IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2015; **10**: 186–194
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.22052

Paper

Decision-Making Strategies in Fixed-Block Railway Signaling Systems: A Discrete Event Systems Approach

Mustafa Seçkin Durmuş*, Non-member
Shigemasa Takai**[a], Non-member
Mehmet Turan Söylemez***, Non-member

Ensuring the system safety at all times is most important in railway systems where small failures may result in a large number of casualties and property loss. Although there are various design methods, safety precautions, and recommendations of the railway-related safety standards, sometimes the occurrence of accidents cannot be prevented. In this paper, according to the recommendations of the railway-related safety standards, an interlocking system architecture, which consists of two controllers and a coordinator, for a fixed-block railway signaling system is studied. Based on the Petri net models of railway field components, decision-making strategies including fault diagnosis are presented. © 2014 Institute of Electrical Engineers of Japan. Published by John Wiley & Sons, Inc.

Keywords: safety-related standards, fixed-block railway signaling systems, fault diagnosis, Petri nets

Received 19 March 2014; Revised 10 July 2014

1. Introduction

From the safety-related standards point of view, fault diagnosis is regarded as the activity of checking whether a system is in a faulty state. To prevent the effect of incorrect results, fault diagnosis should be applied at the smallest subsystem level [1]. Especially for large and complex systems, diagnosis of faults becomes a critical and stringent task. Therefore, development of these systems has been guided by the safety-related standards such as IEC 61508. This standard is the functional safety standard for all electrical, electronic, and programmable electronic safety-related systems. EN 50126, EN 50128, EN 50129, and EN 50238 are some of the mandatory railway-related safety standards in addition to IEC 61508 [2]. Designers should take the requirements of these standards into account for developing the signaling system software (i.e., the interlocking software) for fixed-block railway signaling systems [3]. As described in Refs. [3] and [4], fixed-block railway systems have been in use since mid-1800s to provide safe travel and transportation all over the world. Since the interlocking software is a vital component of a fixed-block railway system, designers must follow the requirements of the related safety standards described above.

The fixed-block railway signaling systems can be evaluated from the perspective of discrete event systems (DESs) [5] since the DES modeling methods are regarded as semi-formal methods in Tables A.1, A.2, and A.3 in Ref. [6] and Table A.2 in Ref. [7]. These modeling techniques are also highly recommended to design SIL3 or SIL4 interlocking software, where SIL stands for safety integrity level.

[a] Correspondence to: Shigemasa Takai.
E-mail: takai@eei.eng.osaka-u.ac.jp

* Electrical and Electronics Engineering Department, Pamukkale University, Kinikli, 20070, Denizli, Turkey
** Division of Electrical, Electronic and Information Engineering, Osaka University, 2-1 Yamada-Oka, Suita, Osaka 565-0871, Japan
*** Control Engineering Department, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey

The study of Sampath *et al*. is a pioneering automata-based work on failure diagnosis of DESs [8,9]. Ushio *et al*. [10] extended the diagnosability condition of Sampath *et al*. to unbounded Petri nets. In addition to theoretical studies, the fault diagnosis approach is applied to industrial systems [11–13]. Online monitoring using the time Petri net modeling technique has been studied for a railway system [14]. In Ref. [15], fixed-block signaling system components are modeled by Petri nets and a diagnoser is designed to show diagnosability of the system. It is also important to note that the definitions of fault and failure are slightly different in the safety standards, but in this study both of them are used to mean that the system does not work as desired.

According to the software design steps mentioned in the V-model [6], designers should choose approved combinations of software architectures from Table A.3 of Ref. [7] to achieve the required SIL, which should be at least at 3 for railway applications [16]. For instance, the use of Petri nets and finite state machines is highly recommended in Table A.18 of [7] as a semi-formal modeling method. In this paper, use is made of Petri nets and the combination of defensive programming, diverse programming, and failure assertion programming architectures as recommended to develop SIL3 software. The purpose of defensive programming is to consider the worst case from all inputs and respond to it in a predetermined and plausible way. Input variables and the effect of output variables should be checked, the coding standards should be used, and the code should be as simple as possible [17]. The main purpose of failure assertion programming is to detect software design faults while executing a program and continue the operation for high reliability [1]. The main aim of diverse programming is to develop N different program versions for the given input–output specifications. These different versions should be developed by different workgroups so that they do not fail at the same time because of the same reason. These different versions are combined together under a coordinator (i.e., a voter) where their responses are subjected to a voting operation. Diverse programming does not overcome possible software design faults, but this method enables us to handle unpredictable and unknown design faults, prevents system failure, and provides the continuity of the system operation in a safe way [1].

Detailed definitions of different voting strategies can be found in Refs. [18–22]. If the system is not fail-safe (where the safe state of the system is not predetermined), then generalized voting strategies can be used, and generally N is chosen as 3 [23]. By contrast, as mentioned in Ref. [7, Sec. B.17], if the system has a safe state (or the system is fail-safe), then it is feasible to demand complete agreement; in other words, complete agreement of the program versions can be sought before getting into an unsafe state. In this case, typically N is chosen as 2 [7,22].

In Ref. [22], according to the recommendations of the railway-related safety standards, an interlocking system architecture, which consists of two controllers (called modules in Ref. [22]) and a coordinator (called a voter in Ref. [22]), is proposed for a fixed-block railway signaling system, and certain synchronization problems between controllers are addressed. In this paper, based on the Petri net models of railway field components, decision-making strategies including fault diagnosis are presented in the interlocking system architecture of Ref. [22]. Since the use of conventional Petri net models is enough to achieve the purpose of control and diagnosis considered in this paper, and using time Petri net models makes the analysis more complicated, conventional Petri net models are used in this paper.

The paper is organized as follows. In Section 2, some preliminary definitions are given. The definitions of the basic components of fixed-block railway signaling systems are given in Section 3. The control architecture and the working principle of the controllers are explained in Section 4. The decision-making strategies of the controllers and the coordinator are discussed in Section 5. Finally, the paper ends with a brief conclusion in Section 6.

2. Preliminaries

2.1. Petri net A Petri net is defined in Ref. [24] as

$$PN = (P, T, F, W, M_0), \tag{1}$$

where

- $P = \{p_1, p_2, \ldots, p_k\}$ is the finite set of places,
- $T = \{t_1, t_2, \ldots, t_z\}$ is the finite set of transitions,
- $F \subseteq (P \times T) \cup (T \times P)$ is the set of arcs,
- $W: F \to \{1, 2, 3, \ldots\}$ is the weight function,
- $M_0: P \to \{0, 1, 2, 3, \ldots\}$ is the initial marking,
- $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

We use $I(t_j)$ and $O(t_j)$ to represent the sets of input places and output places of transition t_j, respectively, as

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in F\}, \tag{2}$$

$$O(t_j) = \{p_i \in P : (t_j, p_i) \in F\} \tag{3}$$

For a marking $M : P \to \{1, 2, 3, \ldots\}$, $M(p_i) = n$ means that the ith place has n tokens [24]. A marking M can also be represented by a vector with k elements, where k is the total number of places.

Definition 1 [5]: A transition t_j is said to be enabled at a marking M if each input place p_i of t_j has at least $W(p_i, t_j)$ tokens, where $W(p_i, t_j)$ is the weight of the arc from place p_i to transition t_j, that is, $M(p_i) \geq W(p_i, t_j)$ for all $p_i \in I(t_j)$.

Note that if $I(t_j) = \emptyset$, transition t_j is always enabled. An enabled transition may or may not fire (depending on whether or not the event actually takes place). The firing of an enabled transition t_j removes $W(p_i, t_j)$ tokens from each $p_i \in I(t_j)$ and adds $W(t_j, p_i)$ tokens to each $p_i \in O(t_j)$, where $W(t_j, p_i)$ is the weight of the arc from t_j to p_i. That is

$$M'(p_i) = M(p_i) - W(p_i, t_j) + W(t_j, p_i), \tag{4}$$

where $M'(p_i)$ is the number of tokens in the ith place after the firing of transition t_j, and we let $W(p_i, t_j) = 0$ if $(p_i, t_j) \notin F$ and $W(t_j, p_i) = 0$ if $(t_j, p_i) \notin F$. The notation $M[t_j>$ denotes that a transition t_j is enabled at a marking M. Also, $M[t_j>M'$ denotes that, after the firing of t_j at M, the resulting marking is M'. These notations can be extended to a sequence of transitions.

Definition 2 [24]: A Petri net PN is said to be *pure* if it has no self-loops and said to be *ordinary* if all of its arc weights are 1.

Definition 3 [24]: A marking M_n is reachable from the initial marking M_0 in a Petri net PN if there exists a sequence of transitions $t_1 t_2 \ldots t_n$ such that $M_0[t_1>M_1[t_2> \ldots M_{n-1}[t_n>M_n$, and $R(M_0)$ denotes the set of all reachable markings from M_0.

Definition 4 [24]: A Petri net PN is said to be m-bounded if the number of tokens in each place does not exceed a finite number m, that is, $\forall M_k \in R(M_0)$, $\forall p_i \in P : M_k(p_i) \leq m$. Additionally, PN is *safe* if it is 1-bounded.

Definition 5 [24]: A Petri net PN is said to be *deadlock-free* (complete absence of deadlocks) if at least one transition is enabled at every reachable marking $M_k \in R(M_0)$.

The set P of places is partitioned into the set P_o of observable places and the set P_{uo} of unobservable places [10]. Similarly, the set T of transitions is partitioned into the set T_o of observable transitions and the set T_{uo} of unobservable transitions. That is

$$P = P_o \cup P_{uo} \text{ and } P_o \cap P_{uo} = \emptyset, \tag{5}$$

$$T = T_o \cup T_{uo} \text{ and } T_o \cap T_{uo} = \emptyset. \tag{6}$$

Because of the existence of unobservable places, some markings cannot be distinguished. If the observations of markings M_1 and M_2 are the same, that is, if $M_1(p_i) = M_2(p_i)$ for any $p_i \in P_o$, it is denoted by $M_1 \equiv M_2$. The quotient set with respect to the equivalence relation ($\equiv$) is denoted by $\hat{R}(M_0) := \hat{R}(M_0)/\equiv := \{\hat{M}_0, \ldots, \hat{M}_n, \ldots\}$, where $M_0 \in \hat{M}_0$ [25]. An element of $\hat{R}(M_0)$ is referred to the observation of a marking or an observable marking.

A subset T_F of T_{uo} represents the set of faulty transitions. It is assumed that there are n different failure types, and $\Delta_F = \{F_1, F_2, \ldots, F_n\}$ is the set of failure types. That is

$$T_F = T_{F_1} \cup T_{F_2} \cup \cdots \cup T_{F_n} \tag{7}$$

where $T_{F_i} \cap T_{F_j} = \emptyset$ if $i \neq j$. The label set is defined as $\Delta = \{N\} \cup 2^{\Delta_F}$, where N denotes the label 'normal', which indicates that no faulty transition has fired, and 2^{Δ_F} denotes the power set of Δ_F, that is, 2^{Δ_F} is the set of all subsets of Δ_F.

2.2. Failure diagnosis The concept of failure diagnosis from the DES point of view is to detect the firing of some unobservable faulty transition in T_F by checking the observed behavior of the system. For simplicity, we impose the following two assumptions:

Assumption 1 [8,10]: A Petri net system PN defined by (1) is *bounded* and *deadlock-free*.

Assumption 2 [8,10]: There does not exist a sequence of unobservable transitions whose firing generates a cycle of markings which have the same observation, that is, for any $M_i \in R(M_0)$ and $t_i \in T_{uo}$, $(i = 1, 2, \ldots, n)$, $M_1[t_1>M_2[t_2> \ldots M_n[t_n>M_1 \Rightarrow \exists i, j \in \{1, 2, \ldots, n\}: M_i \not\equiv M_j$.

The notation $Q = 2^{R(M_0)\times\Delta}$ denotes the power set of the Cartesian product $R(M_0) \times \Delta$, that is, each element of Q is a subset of $R(M_0) \times \Delta$ and is of the form $\{(M_1, l_1), (M_2, l_2), \ldots, (M_n, l_n)\}$. The diagnoser is an automaton given by

$$G_d = (Q_d, \Sigma_o, \delta_d, q_0) \tag{8}$$

where $Q_d \subseteq Q$ is the set of states, $\Sigma_o = \hat{R}(M_0) \cup T_o$ is the set of events, $\delta_d : Q_d \times \Sigma_o \to Q_d$ is the partial state transition function, and $q_0 = \{(M_0, N)\}$ is the initial state. The state set Q_d is the set of states in Q that are reachable from the initial state q_0 under the state transition function δ_d. Each observed event $\sigma_o \in \Sigma_o$ represents the observation of a marking in $\hat{R}(M_0)$ or an observable transition in T_o. The transition function δ_d is defined using the label propagation function and the range function. The label propagation function $LP : R(M_0) \times \Delta \times T^* \to \Delta$ propagates the label (normal or faulty) over a sequence $s \in T^*$ of transitions, where T^* is the set of all finite sequences of elements of T, as follows [8,10]:

$$LP(M,l,s) = \begin{cases} N, & \text{if } (l = N) \wedge (\forall F_i \in \Delta_F : T_{F_i} \notin s) \\ \{F_i : F_i \in l \vee T_{F_i} \in s\}, & \text{otherwise,} \end{cases} \tag{9}$$

where $T_{F_i} \in s$ (respectively, $T_{F_i} \notin s$) indicates that a sequence $s \in T^*$ of transitions contains (respectively, does not contain) a faulty transition with failure type F_i. Then, the range function $LR : Q \times \Sigma_o \to Q$ is defined as follows [8,10]:

$$LR(q, \sigma_o) = \bigcup_{(M,l) \in q} \bigcup_{s \in T^*(M, \sigma_o)} \{(M', LP(M,l,s))\}, \tag{10}$$

where $M[s>M'$, and $T^*(M, \sigma_o) \subseteq T^*$ is the set of possible transition sequences from M which are consistent with the observed event σ_o and defined as follows [15]:

1. If $\sigma_o \in \hat{R}(M_0)$,

$$T^*(M, \sigma_o) = \begin{cases} \emptyset, & \text{if } M \in \sigma_o \\ \left\{ s \in T^*_{uo} : \begin{array}{l} (M_s \in \sigma_o) \\ \wedge (\forall s' (\neq s) \in \bar{s} : M_{s'} \equiv M) \end{array} \right\}, & \text{otherwise,} \end{cases} \tag{11}$$

   where $M[s>M_s$, $M[s'>M_{s'}$, and $\bar{s}$ denotes the set of all prefixes of s.
2. If $\sigma_0 \in T_o$,

$$T^*(M, \sigma_o) = \left\{ \begin{array}{l} s \in T^*_{uo} \cdot \{\sigma_o\} \\ \quad : (M[s>) \wedge (\forall s' (\neq s) \in \bar{s} : M_{s'} \equiv M), \end{array} \right\} \tag{12}$$

   where $M[s'>M_{s'}$.

Finally, the transition function $\delta_d : Q_d \times \Sigma_o \to Q_d$ is defined as follows [8,10]:

$$\delta_d(q, \sigma_o) = \begin{cases} LR(q, \sigma_o), & \text{if } LR(q, \sigma_o) \neq \emptyset \\ \text{undefined}, & \text{otherwise.} \end{cases} \tag{13}$$

A state q_d of the diagnoser is of the form $q_d = \{(M_1, l_1), (M_2, l_2), \ldots, (M_n, l_n)\}$, which consists of pairs of a marking $M_i \in R(M_0)$ and a label $l_i \in \Delta$. Each M_i represents a possible marking that is consistent with the past observation. If $l_i = N$, the marking M_i is reached without the firing of any faulty transition. By contrast, $F_i \in l_i$ indicates that a faulty transition of failure type F_i fired before M_i is reached.

A system is said to be diagnosable if the type of the fault is always detected within a uniformly bounded number of firings of transitions after the occurrence of the fault [10]. It is possible to classify states in Q_d as follows:

(1) A state $q \in Q_d$ is said to be F_i-certain if $F_i \in l$ for any$(M, l) \in q$.
(2) A state $q \in Q_d$ is said to be F_i-uncertain if there exist (M, l) and $(M', l') \in q$ such that $F_i \in l$ and$F_i \notin l'$. (With a slight abuse of notation, we let $F_i \notin N$.)

If the system is diagnosable, then, after the occurrence of a faulty transition (e.g., $t \in T_{F_i}$), the state of the diagnoser reaches an F_i-certain state within a finite number of firings of transitions [10]. A set $\{q_1, q_2, \ldots, q_n\} \subseteq Q_d$ of F_i-uncertain states are named an F_i-indeterminate cycle [8,10,15] if the following two conditions hold:

(1) The states $q_1, q_2, \ldots, q_n \in Q_d$ constitute a cycle in G_d, that is, there exist $\sigma_1, \sigma_2, \ldots, \sigma_n \in \Sigma_o$ such that, $\delta_d(q_j, \sigma_j) = q_{j+1}$ for each $j = 1, 2, \ldots, n-1$ and $\delta_d(q_n, \sigma_n) = q_1$;
(2) For each $j = 1, 2, \ldots, n, k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$, there exist $(M_j^k, l_j^k), (\tilde{M}_j^r, \tilde{l}_j^r) \in q_j$ which satisfy the following two conditions: In the second condition (b), $M_j^k (j = 1, 2, \ldots, n,\ k = 1, 2, \ldots, m)$ constitute a cycle involving nm markings that carry F_i in their labels, whereas the markings $\tilde{M}_j^r (j = 1, 2, \ldots, n,\ r = 1, 2, \ldots, m')$ constitute a cycle involving nm' markings that do not carry F_i in their labels. We use m and m' to indicate the number of times the cycle $q_1, q_2, \ldots, q_n \in Q_d$ is completed in G_d before the cycles of M_j^k and $\tilde{M}_j^r$ are completed, respectively [8]. Thus, k (respectively, r) is used to denote that the kth (respectively, rth) cycle $q_1, q_2, \ldots, q_n \in Q_d$ is executed in G_d.
    (a) For any $j = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$, and $r = 1, 2, \ldots, m'$, $F_i \in l_j^k$ and $F_i \notin \tilde{l}_j^r$.
    (b) Markings $M_j^k (j = 1, 2, \ldots, n,\ k = 1, 2, \ldots, m)$ and $\tilde{M}_j^r$ $(j = 1, 2, \ldots, n,\ r = 1, 2, \ldots, m')$ satisfy the following conditions:

$$\begin{aligned} &\exists s_j^k \in T^*(M_j^k, \sigma_j) : M_j^k[s_j^k > M_{j+1}^k (j = 1, 2, \ldots, n-1, \\ &\qquad\qquad k = 1, 2, \ldots, m), \\ &\exists s_n^k \in T^*(M_n^k, \sigma_n) : M_n^k[s_n^k > M_1^{k+1} (k = 1, 2, \ldots, m-1), \\ &\exists s_n^m \in T^*(M_n^m, \sigma_n) : M_n^m[s_n^m > M_1^1, \\ &\exists \tilde{s}_j^r \in T^*(\tilde{M}_j^r, \sigma_j) : \tilde{M}_j^r[\tilde{s}_j^r > \tilde{M}_{j+1}^r (j = 1, 2, \ldots, n-1, \\ &\qquad\qquad r = 1, 2, \ldots, m'), \\ &\exists \tilde{s}_n^r \in T^*(\tilde{M}_n^r, \sigma_n) : \tilde{M}_n^r[\tilde{s}_n^r > \tilde{M}_1^{r+1} (r = 1, 2, \ldots, m'-1), \\ &\exists \tilde{s}_n^{m'} \in T^*(\tilde{M}_n^{m'}, \sigma_n) : \tilde{M}_n^{m'}[\tilde{s}_n^{m'} > \tilde{M}_1^1. \end{aligned}$$

Theorem 1 [8,10]: A Petri net system PN is diagnosable if and only if the diagnoser given by (8) does not contain an F_i-indeterminate cycle for any failure type F_i.

3. Basic Components of Fixed-Block Signaling Systems

The *traffic control center* is the dispatcher's office from where all operations and railway traffic are commanded. Route (RT) reservation requests or any other requests related with the railway field components are executed by the traffic control center because the trains can move from one station to another in accordance with the route reservation procedure.

The *interlocking software* receives requests of the traffic control center for evaluation. After evaluation, the requests are accepted if the results of the requests will not cause a dangerous situation; otherwise they are denied immediately. The interlocking software is regarded as the vital component of a signaling system which ensures safe travel and transportation.

Railway blocks (RBs) can be regarded as subsections of railway lines which start and end with a wayside signal explained below. As a basic principle of fixed-block railway signaling systems, only one train is allowed in each railway block at the same time. Track circuits or axle counters can be used for train detection in RBs.

Wayside signals (WSs) allow train drivers to have information about their destinations. Although the use of colors differs from country to country, in general the red color stands for occupation of the next RB; the yellow color signifies that the next RB is free but not the RB after the next. Finally, the green color implies that the next two RBs are unoccupied.

Point machines (PMs) are established in certain locations where track change is needed. PM positions can be controlled either by the officers manually in the railway field or from the control center. The positions of related PMs are also adjusted automatically in case of route reservation.

More detailed definitions of the components of fixed-block railway signaling systems can be found in Refs. [3,26,27].

4. Control Architecture

The control architecture studied in this paper is given in Fig. 1 [22]. We use N-version programming in this control architecture to satisfy the requirements of the safety standards for developing SIL3 software. Since railway signaling systems can be classified as safety-critical, the signaling system has to be fail-safe and N is chosen as 2. We assume that there is a reliable communication (e.g., safe-ethernet) between the coordinator, the control center, and the controllers.

In Fig. 1, the system block represents the railway field. The components in the railway field are desired to be controlled by using two parallel running controllers (e.g., programmable logic controllers) which are assumed to be fail-safe. These controllers are designed by two independent workgroups according to diverse programming, and they do not communicate with each other.

The structure of a controller is shown in Fig. 2. Each block in the controller is the software block for each railway field component and each route in the railway topology. For every railway field component, a single Petri net model is constructed and added to the corresponding software block. The software block also includes the diagnoser and failure recovery module for the component (e.g., a PM block contains the Petri net model, diagnoser, and failure recovery module for the PM).

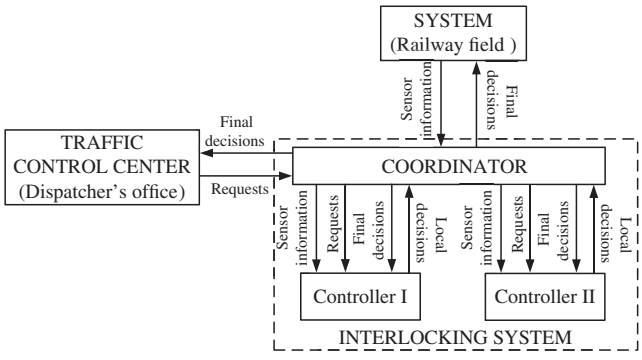

Fig. 1. Control architecture

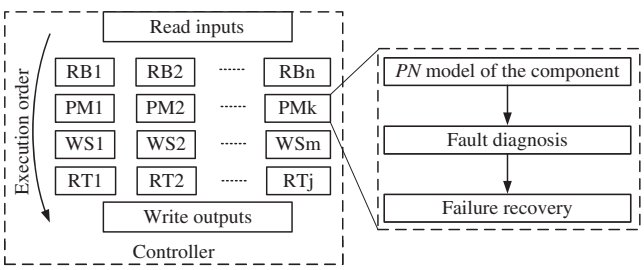

Fig. 2. Structure of a controller

As previously mentioned in Section 1, the use of defensive and failure assertion programming techniques should be implemented in the software block while constructing the Petri net models. For instance, as a general rule that has been accepted all over the world, movement of a PM must be rejected whenever its railway block is occupied. This rule should be taken into account while constructing the Petri net models by the designers as an application of the defensive programming technique. Moreover, as an application of the failure assertion programming technique, position information of a PM and color information of a signal should be checked before and after an incoming request from the control center to ensure it is functioning correctly.

For the purpose of failure diagnosis, we assume that the Petri net model of each railway field component satisfies Assumptions 1 and 2 imposed in Section 2.2. For simplification, we do not address the issue of failure recovery in this paper.

The coordinator receives the requests from the control center and the sensor information from the railway field. The requests of the control center are sent to the controllers immediately, whereas the sensor information are sent to the controllers periodically (e.g., 2 s). Each controller evaluates the requests according to the common knowledge (e.g., a table where the safety precautions are determined), arrives at its decision, and then sends it back to the coordinator. The coordinator gives final decisions by comparing the controllers' decisions using simple logical operations. It is important to note that the controllers have different cycle times and evaluation strategies because of diverse designs [23]. Because of this asynchronous nature, they do not receive the requests from the coordinator simultaneously. Similarly, the coordinator does not receive the decisions of the controllers at the same time.

If one controller detects a fault in a railway field component but the other controller does not, the former reports this fault information to the coordinator immediately and the coordinator sends the fault information to both the traffic control center and the other controller which did not detect the fault. Then, the controller that did not detect the fault acts as there is a faulty condition in the related railway field component (e.g., the controller denies the incoming requests related with the faulty field component).

5. Design of the Coordinator and the Controllers

5.1. Possible failures

Because of the high level of risk involved, railway signaling systems have strict rules. The interlocking system warns the traffic control center or stops the whole system depending on the failure. If the stuck of a PM occurs during a route reservation procedure, the route request should be rejected and information about the failure should be sent to the control center. It is not necessary to stop the whole system in case of such a failure.

In Turkey, most of the railways are rather old compared to those in the other European countries. Especially, most of the equipment used in the old fixed-block railway lines are old, and faults can occur easily in bad weather conditions such as cold and rain. For example, in a cold day, the blades of these old PMs can stick to each other. Similarly, when there is heavy rain, the blades of the PMs can stop in the middle of the movement because of electrical malfunction. Therefore, while developing interlocking software for old railway lines, one of the main requirements of the Turkish State Railways is to detect these kinds of faults.

We suppose that, while a train is moving in the pre-reserved route for itself, due to a faulty decision of the controller(s), the position of the PM changes even if the railway block of the point machine is occupied. As a result, the cars of the train will derail. This situation is illustrated in Fig. 3. The interlocking software must be designed so that this situation can be avoided.

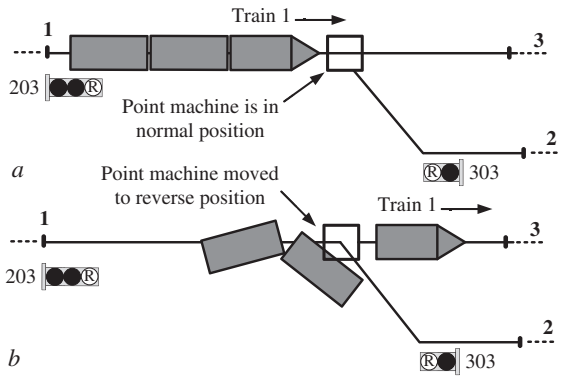

Fig. 3. Derailment of a train

5.2. Petri net models and their diagnosers For each railway field component, its Petri net models are constructed by two different workgroups. Since two controllers obtain the same sensor information on the railway field, we assume that the constructed Petri net models have common observable places and common observable transitions. Two different Petri net models of a PM are shown in Fig. 4. Note that unobservable places and transitions are indicated in Fig. 4 as striped places and transitions.

Representations of the Petri net models shown in Fig. 4(a) and (b) are as follows, respectively.

$$
\begin{aligned}
P_{\mathrm{c1\_}o} &= \{p_{\mathrm{c1\_1}}, p_{\mathrm{c1\_2}}, p_{\mathrm{c1\_5}}, p_{\mathrm{c1\_6}}\},\\
P_{\mathrm{c1\_}uo} &= \{p_{\mathrm{c1\_3}}, p_{\mathrm{c1\_4}}, p_{\mathrm{c1\_7}}, p_{\mathrm{c1\_8}}, p_{\mathrm{c1\_}f1}, p_{\mathrm{c1\_}f2}\},\\
T_{\mathrm{c1\_}o} &= \{t_{\mathrm{c1\_1}}, t_{\mathrm{c1\_2}}, t_{\mathrm{c1\_3}}, t_{\mathrm{c1\_4}}, t_{\mathrm{c1\_5}}, t_{\mathrm{c1\_6}},\\
&\quad t_{\mathrm{c1\_7}}, t_{\mathrm{c1\_8}}, t_{\mathrm{c1\_9}}, t_{\mathrm{c1\_10}}, t_{\mathrm{c1\_11}}, t_{\mathrm{c1\_12}}\},\\
T_{\mathrm{c1\_}uo} &= \{t_{\mathrm{c1\_}f1}, t_{\mathrm{c1\_}f2}, t_{\mathrm{c1\_}f3}, t_{\mathrm{c1\_}f4}, t_{\mathrm{c1\_}f5}, t_{\mathrm{c1\_}f6}, t_{\mathrm{c1\_}f7}, t_{\mathrm{c1\_}f8}\},\\
M_{\mathrm{c1\_0}} &= (M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_1}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_2}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_3}}),\\
&\quad M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_4}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_5}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_6}}),\\
&\quad M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_7}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_8}}), M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_}f1}),\\
&\quad M_{\mathrm{c1\_0}}(p_{\mathrm{c1\_}f2})) = (0,1,\underline{0},\underline{0},0,0,\underline{0},\underline{0},\underline{0},\underline{0}).\\
P_{\mathrm{c2\_}o} &= \{p_{\mathrm{c2\_1}}, p_{\mathrm{c2\_2}}, p_{\mathrm{c2\_5}}, p_{\mathrm{c2\_6}}\},\\
P_{\mathrm{c2\_}uo} &= \{p_{\mathrm{c2\_3}}, p_{\mathrm{c2\_4}}, p_{\mathrm{c2\_7}}, p_{\mathrm{c2\_}f1}, p_{\mathrm{c2\_}f2}\},\\
T_{\mathrm{c2\_}o} &= \{t_{\mathrm{c2\_1}}, t_{\mathrm{c2\_2}}, t_{\mathrm{c2\_3}}, t_{\mathrm{c2\_4}}, t_{\mathrm{c2\_5}}, t_{\mathrm{c2\_6}}, t_{\mathrm{c2\_7}},\\
&\quad t_{\mathrm{c2\_8}}, t_{\mathrm{c2\_9}}, t_{\mathrm{c2\_10}}\},\\
T_{\mathrm{c2\_}uo} &= \{t_{\mathrm{c2\_}f1}, t_{\mathrm{c2\_}f2}, t_{\mathrm{c2\_}f3}\},\\
M_{\mathrm{c2\_0}} &= (M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_1}}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_2}}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_3}}),\\
&\quad M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_4}}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_5}}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_6}}),\\
&\quad M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_7}}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_}f1}), M_{\mathrm{c2\_0}}(p_{\mathrm{c2\_}f2}))\\
&= (0,1,\underline{0},\underline{0},0,0,\underline{1},\underline{0},\underline{0}).
\end{aligned}
$$

The meanings of the places and the transitions in these models are given in Table I. Note that in any marking, underlines are used to indicate unobservable places. It is assumed that there are three different failure types $\Delta_F = \{F_1, F_2, F_3\}$, where $T_{F_1} = \{t_{c1\_f1}, t_{c1\_f4}\}$, $T_{F_2} = \{t_{c1\_f2}, t_{c1\_f3}\}$, and $T_{F_3} = \{t_{c1\_f5},\ t_{c1\_f6}, t_{c1\_f7},\ t_{c1\_f8}\}$ for the Petri net in Fig. 4(a) and $T_{F_1} = \{t_{c2\_f1}\}$, $T_{F_2} = \{t_{c2\_f2}\}$, and $T_{F_3} = \{t_{c2\_f3}\}$ for the Petri net in Fig. 4(b). F_1 and F_2 mean that a PM does not reach its desired position in 7 s while moving to reverse and normal position, respectively. F_3 means that the blades of the PM are stuck. If the actual position (possibly the initial position) of a PM does not change in a predefined time instance after an incoming PM position request, it is assumed that the blades of the PM are stuck.

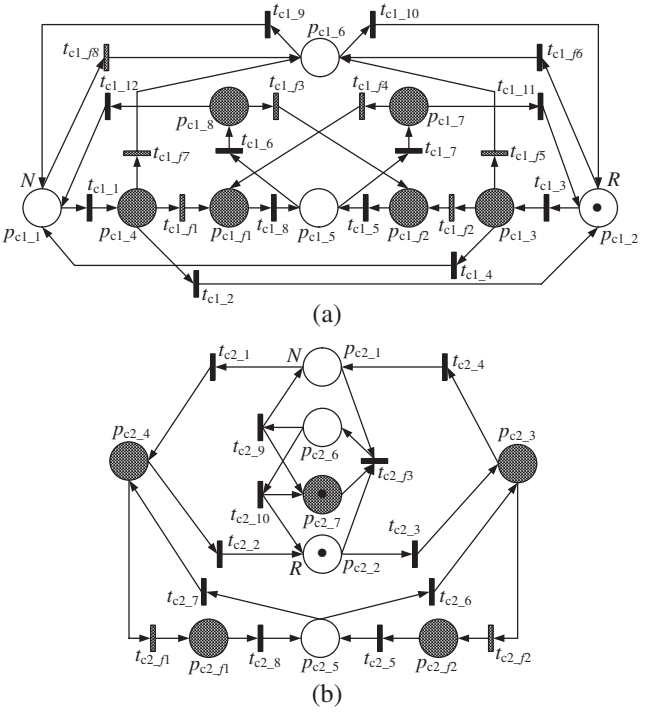

Fig. 4. Different Petri net models of a PM

The Petri net models in Fig. 4 are slightly different from each other because of the difference between the design strategies of two independent workgroups. Assume that the PM is in the reverse position (the tokens are in the places $p_{\mathrm{c1\_2}}$ and $p_{\mathrm{c2\_2}}$, respectively) and by an incoming position request from the coordinator, the tokens are moved to the places $p_{\mathrm{c1\_3}}$ and $p_{\mathrm{c2\_3}}$ by the firing of $t_{\mathrm{c1\_3}}$ and $t_{\mathrm{c2\_3}}$, respectively. In the Petri net model given in Fig. 4(a), the token can move to the place $p_{\mathrm{c1\_}f2}$ by the faulty transition $t_{\mathrm{c1\_}f2}$ or the place $p_{\mathrm{c1\_1}}$ by the observable transition $t_{\mathrm{c1\_4}}$. Although it is likely to occur, the token in the place $p_{\mathrm{c1\_3}}$ can move to the place $p_{\mathrm{c1\_6}}$ by the faulty transition $t_{\mathrm{c1\_}f5}$. By contrast, this failure situation is not considered by the other workgroup. If the last-mentioned situation occurs, only the first controller can detect the occurrence of the fault ($t_{\mathrm{c1\_}f5}$). As mentioned before, the main purpose of the diverse programming is to detect the design faults and prevent the system.

Parts of the diagnosers, defined in Section 2.2, for the Petri net models given in Fig. 4 are shown in Fig. 5. Each state represented by a rectangle includes a pair of marking of places and a label N or F_i ($i \in \{1,2,3\}$). That is, in these parts of the diagnosers, a marking just after an observed event is uniquely determined. Besides, multiple failures are not dealt with in this paper for simplification.

Each state transition of the diagnoser is labeled by the observation of a marking or a pair of observations of a marking and an observable transition with a slight abuse of notation. (According to the definition of (8), each state transition is labeled by the observation of a marking or an observable transition.)

For instance, at the initial state of the diagnoser in Fig. 5(a), the event label $\widehat{M}\,1\_3$ represents that the observable marking $\widehat{M}\,1\_3$ is observed by the firing of the unobservable transition $t_{\mathrm{c1\_}f6}$. Similarly, the state of the diagnoser shown in Fig. 5(b) changes from $\{((0,0,\underline{1},\underline{0},0,0,\underline{1},\underline{0},\underline{0}),N)\}$ to $\{((1,0,\underline{0},\underline{0},0,0,\underline{1},\underline{0},\underline{0}),N)\}$ by the firing of the observable transition $t_{\mathrm{c2\_4}}$ with the observation $\widehat{M}\,2\_2$ of the resulting marking. According to Theorem 1 in Section 2, since there is no F_i-indeterminate cycle in the diagnosers, both Petri net models are diagnosable. The diagnosability

Table I. Meanings of places and transitions in the models shown in Fig. 4

Place	Meaning	Transition	Meaning
p_{c1_1} (p_{c2_1})	PM is in normal position	t_{c1_1} (t_{c2_1})	PM left normal position
p_{c1_2} (p_{c2_2})	PM is in reverse position	t_{c1_2}, t_{c1_11}, (t_{c2_2})	PM reached reverse position
p_{c1_3} (p_{c2_3})	PM is moving from reverse position to normal position	t_{c1_3} (t_{c2_3})	PM left reverse position
p_{c1_4} (p_{c2_4})	PM is moving from normal position to reverse position	t_{c1_4}, t_{c1_12}, (t_{c2_4})	PM reached normal position
p_{c1_5} (p_{c2_5})	PM did not reach desired position	t_{c1_5}, t_{c1_8} (t_{c2_5},t_{c2_8})	7 s passed
p_{c1_6} (p_{c2_6})	PM stuck fault has occurred	t_{c1_6} (t_{c2_6})	Move PM to normal position
p_{c1_7}	PM is moving to reverse position after fault	t_{c1_7} (t_{c2_7})	Move PM to reverse position
p_{c1_8}	PM is moving to normal position after fault	t_{c1_9} (t_{c2_9})	PM stuck fault acknowledge to normal position
(p_{c2_7})	Stuck fault restriction of PM	t_{c1_10} (t_{c2_10})	PM stuck fault acknowledge to reverse position
p_{c1_f1} (p_{c2_f1})	PM position indication fault while moving to reverse position	t_{c1_f1}, t_{c1_f4} (t_{c2_f1})	Indication fault occurs while PM is moving to reverse position
p_{c1_f2} (p_{c2_f2})	PM position indication fault while moving to normal position	t_{c1_f2}, t_{c1_f3} (t_{c2_f2})	Indication fault occurs while PM is moving to normal position
		t_{c1_f5}, t_{c1_f6}, t_{c1_f7}, t_{c1_f8} (t_{c2_f3})	Stuck of PM blades is detected

concept is a must for railway interlocking systems, and checking the diagnosability of the constructed models allows designers to prevent design inadequacy before testing the interlocking software [15].

6. Decision rules of the controllers and the coordinator

We consider the following four requests from the traffic control center:

r_1: Reserve route
r_2: Cancel route
r_3: Move PM to normal
r_4: Move PM to reverse.

Let R be the set of these requests. When the coordinator receives a request $r_j \in R$ from the traffic control center, it sends the request r_j to the controllers. Then, the controllers make local decisions $C_{di}(r_j) \in \{1, 0\}, (i = 1, 2, j = 1, 2, 3, 4)$, where $C_{di}(r_j) = 1$ (respectively, 0) means that the controller i decides to accept (respectively, reject) the request r_j. For example, when a PM position request (r_3 or r_4) is received from the coordinator, each controller checks whether the RB of the PM is occupied, whether the PM is moving from one position to another, and whether there is any faulty condition. The controller rejects the incoming request unless all these criteria are met. That is, the local decision is made as follows:

$$C_{di}(r_j) = \begin{cases} 1, & \text{if all safety criteria related with the request are satisfied} \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

Whether all safety criteria are satisfied is verified using the Petri net models in the controller.

The coordinator compares the local decisions of the controllers and gives its final decision to prevent the whole system from falling into a dangerous situation [22]. The final decision is sent from the coordinator to both controllers, the traffic control center, and the railway field (if necessary).

For some of the requests, the coordinator does not demand full agreement of the controllers, whereas for some of the requests the coordinator demands full agreement of the controllers. We first consider a route request (r_1) for a non-reserved route. When a route request is received, the coordinator demands full agreement of the controllers to reserve the route. That is, the final decision is made as follows:

$$C_d(r_1) = \begin{cases} 1, & \text{if } C_{d1}(r_1) = 1 \wedge C_{d2}(r_1) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

By contrast, for a route cancelation request (r_2) after the route is reserved (the route is electronically locked, the color of the entrance signal of the route is not red, and the train has not entered

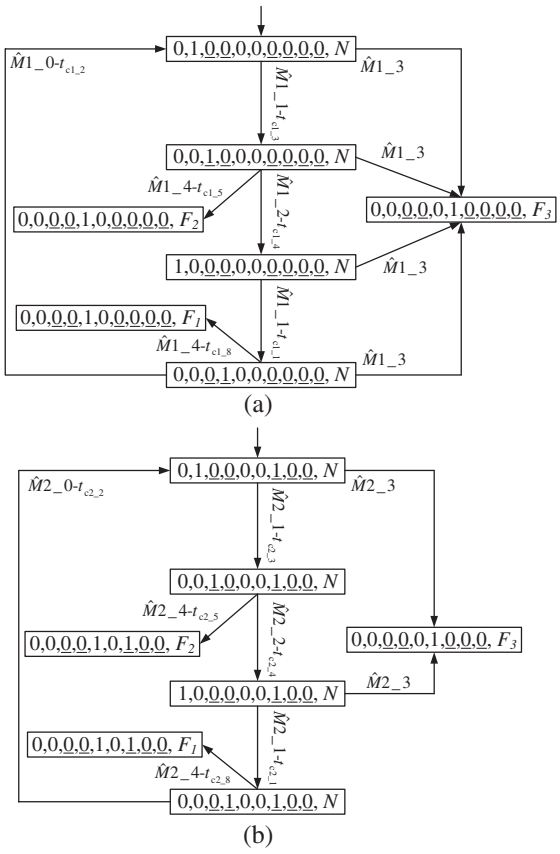

Fig. 5. Diagnosers of the Petri net models given in Fig. 4

the route yet), the coordinator does not demand full agreement of the controllers because keeping a signal red is safer than any other color indication, and the electronic lock of the route will be released after the cancellation procedure. That is, the final decision is given as

$$C_d(r_2) = \begin{cases} 1, & \text{if } C_{d1}(r_2)= 1 \vee C_{d2}(r_2)= 1, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

If the cancellation request (r_2) is received before the route is reserved (the route is not locked electronically and the color of the entrance signal of the route is still red), then the coordinator demands full agreement like in (15).

For a PM position request (r_3 or r_4), the coordinator checks additional requirements. For example, if the RB of a PM is occupied, the coordinator does not move the PM even if both controllers agree. The occupation of the RB is an additional safety check to move a PM to a desired position, and it is represented by a Boolean variable $Oc_{\mathrm{RB}} \in \{1, 0\}$ defined as

$$Oc_{\mathrm{RB}} = \begin{cases} 1, & \text{if unoccupied,} \\ 0, & \text{if occupied.} \end{cases} \tag{17}$$

The final decision for a PM position request (r_3 or r_4) is made as follows:

$$C_d(r_j) = \begin{cases} 1, & \text{if } C_{d1}(r_j)= 1 \wedge C_{d2}(r_j)= 1 \wedge Oc_{\mathrm{RB}}= 1, \\ & (j = 3, 4) \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

For instance, when the request r_3 is received from the traffic control center, the coordinator will accept this request if both controllers accept it and the related railway block is unoccupied. Using the additional check of Oc_{RB}, a kind of derailment shown in Fig. 3 can be prevented. In general, the coordinator is designed as simple as possible to decrease possible design faults. We include this additional safety check in the operations of the coordinator since it involves only the AND operation.

The controllers not only take decisions on the requests but also perform failure diagnosis. When failure information is received from a controller, the coordinator immediately informs the traffic control center without demanding full agreement and provides railway field safety (e.g., all related signals in the railway field become red and all incoming requests will be rejected).

The local diagnosis decision rule C_{fi} $(i = 1, 2)$ of the controllers is defined as a map $C_{fi} : Q_{di} \to \{N, U\} \cup 2^{\Delta_F}$, where for each state $q_{di} \in Q_{di}$ of the diagnoser,

$$C_{fi}(q_{di}) = \begin{cases} \{F_j \in \Delta_F | q_{di} \text{is } F_j\text{−certain}\}, & \text{if } q_{di} \text{is } F_j\text{-certain for some } F_j \in \Delta_F \\ N, & \text{if } l^{ci} = N \text{for any } (M^{ci}, l^{ci}) \in q_{di} \\ U, & \text{otherwise.} \end{cases} \tag{19}$$

Note that U is used to indicate that the local diagnosis is unsure whether a fault has occurred or not. Each controller outputs the local diagnosis decision $C_{fi}(q_{di})$ when its diagnoser's state is q_{di}. The decision fusion rule C_f of the coordinator is also defined as a map $C_f : (\{N, U\} \cup 2^{\Delta_F}) \times (\{N, U\} \cup 2^{\Delta_F}) \to \{N, U\} \cup 2^{\Delta_F}$, where

$$C_f(D_{f1}, D_{f2}) = \begin{cases} D_{f1} \cup D_{f2}, & \text{if } D_{f1} \in 2^{\Delta_F} \text{ and } D_{f2} \in 2^{\Delta_F} \\ D_{f1}, & \text{if } D_{f1} \in 2^{\Delta_F} \text{ and } D_{f2} \in \{N, U\} \\ D_{f2}, & \text{if } D_{f1} \in \{N, U\} \text{ and } D_{f2} \in 2^{\Delta_F} \\ N, & \text{if } D_{f1} = D_{f2} = N \\ U, & \text{otherwise} \end{cases} \tag{20}$$

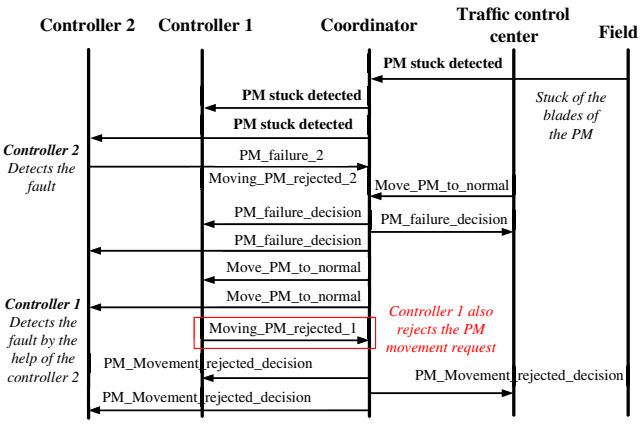

Fig. 6. Data sequence diagram for the PM stuck fault

for any local decisions denoted by D_{f1} and $D_{f2} \in \{N, U\} \cup 2^{\Delta_F}$. For the detection of a fault, the coordinator does not demand full agreement of the controllers, whereas for the decision 'normal' (N) the coordinator demands full agreement of the controllers for the safety of the system.

As an example, we consider the diagnosers shown in Fig. 5. The faults F_1 and F_2 can be detected by the both diagnosers. By contrast, in some cases, the fault F_3 can be detected by only the diagnoser shown in Fig. 5(a). For example, the diagnoser in Fig. 5(a) reaches the state $\{((0, 0, \underline{1}, \underline{0}, 0, 0, \underline{0}, \underline{0}, \underline{0}, \underline{0}), N)\}$ from its initial state $\{((0,1,\underline{0}, \underline{0}, 0, 0,\underline{0}, \underline{0}, \underline{0}, \underline{0}), N)\}$ by the firing of the observable transition t_{c1_3} with the observation $\widehat{M}1\_1$ of the resulting marking. Similarly, the diagnoser in Fig. 5(b) reaches the state $\{((0,0,\underline{1}, \underline{0}, 0,0,\underline{1}, \underline{0}, \underline{0}), N)\}$ from its initial state $\{((0,1,\underline{0}, \underline{0}, 0,0,\underline{1}, \underline{0}, \underline{0}), N)\}$ by the firing of the observable transition t_{c2_3} with the observation $\widehat{M}2\_1$ of the resulting marking. At this state, if the diagnoser in Fig. 5(a) observes $\widehat{M}1\_3$ as the resulting observable marking, the state of the diagnoser becomes $\{((0, 0, \underline{0}, \underline{0}, 0, 1, \underline{0}, \underline{0}, \underline{0}, \underline{0}), \{F_3\})\}$ whereas the state of the diagnoser in Fig. 5(b) remains at $\{((0, 0, \underline{1}, \underline{0}, 0, 0, \underline{1}, \underline{0}, \underline{0}), N)\}$. By (19), the local decisions of the diagnosers are

$$C_{f1}(\{((0, 0, \underline{0}, \underline{0}, 0, 1, \underline{0}, \underline{0}, \underline{0}, \underline{0}), \{F_3\})\}) = \{F_3\}$$

and

$$C_{f2}(\{((0, 0, \underline{1}, \underline{0}, 0, 0, \underline{1}, \underline{0}, \underline{0}), N)\}) = N.$$

Furthermore, by (20), the final decision of the coordinator is given as

$$C_f(\{F_3\}, N) = \{F_3\}.$$

Consequently, the coordinator decides that the fault of the type F_3 has occurred.

The failure diagnosis scheme looks similar to Protocol 3 of Ref. [28]. However, there are certain differences. Since the controllers are designed by different workgroups, the diagnosers are constructed on the basis of different models of the system to be diagnosed. Furthermore, they receive the same global sensor information from the coordinator. By contrast, in Ref. [28], the diagnosers are constructed on the basis of the same system model and they receive different local sensor information.

An example data sequence diagram that illustrates the diagnosis of fault by a single controller and the behavior of the whole is shown in Fig. 6. According to the data sequence diagram, when stuck-blades information is received from the PM position sensors, only the controller 2 detects the PM stuck fault and informs the coordinator of the occurrence of the failure. Then, the coordinator sends this information to the controllers and the traffic control

center. If a request is received related to the faulty PM, both controllers reject it.

The coordinator logs the useful information such as the request time, the decision of each controller, the situation of the railway field components, etc. for retrospective reporting. These reports will then be used by the inspectors in case of any accident.

An advantage of the control architecture and the handshaking process (data flow between the controllers and the coordinator) is that they provide the continuity of the system operation, prevent the system from possible design faults, and provide safety of the system. However, the reduction in system availability due to synchronization problems [22] is a main disadvantage.

7. Conclusion

Satisfying the recommended requirements of the railway-related safety standards is a must for railway interlocking system designers. Since the DES modeling methods are highly recommended by the railway-related safety standards, a fixed-block railway signaling system is studied from the DES point of view. Construction of railway field component models and their diagnosers allows engineers to represent the whole system in a formal way. Although it may seem as a complex and time-consuming task, this process enables checking the appropriateness of the models before testing the developed software. In this paper, to meet the requirements of the railway-related safety standards, a control architecture which consists of a coordinator and two controllers designed by different workgroups [22] was studied according to diverse programming. The contribution of the paper is in providing decision-making strategies including fault diagnosis.

Acknowledgement

This work was supported by JSPS RONPAKU (Ph.D. dissertation) Program and the Okawa Foundation for Information and Telecommunications.

References

(1) IEC 61508-7. *Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 7: Overview of techniques and measures*. 2010.
(2) List of Mandatory EN Standards. http://www.era.europa.eu/Core-Activities/ERTMS/Pages/Mandatory-EN-std.aspx. Accessed March, 2014.
(3) Hall S. *Modern Signalling Handbook.* Ian Allan Publishing: Birmingham, UK; 2001.
(4) Clark S. A history of railway signalling. *Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems.* 2012, 6–25.
(5) Cassandras CG, Lafortune S. *Introduction to Discrete Event Systems*, 2nd Edition. Springer: New York, USA; 2008.
(6) IEC 61508-3. *Functional safety of electrical/electronic/programmable electronic safety-related systems, Part 3: Software requirements*. 2010.
(7) EN 50128. *Railway applications-communications, signalling and processing systems, Software for railway control and protection systems*. 2001.
(8) Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control* 1995; **40**(**9**): 1555–1575.
(9) Sampath M, Sengupta R, Lafortune S, Sinnamohideen K, Teneketzis D. Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology* 1996; **4**(**2**): 105–124.
(10) Ushio T, Onishi I, Okuda K. Fault detection based on Petri net models with faulty behaviors. *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics.* 1998, 113–118.
(11) Sampath M. A hybrid approach to failure diagnosis of industrial systems. *Proceedings of the 2001 American Control Conference.* 2001, 2077–2082.
(12) Sengupta R. A discrete event approach for vehicle failure diagnostics. *Proceedings of the 2001 American Control Conference.* 2001, 2083–2086.
(13) Sinnamohideen K. Discrete-event diagnostics of heating, ventilation, and air-conditioning systems. *Proceedings of the 2001 American Control Conference.* 2001, 2072–2076.
(14) Ghazel M. Monitoring and diagnosis of discrete event systems using time Petri nets: A railway case study. *Fault Detection: Theory, Methods and Systems* 2010; 69–95.
(15) Durmuş MS, Takai S, Söylemez MT. Fault diagnosis in fixed-block railway signaling systems: A discrete event systems approach. *IEEJ Transactions on Electrical and Electronic Engineering* 2014; **9**(**5**): 523–531.
(16) Durmuş MS, Yıldırım U, Söylemez MT. The application of automation theory to railway signaling systems: The Turkish National Railway Signaling Project. *Pamukkale University Journal of Engineering Sciences* 2013; **19**(**5**): 216–223.
(17) Denault A. Introduction to Software Systems Lecture 18, http://www.cs.mcgill.ca/∼adenau/teaching/cs206/lecture18.pdf. Accessed March, 2014.
(18) Parhami B. Voting algorithms. *IEEE Transactions on Reliability* 1994; **43**(**4**): 617–629.
(19) Latif-Shabgahi G, Bennett S, Bass JM. Smoothing voter: A novel voting algorithm for handling multiple errors in fault-tolerant control systems. *Microprocessors and Microsystems* 2003; **27**(**7**): 303–313.
(20) Latif-Shabgahi GR. A novel algorithm for weighted average voting used in fault tolerant computing systems. *Microprocessors and Microsystems* 2004; **28**(**7**): 357–361.
(21) Singamsetty PK, Panchumarthy SR. A novel history based weighted voting algorithm for safety critical systems. *Journal of Advances in Information Technology* 2011; **2**(**3**): 139–145.
(22) Durmuş MS, Eriş O, Yıldırım U, Söylemez MT. A new bitwise voting strategy for safety-critical systems with binary decisions. *Turkish Journal of Electrical Engineering and Computer Sciences* http://online.journals.tubitak.gov.tr/openAcceptedDocument.htm?fileID=338940&no=72903. Accessed March, 2014.
(23) Lyons RE, Vanderkulk W. The use of triple-modular redundancy to improve computer reliability. *IBM Journal of Research and Development* 1962; **6**(**2**):200–209.
(24) Murata T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 1989; **77**(**4**):541–580.
(25) Chung S-L. Diagnosing PN-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing* 2005; **18**(**2–3**): 158–169.
(26) White C. Interlocking principles. *Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems* 2010; 65–78.
(27) Palmer JW. The need for train detection. *Proceedings of the IET Professional Development Course on Railway Signalling and Control Systems* 2010; 52–64.
(28) Debouk R, Lafortune S, Teneketzis D. Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications* 2000; **10**(**1–2**): 33–86.

Mustafa Seçkin Durmuş (Non-member) received the B.S. and M.S. degrees from Pamukkale University (PU), Turkey, in 2002 and 2005, respectively, and the Ph.D. degree from Istanbul Technical University (ITU) in 2014. From 2007 to 2014, he was a Research and Teaching Assistant with the Department of Control and Automation Engineering, ITU. He is currently a Research and Teaching Assistant with the Department of Electrical and Electronics Engineering, PU, and also a Ronpaku fellow of the Japan Society for the Promotion of Science, working with the Division of Electrical, Electronic and Information Engineering, Osaka University. His research interests include railway signaling systems and fault diagnosis of discrete event systems. Dr. Durmuş is a student member of the IEEE.

Shigemasa Takai (Non-member) received the B.E. and M.E. degrees from Kobe University, Japan, in 1989 and 1991, respectively, and the Ph.D degree from Osaka University (OU) in 1995. From 1992 to 1998, he was a Research Associate with OU. In 1998, he joined Wakayama University as a Lecturer, and became an Associate Professor in 1999. From 2004 to 2009, he was an Associate Professor with Kyoto Institute of Technology. Since 2009, he has been a Professor with the Division of Electrical, Electronic and Information Engineering, OU. His research interests include supervisory control and fault diagnosis of discrete event systems. Prof. Takai is a member of the IEICE, SICE, ISCIE, and IEEE.

Mehmet Turan Söylemez (Non-member) is a Professor with the Department of Control Engineering, Istanbul Technical University, Turkey. His research interests include pole assignment (pole placement, pole shifting, eigenstructure assignment), linear control systems theory, multivariable systems, robust control, matrix theory, computer algebra, computer-aided control system design (CACSD), numerical analysis, optimization, genetic algorithms, PID controllers, low-order controller design, simulation of DC traction railway systems, railway signaling systems, flight control systems, and emergency management systems.