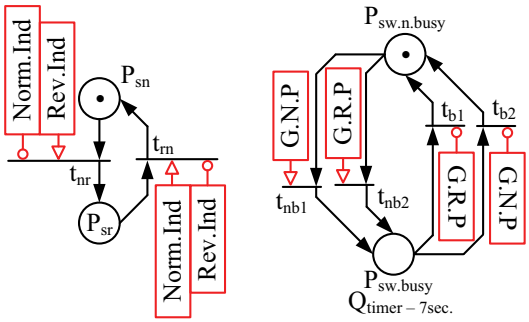

Figure 4. Switch can be on normal or on reverse position (left) and switch can be busy or not (right).

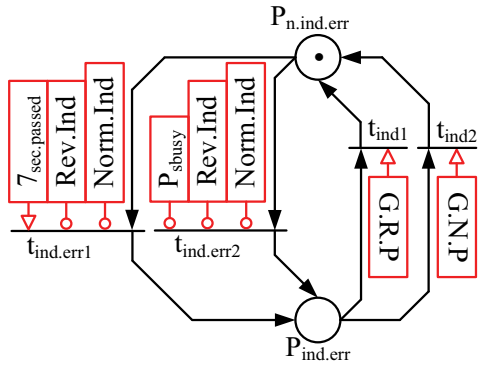

Figure 5. Indication error of a switch.

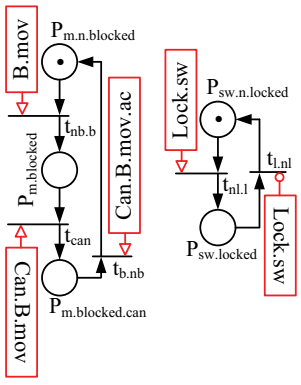

Figure 6. Switch movement can be blocked (left) and switch have to be locked if it is reserved (right).

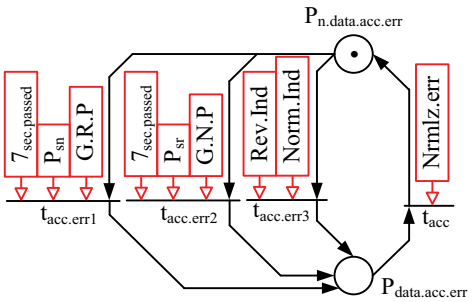

Figure 7. Data accordance error of a switch.

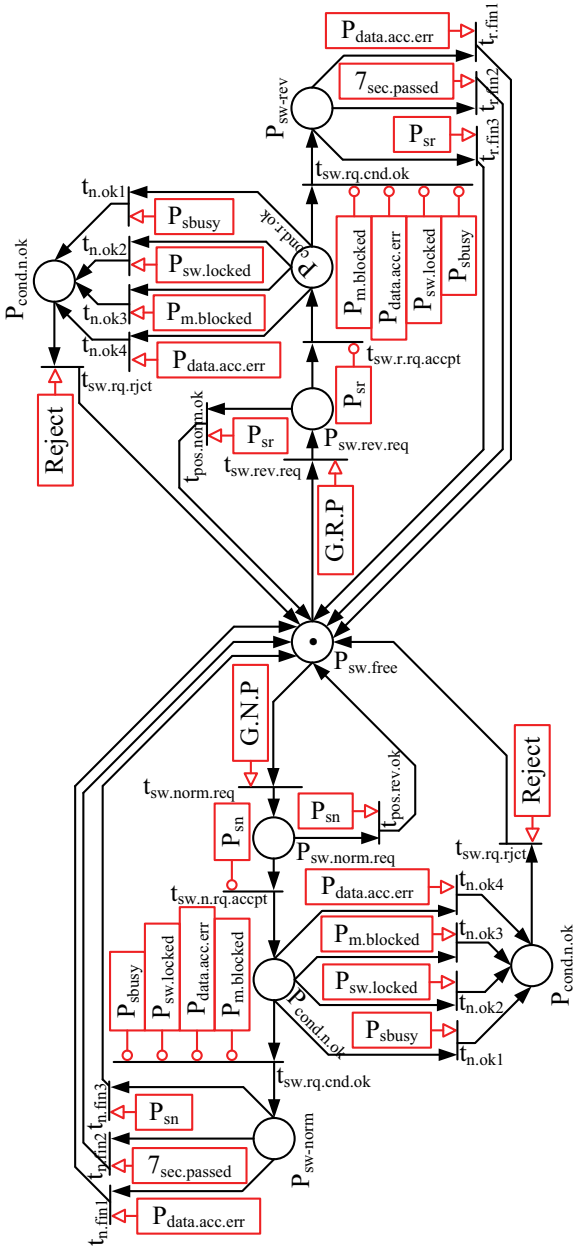

Figure 8. Movement of switch from normal to reverse and vice versa.

TABLE II. DEFINITIONS OF PLACES AND TRANSITIONS FOR SWITCH MODEL

Places	Definitions	Places	Definitions
P_{sn}	Sw is on normal position	$P_{cond.ok}$	Conditions are satsified
P_{sr}	Sw is on reverse position	$P_{cond.n.ok}$	Conditions are not satisfied
$P_{sw.n.busy}$	Sw is not busy	$P_{sw-norm}$	Sw is moving to normal position
$P_{sw.busy}$	Sw is busy	P_{sw-rev}	Sw is moving reverse position
$P_{n.ind.err}$	Sw has no indication Err	G.R.P	Go to Reverse Position
$P_{ind.err}$	Sw has indication Err	G.N.P	Go to Normal Position
$P_{m.n.blocked}$	Movement of Sw isn't blocked	Norm.Ind	Sw normal position indication
$P_{m.blocked}$	Movement of Sw is blocked	Rev.Ind	Sw reverse position indication
$P_{sw.n.locked}$	Sw is not locked	7sec.passed	Sw has to change its position in 7 seconds
$P_{sw.locked}$	Sw is locked	B.mov	Block movement of Sw
$P_{n.data.acc.err}$	Sw has no data accordance Err	Can.B.Mov	Cancel blocking
$P_{data.acc.err}$	Sw has data accordance Err	Can.B.mov.acc	Cancelling od blocking is accepted
$P_{sw.free}$	Sw doesn't have a position	Lock.sw	Lock Sw
$P_{sw.norm.req}$	Request came to Sw for normal position	Nrmlz err	Normalize Err
$P_{sw.rev.req}$	Request came to Sw for reverse position	Reject	Rejection of request
Transitions	**Definitions**	**Transitions**	**Definitions**
t_{rn}	Sw is moving from normal to reverse position	$t_{sw.rq.accpt}$	Sw position request is accepted
t_{nr}	Sw is moving from reverse to normal position	$t_{sw.rq.cnd.ok}$	Sw position conditions are accepted
$t_{ind.err}$	Indication Err	$t_{nl.l}$	Sw is locking
$t_{nb.b}$	Movement of Sw is blocking	$t_{l.nl}$	Sw is unlocking
$t_{b.nb}$	Movement of Sw is unblocking	t_{nb}	Sw is getting busy
t_{can}	Blocking on Sw is canncelling	t_{b}	Sw is getting not busy
$t_{acc.err}$	Accordance Err on Sw	$t_{sw.req.rjct}$	Sw position request is rejected
t_{acc}	Accordance Err on Sw is accepted	$t_{n.ok}$	Sw position conditions are not accepted
t_{ind}	Indication Err is disappeared	$t_{sw.req.}$	Sw position request

(Sw is used to indicate *Switch* and Err is used to indicate *Error*)

The components of a switch are modelled separately in order to achieve easy error tracking and visuality on the software development part which is given in [22]. For example in figure 4, position and occupation of the switch is modelled. In figure 5, possible indication errors, and in figure 6 blocking (movement of the switches can be blocked if necessary) and locking (switch have to be locked on a relevant position in a route reservation) of the switch are modelled. In figure 7, possible data accordance errors are modelled and lastly in figure 8, movement of a switch from one position to another is modelled. Rectangles represent sensor readings and some incoming commands from TCC. When there is no request on the switch it is assumed that it is on a free position (figure 8 place $P_{sw.free}$). After command *Go-Reverse-Poisition* - G.R.P (or *Go-Normal-Position* - G.N.P) token on $P_{sw.free}$ passes from transition $t_{sw.req}$ to place $P_{sw.rev.req}$. If switch is on desired position the token moves from $P_{sw.rev.req}$ to $P_{sw.free}$ by the transition $t_{sw.ok}$. If switch is not on the desired position then it has to reach to desired position in 7sec. If it does not do so the interlocking sends an *indication error*. While the switch is moving from one position to another there can be a possible error named *data accordance error* which means both position sensors sends 1 at the same time. Switches can also be moved by the dispatcher manually.

Likewise the model of a switch, a level crossing can be modelled as shown in figures 9-10:

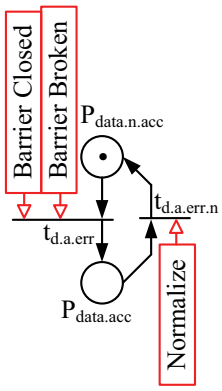

Figure 9. Data accordance error of level crossing.

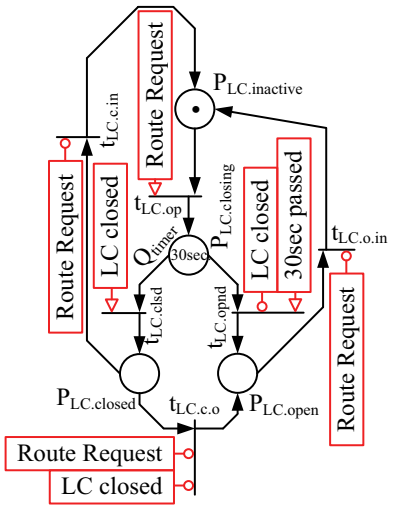

Figure 10. Opening the barrier of level crossing (LC).

The model given in figure 9 represents possible data accordance error on level crossing (if both sensors readings sends 1 at the same time it is assumed as an error by interlocking) and the model in figure 10 represents the opening and closing procedure of the level crossing. When there is no reservation or route request level crossing is always stays in open position (this is provided by sending inactive signal to level crossing) to allow road traffic (Token on $P_{LC.inactive}$). After a route request the inactive signal interrupted by interlocking and timer begin to count 30 sec. (token passes from $P_{LC.inactive}$ to $P_{LC.closing}$). If barrier of the level crossing is closed within 30 sec. interlocking assumes that there is no error and the level crossing is closed (token moves from $P_{LC.closing}$ to $P_{LC.closed}$). If it is not closed within 30 sec. interlocking assumes level crossing not closed and sends an error message. When the route reservation is ended the level crossing opens again by sending inactive signal from interlocking. Definitions of places and transitions are given in table 3.

TABLE III. DEFINITIONS OF PLACES AND TRANSITIONS FOR LEVEL CROSSING MODEL

Places	**Definitions**	**Transitions**	**Definitions**
$P_{data.n.acc}$	LC has no data accordance Err	$t_{d.a.err}$	LC data accordance Err conditions are met
$P_{data.acc}$	LC has data accordance Err	$t_{d.a.err.n}$	LC data accordance Err normalized
$P_{LC.inactive}$	LC is receiving inactive signal	$t_{LC.op}$	Open LC
$P_{LC.closing}$	Barrier of LC is closing	$t_{LC.clsd}$	LC is closed
$P_{LC.closed}$	Barrier of LC is closed	$t_{LC.opnd}$	LC is opened
$P_{LC.open}$	Barrier of LC is opened	$t_{LC.c.o}$	LC is closed then opened
30sec. passed	LC has to be close in 30 sec.	$t_{LC.c.in}$	LC is receiving inactive signal while its close
Normalize	Data accordance Err is normalized	$t_{LC.o.in}$	LC is receiving inactive signal while its open

(LC is used to indicate *Level Crossing* and Err is used to indicate *Error*)

RESULTS

In this study, modelling of railway equipments by Automation Petri Nets for interlocking and signalization design is achieved by considering the functional safety requirements recommended by CENELEC standards. Switch and level crossing PN models that include possible errors are considered. Considering possible error situations brings designers more comprehensive solutions. Modelling the components of signalization system also provides easy error tracking and increases the reliability and the compatibility of the developed models.

Obtained models can then be converted to a PLC code using Sequential Function Charts-SFC which is directly related with Petri Net models. These models are also converted to SFC codes and tested on a computer interface for various conditions.

ACKNOWLEDGMENT

This work is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) project number 108G186 – The National Railway Signalization Project.

REFERENCES

[1] S. Hall, “Modern Signalling Handbook,” Ian Allan Publishing, England, 2001.

[2] G. J. Kuepper, “150 Years of Train-Disasters - Practical Approaches for Emergency Responders,” 9-1-1 Magazine September/October issue, pp. 30-33, 1999.

[3] M. T. Söylemez, “Functional Safety Applications on Railway Systems: Turkish National Railway Signalization Project,” (in Turkish), 2nd International Industrial Safety Systems Conference, Istanbul, Turkey, 2010.

[4] M. T. Söylemez, M. S. Durmuş, U. Yıldırım, S. Türk and A. Sonat, “The Application of Automation Theory to Railway Signalization Systems: The Case of Turkish National Railway Signalization Project,” (*accepted for the IFAC World Congress 2011*).

[5] IEEE Standart Dictionary of Electrical and Electronic Terms.

[6] C. G. Cassandras and S. Lafortune, “Introduction to Discrete Event Systems”, Kluwer Academic Publishers, 1999.

[7] T. Murata, “Petri Nets: Properties, Analysis and Applications”, Proc. of IEEE, vol. 77, no. 4, pp. 541-580, 1989.

[8] M. Uzam, “Petri-net-based Supervisory Control of Discrete Event Systems and Their Ladder Logic Diagram Implementations,” PhD. Thesis, University of Salford, SALFORD, M5 4WT, UK, 1998.

[9] A. Giua, “Petri Net Techniques for Supervisory Control of Discrete Event Systems,” 1st Int. Workshop on Manufacturing and Petri Nets, Osaka, Japan, 1996.

[10] M. Uzam and A. H. Jones, “Discrete Event Control System Design Using Automation Petri Nets and Their Ladder Diagram Implementation,” The International Journal of Advanced Manufacturing Technology, vol. 14, no. 10, 1998, pp. 716-728.

[11] M. S. Durmuş, M. T. Söylemez, “Railway Signalization and Interlocking Design via Automation Petri Nets”, ASCC 2009, The 7th Asian Control Conference, Hong Kong Convention & Exhibition Center, 27-29 August, Hong Kong, 2009.

[12] M. S. Durmuş, M. T. Söylemez, E. Avşaroğulları, “Coloured Automation Petri Nets Based Interlocking and Signalization Design,”. The 6th IFAC International Workshop on Knowledge and Technology Transfer in/to Developing Countries - DECOM-IFAC-09. Ohridean Riviera, R. Macedonia, September 26-29, 2009.

[13] M. S. Durmuş, U. Yıldırım, A. Kursun, M. T. Söylemez, “Fail-Safe Signalization Design for a Railway Yard: A Level Crossing Case”, WODES'10, International Workshop on Discrete Event Systems, 30 August - 01 September, Berlin, Germany, , 2010.

[14] M. S. Durmuş, U. Yıldırım, M. T. Söylemez, “Signalization and Interlocking Design for a Railway Yard: A Supervisory Control Approach by Enabling Arcs,” The 7th International Symposium on Intelligent and Manufacturing Systems, IMS 2010, 15-17 September, Sarajevo, Bosnia Herzegovina, 2010.

[15] U. Yıldırım, M. S. Durmuş, M. T. Söylemez, “Fail-Safe Signalization and Interlocking Design for a Railway Yard: An Automation Petri Net Approach,” The 7th International Symposium on Intelligent and Manufacturing Systems, IMS 2010, 15-17 September, Sarajevo, Bosnia Herzegovina, 2010.

[16] A. Giua and C. Seatzu, “Modeling and Supervisory Control of Railway Networks Using Petri Nets,” IEEE Trans. On Automation Science and Engineering, vol. 5, no. 3, July 2008, , pp. 431-445.

[17] A. M. Hagalisletto, J. Bjork, I. C. Yu and P. Enger, “Constructing and Refining Large-Scale Railway Models Represented by Petri Nets,” IEEE Trans. On System, Man and Cybernetics-Part C: Applications and Reviews, vol. 37, no. 4, July, 2007, pp. 444-460.

[18] R. Zurawski and M. C. Zhou, “Petri Nets and Industrial Applications: A Tutorial,” IEEE Trans. on Industrial Electronics, vol. 41, no. 6, 1994, pp. 567-583.

[19] M. C. Zhou, V. Kurapati. Modeling, Simulation, & Control of Flexible Manufacturing Systems: A Petri Net Approach. World Scientific Publishing, 1998.

[20] J. Palmer, “The Need for Train Detection,” The 11th IET Professional Development Course on Railway Signalling and Control Systems, York, UK, 2006, pp. 47-53.

[21] S. Anandarao, C. D. Martland, “Level Crossing Safety on East Japan Railway Company: Application of Probabilistic Risk Assessment Techniques,” Transportation, vol(25), 1998, pp. 265-286.

[22] U. Yıldırım, M. S. Durmuş, M. T. Söylemez, “Application of Functional Safety on Railways Part II: Software Development,” (Accepted).