A Modular Petri Net Model for the Interlocking Post Control System

Mariusz Kaniecki[1], Marcin Piątkowski[1,2]

[1]Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, ul. Chopina 12/18, 87-100 Toruń, Poland
[2]Iterdisciplinary Centre for Mathematical and Computational Modeling, University of Warsaw, ul. Tyniecka 15/17, 02-630 Warsaw, Poland

Abstract
A railway network consists of stations and interlocking posts connected by one or more tracks. Each station and interlocking post is a complex system comprising multiple tracks, switches, signalling mechanisms, and other essential components. To ensure safety, within the station or interlocking post, trains are allowed to operate only on pre-defined locked routes. Therefore, for each station or interlocking post a so-called route control table is created. Such a table defines all authorized routes and specifies the corresponding safety conditions required for train movements. In this paper, we present a Petri net model of an interlocking post, which enables the simulation of its real-life behaviour as well as train traffic. The model is created based on the route control table incorporating all essential components of the interlocking post together with the interfaces for connecting neighbouring interlocking posts. The proposed approach is an initial step toward designing a set of reliable tools for: (1) verification of safety conditions for already existing interlocking posts, (2) assisting in the design of route control tables for new interlocking posts, and (3) establishing a framework for interlocking post software and hardware control systems.

Keywords
Petri net, modelling, railway transport, interlocking post

1. Introduction

Railway transport is a cornerstone of sustainable transport with a low carbon footprint. Ensuring the safety of train operations is crucial for protecting passengers, railway staff, and infrastructure. Railway transport involves complex systems where even minor failures can lead to severe consequences, including accidents and service disruptions. Advanced signalling, interlocking systems, and real-time monitoring play a key role in preventing collisions and derailments. Additionally, strict safety regulations, regular maintenance, and automated control mechanisms help minimize risks. Since both designing and operating the railway infrastructure are prone to human errors, developing supporting tools is of utmost importance.

A railway network consists of numerous stations and interlocking posts connected by one or more tracks. An interlocking post is the smallest autonomous unit in the railway network. It is responsible for controlling turnouts and signals at railway junctions, level crossings, or intermediate points along a route to ensure the safety of train movement. A station encompasses a broader operational area, including passenger platforms, train storage facilities, and scheduling coordination. Depending on its size, a single station may consist of one or more interlocking posts, each controlling a separate area and cooperating with the neighbouring posts. Since we aim to create a formal model that ensures the safety of train operations, we focus on modelling a single interlocking post and omit elements not directly related to the train movement. However, our model is designed in a modular way and provides a way to connect adjacent interlocking posts.

Petri nets have emerged as a powerful formalism for modelling and verification of large-scale real-life systems due to their ability to represent concurrent, asynchronous, and synchronized events. As a mathematical and graphical tool, Petri nets provide a structured means to model train movements,

PNSE'25: International Workshop on Petri Nets and Software Engineering, June 23–24, 2025, Paris, France
kanies@mat.umk.pl (M. Kaniecki); marcin.piatkowski@mat.umk.pl (M. Piątkowski)
0000-0001-7882-8244 (M. Kaniecki); 0000-0001-5636-9497 (M. Piątkowski)

interlocking rules and resource constraints, allowing for precise system behaviour analysis under various conditions. By leveraging Petri net-based models, it is possible to detect and mitigate conflicts, optimize scheduling, and verify safety properties such as collision avoidance, deadlock prevention, and route exclusivity.

Our contribution We present a model of an interlocking post based on 1-bounded Petri nets. To effectively model any interlocking post and be able to adapt to its specific configuration, we propose a modular approach where each component of the model corresponds to a physical element within the interlocking post control system (such as switch, track, etc.) or a logical concept, such as the route. The physical components are created based on the interlocking post structure (however, the exact spatial placement of modeled physical elements is not relevant), whereas the logic components and the connections between all the components are based on the route control table. Moreover, utilizing the technique described in Section 3.5, it is possible to model systems consisting of several interlocking posts. Such an approach ensures the accurate representation of real-world interlocking post operations while allowing scalability and adaptability to different railway configurations. In addition, the train traffic may also be simulated. The set of example interlocking post specifications with corresponding models in PNML is available at [1].

To the best of our knowledge, there is no similar low-level model based on 1-bounded Petri nets, having direct mapping into physical elements of the railway control system.

Related work There are several papers related to the modelling and safety verification of railway systems. Most of them are devoted to the validation of route control tables, see, for instance, verification based on formal methods [2], CTL [3], FSM and NuSMV [4], UPPAAL [5], a hierarchical state machines formalism [6], combination of SAT and differential equations [7], and timed automata [8], [9], [10].

Approaches to modelling and verification based on Petri nets include [11], where a procedure is presented to identify all possible routes and validate route control tables. However, the verification is done in a static way, and no train traffic is taken into account. In [12] authors propose a construction of so-called supervisors by removing forbidden and unreachable states from the reachability graph.

More complex models, based on coloured Petri nets, have been presented in [13], [14] and [15]. The first two works focus on the topology of a typical Thai double-line station with one passing loop. The results include building a high-level model (built upon folded coloured Petri nets) for a given route control table, verification of the control table, and the simulation of the train movement. Some nonstandard Petri nets extensions have also been considered, for instance mobile Petri nets [16] and automation Petri nets [17] .

2. Preliminaries

In this section, we introduce the theory necessary to understand the details of our model of interlocking post presented in Section 3.

2.1. Petri nets

In this section, only the basic theory related to inhibitor Petri nets is recalled. We refer interested readers to [18, 19, 20, 21] for a more detailed introduction.

An *(initially marked) inhibitor Petri Net* is defined as a quintuple $S = (P, T, F, I, M_0)$, where P and T are finite and disjoint sets of places and transitions, respectively, $F : P \times T \cup T \times P \to \mathbb{N}$ is the arc weight function (called the flow), $I \subseteq P \times T$ is an inhibition relation, and $M_0 \in \mathbb{N}^P$ is a multiset of places called the initial marking (where the marking is a mapping $M : P \to \mathbb{N}$ representing the number of tokens in each place). For a transition $t \in T$, the set of *entry places* is denoted by ${}^\bullet t = \{p \in P \mid F(p,t) > 0\}$, the set of *output places* by $t^\bullet = \{p \in P \mid F(t,p) > 0\}$, the set of *inhibitor places* by ${}^\circ t = \{p \in P \mid (p,t) \in I\}$.

A transition $t \in T$ is enabled at a marking M, denoted by $M[t\rangle$, if $\forall_{p \in \bullet t}\ M(p) \geq F(p,t)$ (all its entry places are marked) and $\forall_{p \in \circ t}\ M(p) = 0$ (all its inhibitor places are empty). The execution of an enabled transition t leads to a new marking M' (denoted by $M[t\rangle M'$), where $\forall_{p \in P}\ M'(p) = M(p) - F(p,t) + F(t,p)$. The notions of enabledness and execution may be extended in a natural way, to sequences of transitions $M[w\rangle M'$ for $w \in T^*$. The set of all markings reachable from M is denoted by $[M\rangle$. A marking M is called *deadlock* if it does not enable any transition, i.e. $\forall_{t \in T}\ \exists_{p \in P}\ M(p) < F(p,t)$.

A Petri net N is called bounded if the number of tokens in each place does not exceed a certain finite limit for any reachable marking. In particular, a Petri net N is called 1-bounded if every place can contain at most a single token, i.e. $\forall_{p \in N}\ M(p) \leq 1$. For bounded nets, markings are typically represented by nonnegative integer vectors of dimension $|P|$, assuming that P is totally ordered. In the case of 1-bounded nets, markings may be represented as binary vectors.

In bounded nets, inhibitors are not strictly necessary, as an equivalent net (though more complex) can be built without them. However, to keep the model's logic simpler and avoid introducing unnecessary transitions and places, we chose to use inhibitors.

2.2. Interlocking post

An interlocking post is the smallest autonomous unit in the railway network. Its primary functions include:

- Route Control: Ensuring trains are directed onto predefined, non-conflicting paths.
- Track Occupancy Management: Preventing two trains from simultaneously occupying the same track section.
- Signal Coordination: Ensuring signals display the correct aspects based on track conditions and train movements.
- Fail-Safe Mechanisms: Implementing redundancy and verification methods to prevent signal and switch operations errors.

The physical structure of an interlocking post consists of tracks, turnouts, signalling systems, etc. An example interlocking post is depicted in Figure 1. It is a small interlocking post allowing departure/arrival in/from three different directions *A*, *B* and *K*. It contains three main tracks 1, 3 and 5, a siding track numbered 7, which is connected by turnout number 5 and equipped with derailer *DR1*, and track 9, which provides a flank protection for the station tracks 3 and 5. The tracks are connected by turnouts ①, ②, ③, ④, ⑤, ⑥, ⑦, ⑧ and ⑨, and secured by semaphores *A*, *B*, *C*, *D*, *E*, *F*, *G*, *H* and *K*.

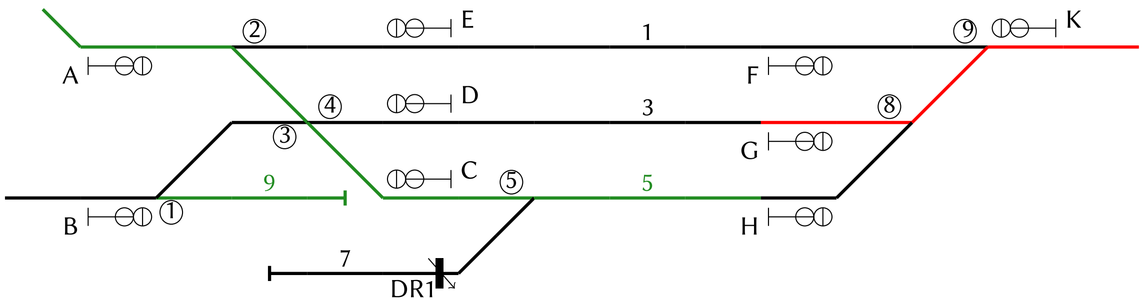

Figure 1: A small interlocking post with three possible departing directions. Two example routes are highlighted with colour: the route C^2_{dirA} in green and the route K^2_3 in red (compare with Table 1).

To ensure the safety of train movement within the interlocking post, trains are allowed to operate only on pre-defined locked routes. The list of all possible routes and the corresponding safety conditions (junction setting, signals required, etc.) form a logical layer built over the physical structure of railway components. This logical layer is represented by a route control table (see Table 1).

The modelling of individual components, including the route K^2_3 (marked in red in Figure 1), is presented in Section 3, while the entire model for the considered interlocking post is shown in the appendix.

Signals	Routes	Turnouts							Derailers
		1	2	3	4	5	8	9	DR1
A_1^1	from the direction A to the track 1		+						
A_3^2	from the direction A to the track 3	+	-	-	+				
A_5^2	from the direction A to the track 5	+	-	-	-	+			+
B_3^1	from the direction B to the track 3	-		+	+				
B_5^1	from the direction B to the track 5	-		+	-	+			+
C_A^2	from the track 5 to the direction A	+	-	-	-	+			+
C_B^2	from the track 5 to the direction B	-		+	-	+			+
D_B^1	from the track 3 to the direction B	-		+	+				
D_A^2	from the track 3 to the direction A	+	-	-	+				
E_A^1	from the track 1 to the direction A		+						
F_K^1	from the track 1 to the direction K							+	
G_K^2	from the track 3 to the direction K						+	-	
H_K^2	from the track 5 to the direction K					+	-	-	+
K_1^1	from the direction K to the track 1							+	
K_3^2	from the direction K to the track 3						+	-	
K_5^2	from the direction K to the track 5					+	-	-	+

Table 1
The route control table for the interlocking post depicted in Figure 1. The colour highlighting of route descriptions corresponds to the marking in Figure 1.

3. A model of an interlocking post control system

In this section, we outline the details of our model of an interlocking post. Its operational behaviour is based on the technical rules used in the Polish railway system (see, for instance, [22] and [23]). However, it may be adapted to follow a modified set of technical rules if need be[1]. To make all the details accessible to a general audience, we use terminology adapted from [24].

The presented model consists of two layers: the physical one, modelling the physical elements of the railway structure (turnouts, tracks, semaphores, etc.), and the logical one, consisting of all route descriptions given by the route control table.

To build our model properly, only the route control table is required. The components contained in the physical layer correspond directly to the physical elements of the interlocking post's railway structure, however, their relative positions and the details of the railway system topology are not taken into account. The logical layer, based on the route control table, contains a single component for each pre-defined locked route. This component, in addition to its internal structure, also includes the necessary connections to the physical layer components to ensure the correctness and safety of the modelled locked route. Such an approach leads to a modular model structure. Thanks to that, any kind of interlocking post can be modelled independently of its topology and structure.

Most components of the railway systems (turnouts, semaphores, etc.) may be considered as having binary states (we do not consider the time and the train speed). In the presented model those binary states are represented by the presence and absence of a token in a specific place. Moreover, the construction rules ensure that each place could at most a single token. Therefore, the considered nets are 1-bounded.

Since the considered model corresponds to a real-time system, the marking of the underlying Petri net is dynamically changing. However, we provide it with the *safe initial marking*, which should be set in the case of any system failure. Such a marking consists of a set of automatic settings (e.g. neutral state for each turnout) and the set of settings to be applied manually, possibly with the assistance of physical sensors (e.g. the current track occupancy by a train).

3.1. Tournout

A *turnout* is a mechanical device designed to divert the path of a train from one track to a different one, typically at a junction or a crossing. It has two possible states called *normal (+)* and *reverse (-)* alignment.

[1] The core physical components remain unchanged. Any differences in operational rules and safety procedures (eg. train entry authorization, route locking, train approach signaling, etc.) may require modifications of the logical components to accurately reflect these variations.

In the presented model, each turnout is represented by two places, corresponding to different switch alignments and two transitions used to change this alignment (see Figure 2). The initial conditions for these transitions allowing or preventing their execution are established by the locking of a route, which requires a fixed switch position and enforced by properly connecting them to the component representing a route (see Section 3.4). Notice that since we focus solely on the route control table and disregard the network topology, other elements of the routes may be modelled similarly. This applies also to cases where one of the states is required by specific routes, such as double slips or derailers.

Figure 2: A turnout (left) and the corresponding component of the model (right).

Safety conditions The subnet modelling the turnout contains exactly one token. Its placement in the place + or - corresponds to the turnout's switch position. The safety conditions requiring a specific turnout alignment are enforced by connecting to the components representing routes. However, those additional connections do not produce additional tokens in the considered subnet (see Section 3.4 for details).

3.2. Station track

A *station track* is a segment of a railway track situated within or adjacent to an interlocking post, typically with both ends controlled by semaphores. For each station track secured by semaphores *sem1* and *sem2*, we use two places corresponding to the signal allowing/preventing passage on the respective semaphore. Additionally, two places (*Free* and *Train*) are used to control the occupation and clearance of the considered track section. The marking of those places is updated by the route control system (see Section 3.4), therefore there is no additional transition in this component.

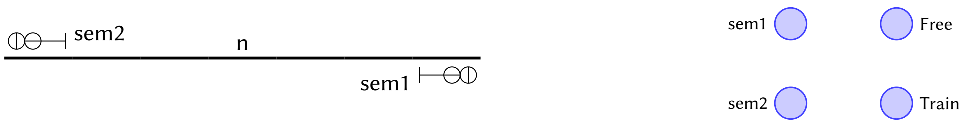

Figure 3: A station track (left) and the corresponding component of the model (right).

Safety conditions For safety reasons, there can be at most one train on each section of the track. Therefore, the places *Train* and *Free* representing a single train or no train at the track, respectively, are mutually exclusive: at any given time only one of them may contain a token.

A token in *sem1* (resp. *sem2*) indicates that there exists a module for a route (see Section 3.4) connected to this semaphore and having a token in its place *P*. This way, we ensure that the signal allowing train movement is given only for a locked route.

3.3. Direction

A *direction* is a designated route or path that an incoming (respectively outgoing) train follows when approaching (respectively leaving) an interlocking post. Usually, it consists of a track segment secured by a semaphore at the boundary of the interlocking post. The opposite open end of the track is secured by a semaphore at the neighbouring interlocking post (see Section 3.5). Operating on such a track segment requires cooperation with the neighbouring post – when a route is set in one direction, the opposite or conflicting route cannot be activated simultaneously. In general, only one of the neighbouring posts *A* and *B* can have permission for the train entry (for the other dispatching of the train should be prohibited). Additionally, *A* needs confirmation from *B* that the route is clear before the train is dispatched. Furthermore, after the train arrives at B, a confirmation is sent back to A. The safeguard mentioned above is called the *opposite locking*. An example interlocking post with three incoming/outgoing directions is depicted in Figure 1, while the component of the model representing a single direction – in Figure 4.

The state of the opposite locking on the track between two posts is described by five places: *poz* – indicates the possibility of dispatching a train onto this track, *bpoz* – indicates that dispatching a train onto this track is prohibited (the adjacent post is able to dispatch a train), *po* – indicates that a train has been dispatched from our post (we are waiting for confirmation of its arrival at the neighbouring post before unlocking the track), *ko* – indicates the departure of a train from the neighbouring post in our direction (symmetric to *po*), *sem* – represents the signal allowing/prohibiting the entrance of the train from this direction. The opposite locking is controlled by four transitions: *tr1* (respectively *tr2*) – represents granting (resp. receiving from the neighbouring post) the permission to dispatch the train, *tr3* – models the confirmation of the arrival of the train at the neighbouring post and releasing the track, *tr4* – models the train dispatch by the neighbouring post in the direction of our post.

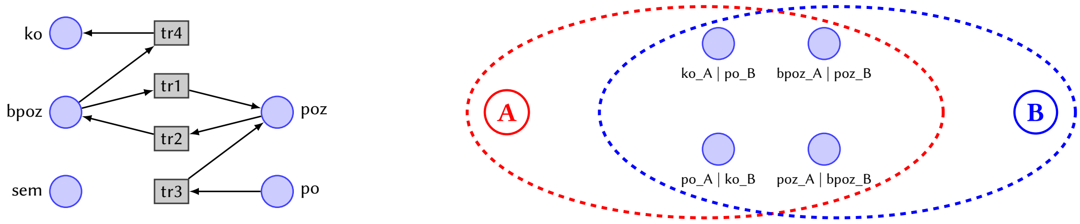

Figure 4: A component representing a single direction (left) and an overview of the connection between adjacent interlocking posts (right).

Safety conditions There may be at most one token in any of the places *ko*, *po*, *poz* and *bpoz*. The signal for a route dispatching a train in a given direction requires the use of a token from the place *poz*. This ensures that dispatching two trains in either direction is not possible, and that permission to dispatch a train to a specific track is required. The place *sem* is used analogously to semaphore places for the station track module.

3.4. Route

A *route* is the predefined path followed by a train through the interlocking post, extending from a designated entry point to a defined exit point. It must ensure the satisfaction of all required dependencies and provide the signal authorizing the train to proceed (see [22]). The allowed routes are specified at each control point in the *route control table* or in the relevant dependency records (see [23]).

The component representing a route (see module A in Figure 5) consists of a single place P indicating locking of this route and four transitions: T – blocking the route, L – releasing the blocked route if it is not locked, C – locking the route and updating the entrance signals to allow the train movement, S – modelling the train movement along the route.

The components representing routes form the central element of the proposed model. Connections to other components ensure compliance with the specifications outlined in the route control table. The required connections are as follows. For simplicity, the place P from the module M is denoted as M^P. Let T_d be a route toward the direction d from the track secured by semaphore T requiring the switches $s_1, \ldots, s_p$ to be set in the normal position, and the switches $s'_1, \ldots, s'_m$ in the reverse position. For each $1 \leq i \leq p$ (resp. $1 \leq i \leq m$), we add an edge from the transition T to the place s_i^+ (resp. $s_i'^-$) and the reverse edge from the place s_i^+ (resp. s'^-_i) to the transition T. The inhibitor arcs connecting the place P to the transition *to-* (resp. *to+*) for all switches $s_1, \ldots, s_p$ (resp. $s'_1, \ldots, s'_m$), lock those switches in the required alignment. For the transition C and S we have to consider the following cases:

1. If d is a station track we add edges connecting the place d^{Free} to the transition C (indicate that the track is no longer free)and the transition S to the place d^{Train} (allow a train to enter the track).

2. If d is a direction we add edges connecting the place d^{poz} to the transition C (revoke the permission to enter the track) and the transition S to the place d^{po} (notify neighbouring post of the incoming train).

Moreover, we have to consider the following cases for the entrance of the route:

1. If T is a semaphore at the station track m we add edges connecting places m^{Train} and $m^{sem.\ T}$ to the transition S (move the train from the track and update the semaphore state), transition S to the place m^{Free} (release the track) and transition C to the place $m^{sem.\ T}$ (update the semaphore state). Moreover, we add inhibitor arcs between the place $m^{sem.\ T}$ and the transitions C and L (C and L are enabled only for the proper semaphore state).
2. If d is a direction we add edges connecting places d^{ko} and d^{sem} to the transition S and transition S to the place d^{bpoz} (release the track), and the transition C to the place d^{sem} (update the semaphore state). Moreover, we add inhibitor arcs between the place d^{sem} and transitions C and L (C and L are enabled only for the proper semaphore state).

Example 1. *Let us consider the route G^2_{dirK} defined in Table 2.2 (module A in Figure 5). To lock this route, the normal position of turnout* 8 *(module B) and the reverse position of turnout* 9 *(module C) are required. To give a signal on the semaphore* G *(place* sem. G *in the module D representing the track* 3*), the transition* C *needs a token at the place* poz *in the module representing the direction* dirK *(module E). Observe that if the signal to proceed has been given, the release of the locking by the transition* L *is inhibited. The transition S releases track* 3 *and dispatches the train in the direction* dirK, *which is modelled by adding a token to place* po *in module E.*

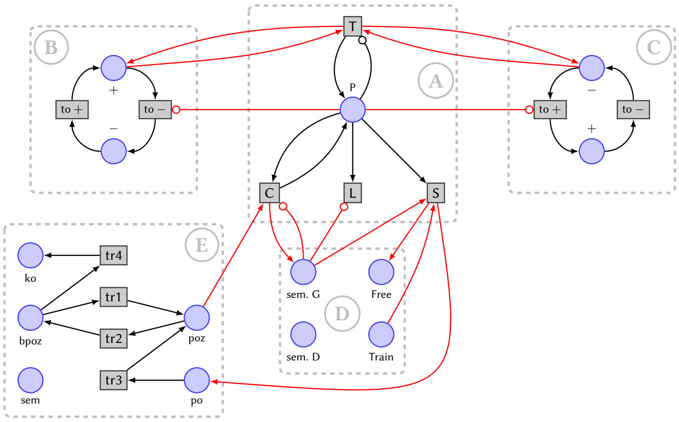

Figure 5: The connections of the module **A** representing Route G^2_{dirK} with the modules: **B** –representing turnout 8, **C** – representing turnout 9, **D** – representing track 3, **E** – representing direction *dirK*.

Safety conditions A route may be blocked (transition T places a token at place P) only if all required switches are aligned according to the route specification in the route control table. The token at P confirms the locking of the route, and inhibits changing the alignment of all switches along this route path until the execution of the transition S (modelling train movement). Locking the route and updating the signal on the securing semaphore (i.e. execution of transition C) is possible only for a blocked route with the permission to dispatch a train (place *po* in the module representing direction is marked). Similarly, releasing the route (i.e. execution of transition L) is not possible until the train left the track and the signal on the securing semaphore was updated.

3.5. Connecting the neighbouring interlocking posts

To simulate train traffic between neighbouring interlocking posts we need to specify how the models of the two individual posts are connected. Let A and B be adjacent interlocking posts. To distinguish places and transitions at individual posts, we add the name of the post in parentheses. For example, *poz(A)* denotes the place *poz* at the post A, and *tr3(B)* denotes the transition *tr3* at the post B.

The places '*ko*' and '*po*' are fully symmetrical, meaning that throughout the model's operation, the markings of the places $dirA^{po}(B)$ and $dirB^{ko}(A)$ (respectively $dirB^{po}(A)$ and $dirA^{ko}(B)$) must be identical. In this context, transition *tr4(A)* (resp. *tr4(B)*) corresponds to the transition *S* for one of the routes departing from A toward *B* (resp. from *B* toward *A*). Meanwhile, transition *tr3(A)*, which indicates the arrival of the train at the post *B*, corresponds to the transition *S* for one of the routes incoming from direction A (resp. *B*) at the post *B* (respectively *A*).

Similarly, the transition *tr1(A)* (resp. *tr1(B)*) uniquely corresponds to the transition *tr2(B)* (resp. *tr2(A)*). However, the markings of places *poz(A)* and *bpoz(B)* (resp. *poz(B)* and *bpoz(A)*) may differ during the model operation. Executing one of the transitions *C(A)* (resp. *C(B)*), which closes the route in direction *B* (resp. *A*), consumes the token from *poz(A)* (resp. *poz(B)*). The token is returned to *poz(A)* (resp. *poz(B)*) after the train reaches the interlocking post *B* (resp. *A*). As a safety condition, while the train is moving between the two posts, the departure from either of them must be forbidden, which is ensured by the absence of tokens in places *bpoz* and *poz* in the models of both posts.

3.6. Possible applications of the model

A reachablity graph for a Petri net N denoted by $RG(N)$ is a convenient way of representing its state space. The nodes of $RG(N)$ correspond to the set of all markings of N reachable from its initial marking. There is an edge $M \to M'$ in $RG(N)$ if there exists a transition t, such that $M[t\rangle M'$. A bounded net's reachability graph is finite, enabling analysis of the entire net behaviour.

One of the possible applications of the model presented in this paper may be the verification of safety conditions for already existing interlocking posts. Analysis of the structure of the reachable graph computed for the model allows the detection of potentially conflicting trajectories, dead markings (a real-time system should be able to operate without deadlocks) and other undesirable system behaviours. Moreover, to assist in dealing with exceptional situations affecting the system's functionality such as maintenance or failure of a hardware component (track, turnout, etc.) the reachable graph for a subnet of the whole model (i.e. with some transitions excluded) may be computed and examined.

A similar technique may be applied to assist the design of route control tables and establishing the safety conditions for newly projected interlocking posts.

Last but not least, the presented model may be used as a low-level framework for interlocking post software and hardware control systems. The components related to the physical structure of the railway system (turnouts, tracks, semaphores, etc.) may be implemented to directly interact with the modelled devices. On the other hand, the logical layer of the model, representing the connections and interactions between the physical components, could be implemented as an underlying software part ensuring the safety and reliability of any high-level control system built on top of it.

4. Conclusions and future work

Railway transport is a key element of sustainable mobility. Ensuring train operation safety is critical, as even minor failures can lead to severe consequences. Therefore, the reliability of railway traffic heavily depends on the effectiveness and reliability of interlocking post control systems.

In this paper, we presented a model for interlocking post control systems based on 1-bounded Petri nets, offering direct mapping to hardware components connected by a logical part following the route control table. The key feature of the model is its modular structure, enabling flexible adaptation to different railway configurations while ensuring safe and efficient train routing.

Our future research plans include using the model for scalability testing on large and complex railway networks to evaluate its performance and computational efficiency. Moreover, the model may be extended with advanced verification techniques such as model checking to provide formal guarantees of safety properties. The most desired, although technically the most difficult, future plan is the integration of our model with actual railway interlocking control systems to assess its practical applicability in real-time railway traffic control.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] M. Kaniecki, M. Piątkowski, Set of example interlocking post models, 2025. https://www.mat.umk.pl/~martinp/pnse-2025.zip.
[2] M. Banci, D. Studi, D. Firenze, S. Gnesi, Some experiences on formal specification of railway interlocking systems using statecharts (2005).
[3] G. Lukács, B. Farkas, T. Bartha, Experiences with the formal modeling of the geographical and tabular principles of interlocking systems, 2017.
[4] A. Mirabadi, M. YAZDI, Automatic generation and verification of railway interlocking control tables using fsm and nusmv, Transport Problems : an International Scientific Journal 4 (2009).
[5] P. L. Laursen, V. A. T. Trinh, A. E. Haxthausen, Formal modelling and verification of a distributed railway interlocking system using UPPAAL, in: 9th International Symposium on Leveraging Applications of Formal Methods, Springer-Verlag, 2020, p. 415–433.
[6] M. Benerecetti, R. De Guglielmo, U. Gentile, S. Marrone, N. Mazzocca, R. Nardone, A. Peron, L. Velardi, V. Vittorini, Dynamic state machines for modelling railway control systems, Science of Computer Programming 133 (2017) 116–153.
[7] T. Kolárik, S. Ratschan, Railway scheduling using boolean satisfiability modulo simulations, in: Formal Methods, Springer International Publishing, 2023, pp. 56–73.
[8] U. Khan, J. Ahmad, M. Saeed, S. Mirza, On the real time modeling of interlocking system of passenger lines of Rawalpindi Cantt train station, Complex Adaptive Systems Modeling 4 (2016).
[9] C. Avram, K. Bezerra, D. Radu, J. Machado, A. Astilean, A formal approach for railroad traffic modelling using timed automata, in: Innovation, Engineering and Entrepreneurship, Springer International Publishing, 2019, pp. 307–314.
[10] É. André, Tuning trains speed in railway scheduling, in: Formal Methods and Software Engineering, Springer Nature Singapore, 2024, pp. 37–50.
[11] B. Farkas, T. Bartha, Automated railway interlocking plan verification using Petri nets, in: Proceedings of the Fifth International Conference on Railway Technology: Research, Development and Maintenance, 2022.
[12] M. S. Durmuş, U. Yıldırım, M. T. Söylemez, Automatic generation of Petri net supervisors for railway interlocking design, in: 2012 2nd Australian Control Conference, 2012, pp. 180–185.
[13] S. Vanit-Anunchai, Modelling railway interlocking tables using coloured Petri nets, in: Coordination Models and Languages, Springer Berlin Heidelberg, 2010, pp. 137–151.
[14] S. Vanit-Anunchai, Modelling and simulating a Thai railway signalling system using coloured Petri nets, International Journal on Software Tools for Technology Transfer 20 (2018) 243–262.
[15] P. Sun, S. Collart-dutilleul, P. Bon, A model pattern of railway interlocking system by Petri nets, in: 2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), 2015, pp. 442–449.
[16] S. A. Khan, N. A. Zafar, F. Ahmad, S. Islam, Extending Petri net to reduce control strategies of railway interlocking system, Applied Mathematical Modelling 38 (2014) 413–424.
[17] M. S. Durmus, M. T. Soylemez, Railway signalization and interlocking design via automation Petri nets, in: 2009 7th Asian Control Conference, 2009, pp. 1558–1563.
[18] G. Chiola, S. Donatelli, G. Franceschinis, Priorities, inhibitor arcs and concurrency in P/T nets, in: Proc. of ICATPN, volume 91, 1991, pp. 182–205.
[19] W. Reisig, Petri nets: an introduction, volume 4, Springer Science & Business Media, 2012.
[20] T. Murata, Petri nets: Properties, analysis and applications, Proceedings of the IEEE 77 (1989) 541–580.

[21] E. Best, R. Devillers, Petri Net Primer. A Compendium on the Core Model, Analysis, and Synthesis, Birkh auser, 2024.
[22] Instrukcja le-4 (WTB-E10). Wytyczne techniczne budowy urządzeń sterowania ruchem kolejowym. (in Polish), Polskie Linie Kolejowe, Warszawa, 2014.
[23] Instrukcja o prowadzeniu ruchu pociągów Ir-1. (in Polish), Polskie Linie Kolejowe, Warszawa, 2017.
[24] J. Pachl, Railway glossary, `http://www.joernpachl.de/glossary.htm`, downloaded: 17.03.2025.