Int. J. Appl. Math. Comput. Sci., 2021, Vol. 31, No. 4, 647–657
DOI: 10.34768/amcs-2021-0045

ANALYSIS OF SAFENESS IN A PETRI NET–BASED SPECIFICATION OF THE CONTROL PART OF CYBER–PHYSICAL SYSTEMS

MARCIN WOJNAKOWSKI [a], REMIGIUSZ WIŚNIEWSKI [a,*], GRZEGORZ BAZYDŁO [a],
MATEUSZ POPŁAWSKI [a]

[a] Institute of Control and Computation Engineering
University of Zielona Góra
ul. Szafrana 2, 65-516 Zielona Góra, Poland
e-mail: r.wisniewski@issi.uz.zgora.pl

The paper proposes an algorithm for safeness verification of a Petri net-based specification of the control part of cyber-physical systems. The method involves a linear algebra technique and is based on the computation of the state machine cover of a Petri net. Contrary to the well-known methods, the presented idea does not require obtaining all sequential components, nor the computation of all reachable states in the system. The efficiency and effectiveness of the proposed method have been verified experimentally with a set of 243 test modules (Petri net-based systems). The results of experiments show high efficiency of the proposed method since a solution has been found even for such nets where popular techniques are not able to analyze the safeness of the system. Finally, the presented algorithm is explained in detail using a real-life case-study example of the control part of a cyber-physical system.

Keywords: safeness, control part of the cyber-physical system, Petri net, state machine cover, place invariant.

1. Introduction

Cyber-physical systems (CPSs) refer to a new generation of systems (Rajkumar *et al.*, 2010; Hahanov *et al.*, 2016) that determine a new era of Industry 4.0 (White *et al.*, 2020). CPSs combine cybernetic components with physical processes, whose behaviour is defined by both control (cyber) and physical parts of the system (Lee and Seshia, 2016). CPSs are all around us; they can be found in smart home automation devices (Shih *et al.*, 2016), medical systems (Dey *et al.*, 2018; Zhang *et al.*, 2017), common control systems (Barkalov *et al.*, 2018; Wiśniewski *et al.*, 2011; Guo *et al.*, 2017), power electronic converters (Wiśniewski *et al.*, 2019a), flexible manufacturing systems (Pan *et al.*, 2020; Feng *et al.*, 2020), and transportation systems (Guo *et al.*, 2018; Yang *et al.*, 2018; Huang *et al.*, 2018; Chen *et al.*, 2020).

The operation of a CPS is concurrent by nature, enabling the execution of multiple operations simultaneously. This means that adequate modelling and analysis methods ought to be applied in order to support concurrency. Furthermore, the use of advanced technologies for sensors and actuators, together with the increasing complexity of components generate a serious challenge for design and verification of modern CPSs (Szpyrka and Jasiul, 2017; Lizarraga *et al.*, 2020).

A Petri net is one of the effective tools for modelling and analysis of a wide class of concurrent systems (Jasiul *et al.*, 2015; Zaitsev, 2016; Zhu *et al.*, 2018; Jiang *et al.*, 2018). Modelling a Petri net-based system offers many advantages compared with other modelling techniques. The graphical representation of Petri nets makes the models relatively simple and legible, and well-developed analysis methods such as invariants and reachability easily detect certain anomalies of system behaviours (Ran *et al.* 2018a; 2018b; Koh and DiCesare, 1990; Murata, 1989; Clempner, 2014; Ramirez-Trevino *et al.*, 2007; Ran *et al.*, 2017).

Petri net-based systems are supported by verification, validation, and analysis methods (Li *et al.*, 2018). They permit the verification of the robustness and reliability of the modelled system (Koh and DiCesare, 1990; Jiang *et al.*, 2018) at the early specification phase, which permits a reduction in the time and costs of the designed system.

One of the most important analytic properties of Petri

[*] Corresponding author

net-based systems is safeness (Girault and Valk, 2003; Karatkevich, 2007; Aalst *et al*., 2004; Wiśniewski, 2017; Zhou and Wu, 2009). Such a property gives assurance that in every reachable marking every place contains no more than one token. A model expressed with a safe Petri net gives assurance that the designed system has a finite number of reachable states. Moreover, several design approaches require a safe Petri net as an input (e.g., Carmona *et al*., 2008; Wiśniewski *et al*., 2019b; Finkbeiner *et al*., 2020; Badouel *et al*., 1995; Cheng *et al*., 1995). It is worth noting that safeness is closely related to the other important property, boundedness (each safe Petri net is bounded by definition).

Let us discuss works related to the safeness of the Petri net-based systems. Such a property is considered by Kaid *et al*. (2020), who propose the application of Petri nets to the scheduling and deadlock analysis in reconfigurable manufacturing systems with dynamic changes. The idea is oriented especially towards architectures that are modified during their operations. Many issues related to dynamic reconfiguration (e.g., sharing the resources, synchronization, analysis of concurrency) are considered in the paper. The authors propose a method that utilizes the most important properties of Petri nets, including safeness (which is treated similarly to boundedness). The presented approach is capable of automatic and dynamic modification of the structure of the model expressed by a Petri net without affecting its behavioural properties like liveness, safeness, boundedness, and reversibility.

Carmona *et al*. (2008) an efficient synthesis presented approach for concurrent systems. Based on the theory of general regions, the authors propose an algorithm for bounded Petri net synthesis. The method requires a safe or a bounded Petri net as an input. Similar assumptions are considered by Dideban and Alla (2008). A safe Petri net is required as an input to construct a controller. Next, maximal permissive controllers are determined by computation of invariants. The paper proposes a systematic method to reduce the number of linear constraints corresponding to the forbidden states for a safe Petri net. This is realized by using non-reachable states and by building the constraints using a systematic method.

Modelling of discrete event systems by safe Petri nets is proposed by Giua and Xie (2005). In particular, the theory of supervisory control is used. The authors consider a specification that requires avoiding a set of forbidden markings. Furthermore, the paper discusses other control problems, where not only safeness of the system is analysed (e.g., avoiding a set of forbidden markings), but also where the liveness property is required, as well. Finally, an algorithm to design a maximally permissive deadlock avoidance controller for a given safe net system is presented in the paper.

Advantages of modelling with the use of safe Petri nets are presented by Cheng *et al*. (1995). The authors define a 1-safe net in which a place can contain at most one token. In consequence, the Petri net-based system has a finite state space. The paper discusses the complexity of the verification methods for three Petri net sub-classes: acyclic, conflict-free, and free-choice nets. It should be underlined that analyzed Petri nets (the input to the verification methods) ought to be 1-safe. The 1-safe Petri nets are also considered in (Best and Wimmel, 2000), where the 1-safe Petri net (or the widely k-safe Petri net) is translated into a coloured Petri net and back into an uncoloured net to show that all the nets involved in this construction have the same partially ordered multisets.

Aalst *et al*. (2004) propose a Petri net-based method for verification of the workflow management systems, especially used in prototyping and analysis of business processes. The authors state that the desirable properties of the analysed Petri net are safeness and soundness. The latter attribute strongly corresponds to the liveness and boundedness of the system. Moreover, soundness refers to the proper termination of the executed processes without unfinished operations. The presented method exploits the structure of the Petri net to find potential errors in the design of the workflow.

Wiśniewski *et al*. (2019b) propose efficient concurrency and sequentiality analysis techniques dedicated to the control part of a CPS specified by a safe Petri net. The proposed idea is based on the hypergraph theory and applies computation of subsequent exact transversals in a c-exact hypergraph in a polynomial time. The presented technique is supported by adequate algorithms, theorems, and proofs. Let us underline that the Petri net used to model the control part of the CPS and treated as an input of the proposed methods has to be safe.

Fabre (2006) considers safe Petri nets as a natural and widespread model for concurrent systems and proposes a notion of “pullback” for a Petri net. According to the author, pullbacks can be especially useful to model two concurrent systems that share some resources and are synchronized using common events. The paper provides a direct definition of simple construction for pullbacks of safe Petri nets.

The main subject of research by Cortadella *et al*. (1998) is a safe labelled Petri net. The net considered ought to be safe (a place cannot contain more than one token), while the transitions can have labels with symbols from a given alphabet. Moreover, safe nets are also well suited for verification. The authors state that every finite state system can be expressed as a safe-labelled Petri net, and as a proof they propose an original synthesis method.

Finkbeiner *et al*. (2020) introduced a model checking tool *AdamMC*, dedicated to safe Petri nets. The tool accepts the Petri net-based description as an input in the *Petri Net Markup Language* (PNML), extended with *linear-time temporal logic* (LTL) formulas connected to places and transitions of the net. To reduce the computational complexity, the proposed algorithms include sequential and parallel optimization approaches.

Esparza *et al*. (2014) proposed the coverability analysis of the system modelled by a Petri net. The method is based on utilization of the satisfiability modulo theories (SMT) solver to produce the inductive invariant. The proposed model-checking technique applies integer arithmetic. Moreover, the paper discusses similar approaches using experimental evaluation of algorithms.

The foundations of the compositional analysis of Petri nets are presented and discussed by Zaitsev (2006). The proposed approach can be useful in investigating properties of given arbitrary Petri nets, considered to be a composition of their minimal functional subnets. The idea applies fundamental equations and invariants in order to perform the compositional analysis of a net. The proposed technique is destined for the acceleration of the analysis of the Petri net properties. The author notices that well-known methods for the analysis of Petri net properties are usually related to the exponential computational complexity.

Xia and Li (2021) presented a Petri net-based representation for embedded systems (PRES+), where Petri nets are used for the modelling and verification of the designed system. The proposed approach may avoid the state explosion problem, and therefore it can be very useful in the analysis of large and complex embedded systems. Moreover, in the paper novel methods of merging transitions or transition subnets of PRES+ model are proposed. The authors state that under several additional conditions, the use of these methods preserves reachability, functionality, timing, and liveness of the analysed Petri net.

An interesting application of a Petri net is presented by Zaitsev *et al*. (2019). The paper proposes a model of a hypertorus communication grid in which a particular cell can be the representation of a bioplast cell or packet switching device. The grid is constructed in the form of an infinite Petri net. Furthermore, in order to achieve a finite specification from an infinite Petri net, a special parametric expression is obtained. Moreover, according to the authors, the infinite Petri net proposed in the paper is bounded and conservative.

An efficient and relatively simple method for computation of invariants was proposed by Martínez and Silva (1982). The presented algorithm (denoted further as the *Martinez–Silva algorithm*) generates all invariants of the generalized Petri net. The method applies linear algebra techniques in order to obtain all invariants of the analyzed Petri net. This idea can be successfully applied in the safeness verification of the system, by extracting the state machine components (SMCs) from the minimal support of p-invariants, and further computation of the SM-cover of the net. It should be noted that the "pure" Martinez–Silva algorithm permits neither the obtaining of SMCs, nor the covering of the system. All the required operations (obtaining of SMCs, computation of SM-cover) are presented in this paper (cf. Section 3).

Summarizing the above discussion, safeness is an essential and important property in design, decomposition, verification, and analysis of Petri net-based control systems, especially CPSs. Most techniques require safe Petri nets as the input data in algorithms and methods for further verification and realization.

An algorithm for the analysis of safeness in Petri net-based CPSs is proposed in the paper. The proposed technique is based on the computation of the state machine cover of the net. Contrary to the other well-known methods, the presented idea does not involve computation of all sequential components in the net. To facilitate the analysis of a large system specified by Petri nets, two reduction techniques of the Petri net are applied: a fusion of series places (FSP) and a fusion of series transitions (FST) (Murata, 1989). It is worth mentioning that these reductions preserve the system properties to be analysed (safeness, boundness, liveness). An FSP reduction technique is based on the simple elimination of the places that are modelled as a sequence (Fig. 1(a)). The only limitation is related to the places initially marked (with a token), and they are not reduced. In a strongly analogous way, the FST reduction is executed (Fig. 1(b)). Similarly, this method has the same restriction, and initially marked places are not reduced.

The main contributions can be summarized as follows:

- A method that allows for checking the safeness of the control part of a Petri net-based CPS is proposed.
- The presented idea permits the efficient and effective analysis of the system, which means that the solution is found in the assumed time (within one hour).
- The proposed method is based on the computation of the state machine cover in the Petri net-based system. The idea relies on the obtaining of the particular state machine components from the subsequent place invariants, computed and examined at each step of the algorithm,
- The algorithm has been verified and validated experimentally to confirm its efficiency and effectiveness.
- The method is illustrated by a real-life example of a CPS.
- Although the method is mainly oriented toward cyber-physical systems, it can be applied to other

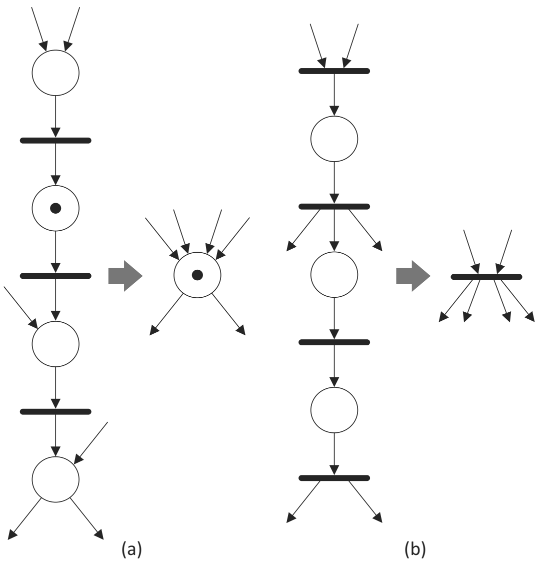

Fig. 1. Reduction of a Petri net: fusion of series places (FSP) (a), fusion of series transitions (FST) (b).

Petri net-based control systems, where safeness analysis is required.

2. Definitions and notation

This section introduces the main definitions and notations for explanation of the proposed algorithm (Best *et al.*, 2001; Murata, 1989; Karatkevich, 2007; Wiśniewski, 2017; Aalst, 2016; Wisniewski *et al.*, 2020).

Definition 1. (*Petri net*) A *Petri net* N is the quadruple

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

where P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs, M_0 is an initial marking.

Definition 2. (*Input (output) places (transitions)*) Sets of input and output places of a transition are defined respectively as follows: $\bullet t = \{p \in P \colon (p,t) \in F\}$, $t\bullet = \{p \in P \colon (t,p) \in F\}$. Analogously, the sets of input and output transitions of a place are defined, $\bullet p = \{t \in T \colon (t,p) \in F\}$, $p\bullet = \{t \in T \colon (p,t) \in F\}$.

Definition 3. (*Marking*) A marking (state) M of a Petri net $N = (P, T, F, M_0)$ is defined as a subset of its places: $M \subset P$. The set of all possible (reachable) markings is denoted by $\mathbb{M}$. A place belonging to a marking is called a *marked place*. A marked place contains a token. $M(p) = 1$ if p contains a token in M (i.e., $p \in M$), otherwise $M(p) = 0$.

Definition 4. (*Firing*) A transition t is *enabled* and can *fire* (be executed), if $\forall p \in \bullet t(M(p) > 0)$. Transition firing removes one token from each input place and adds one token to each output place. A marking can be changed only by a transition firing.

Definition 5. (*Reachability*) Marking M_j is *reachable* from marking M_i, if M_i can be changed to M_j by a sequence of transition firings.

Definition 6. (*Safeness*) A place p of a Petri net $N = (P, T, F, M_0)$ is *safe* if every reachable marking contains not more than one token. The Petri net N is *safe* if every place in every reachable marking is *safe*.

Definition 7. (*Incidence matrix*) Matrix $A_{m\times n}$ is an incidence matrix of a Petri net $N = (P, T, F, M_0)$ with $n = |P|$ columns and $m = |T|$ rows of integers, given by

$$a_{ij} = \begin{cases} -1\,, & (p_j, t_i) \in F, \\ 1\,, & (t_i, p_j) \in F, \\ 0\,, & \text{otherwise}. \end{cases} \tag{2}$$

Note that the Petri nets considered in this paper do not contain self-loops.

Definition 8. (*Place invariant*) A *place invariant (p-invariant)* is an integer vector $\vec{x} \geq 0$ of a Petri net $N = (P, T, F, M_0)$ such that

$$A\vec{x} = 0 \, , \tag{3}$$

where A is the incidence matrix of the net.

Definition 9. (*Support of p-invariant*) The set of places corresponding to the nonzero entries of a p-invariant is called its *support*.

Definition 10. (*SM-net*) A *state machine net* (*SM-net*) is a Petri net for which every transition has exactly one input place and exactly one output place: $\forall t{\in}T \colon |\bullet t| {=} |t\bullet| = 1$.

Definition 11. (*SM-component*) An *SM-component (state machine component, SMC)* of a Petri net $N = (P, T, F, M_0)$ is its subnet $\overline{N} = (\overline{P}, \overline{T}, \overline{F}, \overline{M_0})$ such that $\overline{N}$ is an SM-net.

Definition 12. (*SM-cover*) A *state machine cover (SM-cover)* of a Petri net N is a set $\{\overline{N_1}, \dots, \overline{N_k}\}$ of its SM-components, such that every place of N belongs to the set of places of at least one SM-component of this set.

Proposition 1. (Safeness of the SM-covered net (Best *et al.*, 2001)) *If a Petri net N is SM-covered by SM-components, then it is safe.*

Note that Proposition 1 provides only a sufficient (but not necessary) condition for safeness. This means that if the net is covered by SM-components, it is safe. However, if the net is not covered by SMCs, we do not know whether or not it is safe.

3. Idea of the proposed technique

This section introduces the verification technique for the safeness of the Petri-net based systems. The method is first described; then it is explained by an example.

The proposed algorithm can be divided into the following steps:

1. Initialization:
    (a) Read the Petri net $N = (P, T, F, M_0)$ that describes the control part of the cyber-physical system.
    (b) Initialize the SM-cover: $C = \emptyset$.
2. Reduction of the system:
    (a) Reduce N by fusion of series places. The resulting (reduced) net $N' = (P', T', F', M_0{}')$ contains the sets of reduced places, transitions, and arc, respectively.
    (b) Reduce N' by fusion of series transitions. The net $N'' = (P'', T'', F'', M_0{}'')$ is the result of the subsequent reductions of fusions of series places and transitions.
    (c) Form the unit matrix $Q = [D|A^{\mathrm{T}}]$, where D is the identity matrix with the size of $|P''| \times |P''|$, and A^{T} is the transposed incidence matrix of N''.
3. Searching for the SM-cover: for each column $t{\in}T''$:
    (a) Find row pairs that annul the t-th column of A^{T} (i.e., their sum is equal to 0) and append it to matrix Q.
    (b) Delete all rows of Q in which the intersection with the t-th column is not equal to 0.
    (c) Eliminate non-minimal invariants by reducing redundant rows of Q (i.e., rows that give a binary cover to the other ones).
    (d) Obtain the proper SMCs: for each row r of A^{T} whose binary contains 0 (i.e., the intersection with all columns is equal to 0):
        i. Get the support I of the place invariant corresponding to row r (such a value is directly obtained from matrix D (cf. (Martínez and Silva, 1982)).
        ii. Examine whether I forms a proper SMC (contains exactly one token in the initially marked place). If $|I \cap M_0''| = 1$ then add places of I to the current SM-cover:
        $C = C \cup I$.
    (e) Examine whether C covers all places of the net:
        - If C=P'' then **return true** (the net is covered by SMCs).
        - Otherwise, repeat the procedure from Step 3(a) (for the next transition t).
4. Safeness verification:
    - If the net is covered by SMCs, **the system is safe**,
    - Otherwise, **safeness of the system is not determined**.

The presented verification method is based on the linear algebra technique. However, contrary to the other well-known methods of analysis, it is strictly oriented toward the searching for an SM-cover of the net. Additionally, reduction techniques are applied in order to improve the effectiveness of the method. Let us now describe the presented algorithm in more detail.

Initialization is performed it the first step of the algorithm. At this stage, the Petri net-based system is read. In addition, set C is zeroed. Such a variable holds the current information about the covering of the system by SM-components and it is used further by the algorithm (cf. Step 3(e)).

In the second step, the reduction of the system is executed. In particular, two techniques are subsequently applied. Firstly, the system is reduced according to the series of places (Murata, 1989). After this operation, the reduction of the series of transitions (Murata, 1989) is subsequently applied. The above techniques do not influence the safeness of the system but may greatly decrease the computation time of the next (third) step of the algorithm (see Section 4, where results of experiments are presented and discussed). Finally, a unit matrix Q is formed. This is a conjunction of two matrices: matrix D, and the transposed incidence matrix of the reduced system A^{T}. Note that D is initially set as the identity matrix; however, after the transformations it holds the results of the method (place invariants).

Searching for the state-machine cover is the crucial point of the algorithm. This step is executed subsequently for each transition until an SM-cover is found. Furthermore, at each stage the algorithm searches for the row pairs that annul the adequate column of the transposed incidence matrix of the net. They are appended to the matrix Q and removed from the system. Similarly, non-minimal invariants (that is, those that are supersets of the other ones) are eliminated from Q. Next (Step 3(d)), the current solution is examined according to the existence of the SM-cover of the net. Those rows of A^{T} that contain all zeroes correspond to the proper place invariant. Based on their support, the algorithm verifies whether they form a proper state machine component. Such an examination can be easily done by checking whether the support of the particular p-invariant contains exactly one token in the initially marked place. If this condition is fulfilled, it is added to the set C (the current value of the SM-cover).

Finally, such a set is verified to check whether it contains all the places of the reduced system P''. The existence of an SM-cover (that is, the set C contains SMCs that cover the net) terminates the execution of the algorithm with the result that the net is safe. Otherwise, the procedure is repeated for the next transition.

Eventually, the last (fourth) step of the algorithm examines the covering of the system by state machine components. According to Proposition 1, such a covering guarantees that the system is safe. Otherwise, the safeness of the system remains undetermined.

Let us now explain the proposed method by a real-life cyber-physical system. Figure 2 shows the specification of a (cyber) control part of a multi-robot assembly system, initially presented by Zurawski and Zhou (1994). There are nine places denoted by $p_1, \ldots, p_9$, and six transitions $t_1, \ldots, t_6$ in the Petri net-based model. The robot transfers or obtains parts by operations performed with two robot arms. Activities of the first arm are related to places $p_1, \ldots, p_3$, and transitions $t_1, \ldots, t_3$. Similarly, places $p_4, \ldots, p_6$, and transitions $t_4, \ldots, t_6$ refer to the second arm. The presented system assures the collision-free movements of the robot arms in the common workspace, which is accessed by places p_3 (for the first arm), and p_6 (for the second arm). The collision-free activities are secured by the mutual exclusion zone (place p_7). Two additional places (p_8, p_9) are used as additional buffers. The interpretation of all places and transitions in the system is as follows (Zurawski and Zhou, 1994):

- p_1 (p_4): the first (second) robot arm performs actions outside the common workspace;
- p_2 (p_5): the first (second) robot arm waits for access to the common workspace;
- p_3 (p_6): the first (second) robot arm performs actions inside the common workspace;
- p_7: mutual exclusion;
- p_8, p_9: buffers;
- t_1 (t_4): the first (second) robot arm requests access to the common workspace;
- t_2 (t_5): the first (second) robot arm enters the common workspace;
- t_3 (t_6): the first (second) robot arm leaves the common workspace.

Let us now examine the safeness of the system with the proposed technique. Initially, the variable C that represents the SM-cover set as an empty set. In the second step, the net is reduced. Firstly, a fusion of series places is applied. In the presented example, places p_1 and p_2 are merged into one place, as well as places p_4 and p_5. At the subsequent step of the algorithm, a fusion of series transitions is performed. However, in this particular example, such an operation reduces neither further transitions, nor places. The final result of both reductions is shown in Fig. 3. The merged places are denoted by p_{1_2} for places p_1 and p_2, and by p_{4_5} for places p_4 and p_5. There are seven places $P''=\{p_{1\_2}, p_3, p_{4\_5}, p_6, p_7, p_8, p_9\}$, and four transitions $T''=\{t_2, t_3, t_5, t_6\}$ in the reduced net. Four places are initially marked: $M_0=\{p_{1\_2}, p_{4\_5}, p_7, p_8\}$. Based on the reduced system, the unit matrix $Q = [D|A^{\mathrm{T}}]$ is formed. Such a matrix is presented in Table 1. The left part of the unit matrix refers to the unit matrix, while the values on the right represent the transposed incidence matrix of the reduced net N''.

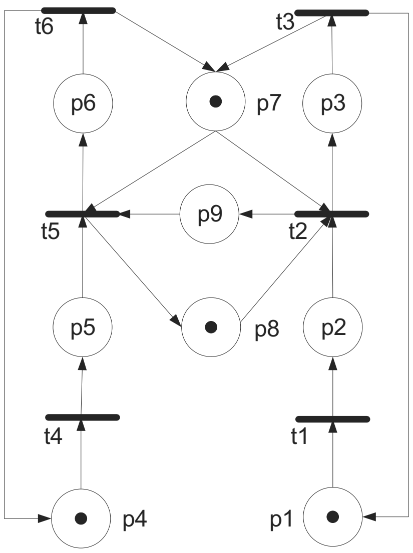

Fig. 2. Multi-robot CPS modelled by a Petri net.

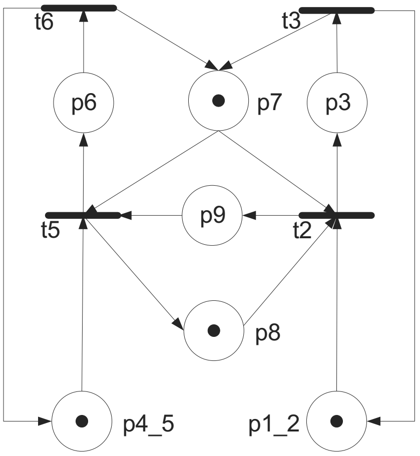

Fig. 3. Reduced Petri net.

The third step of the algorithm refers to the searching of the SM-cover of the system. This procedure is repeated for the subsequent transitions (of the reduced net N''), until SM-cover is found. In the presented example, such a cover is found after two iterations.

Table 1. Initially formed matrix $Q = [D|A^{\mathrm{T}}]$.

p_{1_2}	p_3	p_{4_5}	p_6	p_7	p_8	p_9	t_2	t_3	t_5	t_6
1	0	0	0	0	0	0	−1	0	1	0
0	1	0	0	0	0	0	1	0	−1	0
0	0	1	0	0	0	0	0	−1	0	1
0	0	0	1	0	0	0	0	1	0	−1
0	0	0	0	1	0	0	−1	−1	1	1
0	0	0	0	0	1	0	−1	1	0	0
0	0	0	0	0	0	1	1	−1	0	0

Table 2. Matrix $Q = [D|A^{\mathrm{T}}]$ after the transformations.

p_{1_2}	p_3	p_{4_5}	p_6	p_7	p_8	p_9	t_2	t_3	t_5	t_6
1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0
0	1	1	0	0	1	0	0	0	−1	1
1	0	0	1	0	0	1	0	0	1	−1
0	1	0	1	1	0	0	0	0	0	0
0	2	0	0	1	1	0	0	0	−1	1

Table 2 shows the unit matrix after the transformations (reductions). It can be noticed that transitions t_2 and t_3 are already examined, while t_5 and t_6 have not been proceeded. At this moment four rows of A^{T} are zeroed. According to the proposed algorithm, all those rows are examined according to the proper SMCs. Firstly, the support I of the place invariant that corresponds to the particular row is obtained. In the presented example there are four such supports: $I_1 = \{p_{1\_2}, p_3\}$, $I_2 = \{p_8, p_9\}$, $I_3 = \{p_{4\_5}, p_6\}$, $I_4 = \{p_3, p_6, p_7\}$.

All of the above sets form proper state machine components since they contain exactly one token in the initially marked place. Therefore, they are added to the SM-cover (held by the variable C), which contains the following places: $C = \{p_{1\_2}, p_3, p_{4\_5}, p_6, p_7, p_8, p_9\}$. This means that all places of the net are covered by SMCs. Therefore, the algorithm terminates execution with the result that the system is safe.

Let us emphasize that in the presented example, the proposed algorithm was able to find a solution after the examination of two transitions, while the traditional Martinez–Silva method requires checking all four transitions. Moreover, the initial reductions of the system by fusions of series places and fusions of series transitions are additionally permitted to reduce the initial net by two further transitions.

4. Results of experiments

The proposed solution was verified experimentally. The main aim of the performed experiments was to examine the presented method in terms of efficiency (the run-time of the algorithm) and effectiveness (the correctness of the results). The proposed technique was compared with the well-known Martinez–Silva algorithm (Martínez and Silva, 1982). (Note that the Martinez–Silva method was used for computation of place invariants in the net, while further obtaining and examination of SM-cover was performed in exactly the same way as in the proposed algorithm; cf. Section 3, Step 3(d).) Additionally, the efficiency of the applied reduction methods (the fusion of series places, the fusion of series transitions) was examined, as well. Therefore, the proposed method was tested twice: with and without the reductions.

The experiments were executed with a dedicated computational server: Intel Xeon®Platinum®8160 @2.1 GHz processor and 16 GB of RAM.

The library of benchmarks (test modules) contains 242 Petri net-based models that describe real and hypothetical systems, including cyber-physical systems, and concurrent controllers. The detailed information with a graphical model of all benchmarks can be found online at www.hippo.issi.uz.zgora.pl

Table 3 presents the results of experiments. The particular values in the table are described as follows:

- *Benchmark*: the name of a Petri net-based system;
- *Number of places*: the number of places in the net;
- *Number of transitions*: the number of transitions in the net,
- *SM-cover (Martinez–Silva)*: the result achieved by the Martinez–Silva algorithm (whether the net is SM-covered);
- *Runtime (Martinez–Silva)*: the run-time of the Martinez–Silva algorithm (in milliseconds);
- *SM-cover (the proposed method without reductions)*: the result achieved by the proposed algorithm (without applied reductions);
- *Run-time (the proposed method without reductions)*: the run-time of the proposed algorithm (without applied reductions, in milliseconds);
- *SM-cover (the proposed method with reductions)*: the result achieved by the proposed algorithm;
- *Run-time (the proposed method with reductions)*: the run-time of the proposed algorithm (in milliseconds).

Note that “*timeout*” indicates that an algorithm was not able to compute the solution in the assumed time which was set to one hour ($3.6 \cdot 10^6$ ms). In such a case the “SM-cover” was not obtained for this particular benchmark, and it is denoted by “*n/a*” in the table.

Table 3. Results of experiments.

	Number	Number of	Martinez–Silva		Proposed method		Proposed method	
Benchmark	of places	transitions			without reductions		with reductions	
			SM-cover	Runtime	SM-cover	Runtime	SM-cover	Runtime
PNwD	6	6	false	1.358	false	0.520	false	0.445
silva1	7	7	true	2.408	true	0.544	true	0.467
bit_protocol	8	7	false	2.336	false	0.555	false	0.578
esparza2	15	13	true	2.954	true	0.677	true	0.664
gaubert1	16	8	true	5.347	true	0.920	true	0.893
silva5	16	8	true	2.278	false	0.851	false	0.789
brenner1	16	12	false	2.857	false	0.646	false	0.684
eshuis1	17	15	true	3.372	true	0.611	true	0.691
hulgaard1	19	12	true	17.167	true	1.151	true	1.137
adam1	24	12	true	2.552	true	1.922	true	2.108
ConsistentExample	29	26	false	45546.008	false	1044.836	false	102.491
zuberek4	30	21	true	1.269	true	1.284	true	0.987
zuberek1	30	22	true	18.747	true	1.169	true	0.814
crossroadSM_FPGA	32	12	true	887877.986	true	56.554	true	54.329
zuberek5	41	31	n/a	timeout	true	12.770	true	1.460
PWM_extended	49	31	true	3.875	true	12.569	true	9.354
lnet_p8n1	51	40	false	1.702	false	3.895	false	1.645
cn_crr7	56	15	n/a	timeout	true	34.071	true	34.794
cn_crr10	80	21	n/a	timeout	true	180.462	true	178.471
cn_crr15	120	31	n/a	timeout	true	1404.875	true	1405.037
cn_crr25	200	51	n/a	timeout	true	21379.721	true	21504.510

Let us now analyze the obtained results. Firstly, the comparison between the proposed method (with all applied reductions) and the well-known Martinez–Silva technique is discussed (comparison of the first and third columns in the table). Then, we will analyze the influence of the reduction techniques (comparison of the second and third columns in the table).

The results obtained from the experiments show a very high efficiency for the proposed algorithm (with reductions), compared with the well-known Martinez–Silva method. Although for the relatively small examples (*PNwD*, *silva1*, etc.) both the techniques are fast, the difference is especially notable in the case of larger systems. For example, in the case of *ConsistentExample* the proposed method (with reduction) was able to compute the results over 400 times faster than the traditional one. Such a difference is even much higher in the case of the real-life system *crossroadSM_FPGA*, for which the presented algorithm time was even 16350 times faster than the Martinez–Silva method.

Finally, it should be underlined that for more complicated systems (*zuberek5*, *cn_crr7*, *cn_crr10*, *cn_crr15*, *cn_crr25*) the Martinez–Silva algorithm was not able to compute the solution within one hour. In contrast, the proposed algorithm examined all those nets (while the solution for the most complicated benchmark *cn_crr25* was computed within 21,4 seconds).

Additionally, the efficiency of the applied reduction techniques was examined. It can be observed that a reduction of series of places (transitions) is useful, but it is not always effective. For example, in the case of *ConsistentExample* the run-time of the algorithm with reductions was 10 times faster than the execution of the method without initial reductions. However, for *cn_crr25* the applied reductions even slowed the execution of the algorithm (but very slightly). Obviously, such results strictly depend on the structure of the Petri net-based systems. Nevertheless, the experiments confirm the correctness of reduction techniques.

5. Conclusions

The design of Petri net-based cyber-physical systems involves several analysis aspects. Among others (such as liveness, boundedness, reachability/concurrency verification), safeness is one of the most important, since most of the design methods and tools require such a property on their input. An algorithm oriented toward the safeness verification of a system is proposed in the paper. The method involves linear algebra, and it is based on a search for a state machine cover in the Petri net. Additionally, in order to improve the effectiveness of the method, reduction techniques are applied. In particular, a fusion of series places and a fusion of series transitions are utilized.

The experimental research confirms the high efficiency and effectiveness of the proposed algorithm. The method was able to compute results for all the tested benchmarks, contrary to the well-known Martinez–Silva algorithm, which could not find results for several test modules. Additionally, the usefulness of reduction techniques was also verified.

The proposed method can be successfully applied in the design process of Petri net-based cyber-physical systems, including concurrent control systems, discrete event systems, flexible manufacturing systems, and others. Moreover, it may be included as a conjunction with the existing modelling techniques. Several design methods require safe Petri nets as the input data for further realization (Giua and Xie, 2005; Carmona *et al.*, 2008; Kaid *et al.*, 2020; Wiśniewski *et al.*, 2018). Furthermore, the presented idea can be considered as a complement to algorithms oriented on the design of systems that already include analysis of other properties of the Petri net-based systems, for example deadlock-free manufacturing systems (Luo *et al.*, 2020; Du *et al.*, 2020; Huang *et al.*, 2021), workflow management systems (Aalst *et al.*, 2004), or embedded systems (Xia and Li, 2021).

On the other hand, there are limitations in the proposed technique. First of all, the algorithm applies computation of the state machine cover. If the Petri net is not covered by SMCs, the safeness of the system remains unsolved. In such a situation other techniques should be applied. Furthermore, the presented analysis idea is oriented toward Petri net-based cyber-physical systems. Therefore, it requires specialized knowledge (Petri nets, safeness) on the part of the designer. Finally, although the proposed method greatly improves efficiency and effectiveness in comparison to the traditional Martinez–Silva algorithm, the computational complexity is still exponential. Therefore, it is planned to enhance the proposed idea in the future.

In particular, future research will include the application of the remaining reduction techniques (fusion of parallel places, fusion of parallel transitions). Furthermore, it is planned to include transformations of the incidence matrix in order to form the reduced echelon form of the system.

Acknowledgment

This work is supported by the National Science Centre, Poland, under the grant no. 2019/35/B/ST6/01683.

References

Aalst, W.M.P., Wil, Hee, K. and Kees (2004). *Workflow Management—Models, Methods and Systems*, MIT Press, Cambridge.

Aalst, W.v.d. (2016). *Process Mining: Data Science in Action*, 2nd Edn, Springer, Berlin.

Badouel, E., Bernardinello, L. and Darondeau, P. (1995). Polynomial algorithms for the synthesis of bounded nets, *in* P.D. Mosses *et al.* (Eds), *TAPSOFT'95: Theory and Practice of Software Development*, Springer, Berlin, pp. 364–378.

Barkalov, A., Titarenko, L. and Mielcarek, K. (2018). Hardware reduction for lut–based mealy FSMs, *International Journal of Applied Mathematics and Computer Science* **28**(3): 595–607, DOI: 10.2478/amcs-2018-0046.

Best, E., Devillers, R. and Koutny, M. (2001). *Petri Net Algebra*, Springer, Berlin.

Best, E. and Wimmel, H. (2000). Reducing k-safe Petri nets to pomset-equivalent 1-safe petri nets, *in* M. Nielsen and D. Simpson (Eds), *Application and Theory of Petri Nets 2000*, Springer, Berlin, pp. 63–82.

Carmona, J., Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L. and Yakovlev, A. (2008). A symbolic algorithm for the synthesis of bounded Petri nets, *in* K.M. van Hee and R. Valk (Eds), *Applications and Theory of Petri Nets*, Springer, Berlin, pp. 92–111.

Chen, C., Liu, Z., Wan, S., Luan, J. and Pei, Q. (2020). Traffic flow prediction based on deep learning in internet of vehicles, *IEEE Transactions on Intelligent Transportation Systems* **22**(6): 3776–3789.

Cheng, A., Esparza, J. and Palsberg, J. (1995). Complexity results for 1-safe nets, *Theoretical Computer Science* **147**(1–2): 117–136.

Clempner, J. (2014). An analytical method for well-formed workflow/Petri net verification of classical soundness, *International Journal of Applied Mathematics and Computer Science* **24**(4): 931–939, DOI: 10.2478/amcs-2014-0068.

Cortadella, J., Kishinevsky, M., Lavagno, L. and Yakovlev, A. (1998). Deriving Petri nets from finite transition systems, *IEEE Transactions on Computers* **47**(8): 859–882.

Dey, N., Ashour, A.S., Shi, F., Fong, S.J. and Tavares, J.M.R.S. (2018). Medical cyber-physical systems: A survey, *Journal of Medical Systems* **42**(4) 1–13, Article no. 74.

Dideban, A. and Alla, H. (2008). Reduction of constraints for controller synthesis based on safe Petri Nets, *Automatica* **44**(7): 1697–1706.

Du, N., Hu, H. and Zhou, M. (2020). Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using Petri nets, *IEEE Transactions on Automation Science and Engineering* **17**(4): 1961–1975.

Esparza, J., Ledesma-Garza, R., Majumdar, R., Meyer, P. and Niksic, F. (2014). An SMT-based approach to coverability analysis, *in* A. Biere and R. Bloem (Eds), *Computer Aided Verification*, Springer, Cham, pp. 603–619.

Fabre, E. (2006). On the construction of pullbacks for safe Petri nets, *in* S. Donatelli and P. S. Thiagarajan (Eds), *Petri Nets and Other Models of Concurrency, ICATPN 2006*, Springer, Berlin, pp. 166–180.

Feng, Y., Xing, K., Zhou, M., Wang, X. and Liu, H. (2020). Robust deadlock prevention for automated manufacturing systems with unreliable resources by using general Petri nets, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **50**(10): 3515–3527.

Finkbeiner, B., Gieseking, M., Hecking-Harbusch, J. and Olderog, E.-R. (2020). AdamMC: A model checker for Petri nets with transits against flow-LTL, *in* S.K. Lahiri and C. Wang (Eds), *Computer Aided Verification*, Springer, Cham, pp. 64–76.

Girault, C. and Valk, R. (2003). *Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications*, Springer, Berlin.

Giua, A. and Xie, X. (2005). Control of safe ordinary Petri nets using unfolding, *Discrete Event Dynamic Systems* **15**(4): 349–373.

Guo, H., Man, K.L., Ren, Q., Huang, Q., Hahanov, V., Litvinova, E. and Chumachenko, S. (2017). FPGA implementation of VLC communication technology, *31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), Taipei, Taiwan*, pp. 586–590.

Guo, Y., Hu, X., Hu, B., Cheng, J., Zhou, M. and Kwok, R.Y.K. (2018). Mobile cyber physical systems: Current challenges and future networking applications, *IEEE Access* **6**: 12360–12368.

Hahanov, V., Hussein, M.A.A., Hahanova, A. and Man, K.L. (2016). Cyber physical computing, *2016 IEEE East-West Design Test Symposium (EWDTS), Yerevan, Armenia*, pp. 1–8.

Huang, B., Zhou, M., Wang, C., Abusorrah, A. and Al-Turki, Y. (2021). Deadlock-free supervisor design for robotic manufacturing cells with uncontrollable and unobservable events, *IEEE/CAA Journal of Automatica Sinica* **8**(3): 597–605.

Huang, D., Deng, Z., Wan, S., Mi, B. and Liu, Y. (2018). Identification and prediction of urban traffic congestion via cyber-physical link optimization, *IEEE Access* **6**: 63268–63278.

Jasiul, B., Szpyrka, M. and Śliwa, J. (2015). Formal specification of malware models in the form of colored Petri nets, *in* J.J.J.H. Park *et al.* (Eds), *Computer Science and Its Applications*, Springer, Berlin, pp. 475–482.

Jiang, Z., Li, Z., Wu, N. and Zhou, M. (2018). A Petri net approach to fault diagnosis and restoration for power transmission systems to avoid the output interruption of substations, *IEEE Systems Journal* **12**(3): 2566–2576.

Kaid, H., Al-Ahmari, A., Li, Z. and Davidrajuh, R. (2020). Automatic supervisory controller for deadlock control in reconfigurable manufacturing systems with dynamic changes, *Applied Sciences* **10**(15): 1–34, Article no. 5270.

Karatkevich, A. (2007). *Dynamic Analysis of Petri Net-Based Discrete Systems*, Springer, Berlin.

Koh, I. and DiCesare, F. (1990). Transformation methods for generalized Petri nets and their applications to flexible manufacturing systems, *Rensselaer's 2nd International Conference on Computer Integrated Manufacturing, Troy, USA*, pp. 364–371.

Lee, E.A. and Seshia, S.A. (2016). *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd Edn, MIT Press, Cambridge.

Li, B., Khlif-Bouassida, M. and Toguyéni, A. (2018). On-the-fly diagnosability analysis of bounded and unbounded labeled Petri nets using verifier nets, *International Journal of Applied Mathematics and Computer Science* **28**(2): 269–281, DOI: 10.2478/amcs-2018-0019.

Lizarraga, A., Begovich, O. and Ramírez, A. (2020). Fault diagnosis for a three-wheel omidirectional vehicle: A geometric approach, *17th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico City, USA*, pp. 1–6.

Luo, J., Liu, Z., Wang, S. and Xing, K. (2020). Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources, *IEEE/CAA Journal of Automatica Sinica* **7**(3): 812–821.

Martínez, J. and Silva, M. (1982). A simple and fast algorithm to obtain all invariants of a generalised Petri net, *in* C. Girault and W. Reisig (Eds), *Application and Theory of Petri Nets*, Springer, Berlin, pp. 301–310.

Murata, T. (1989). Petri nets: Properties, analysis and applications, *Proceedings of the IEEE* **77**(4): 541–580.

Pan, L., Yang, B., Jiang, J. and Zhou, M. (2020). A time Petri Net with relaxed mixed semantics for schedulability analysis of flexible manufacturing systems, *IEEE Access* **8**: 46480–46492.

Rajkumar, R.R., Lee, I., Sha, L. and Stankovic, J. (2010). Cyber-physical systems: The next computing revolution, *47th Design Automation Conference, DAC'10, Anaheim, USA*, p. 731.

Ramirez-Trevino, A., Ruiz-Beltran, E., Rivera-Rangel, I. and Lopez-Mellado, E. (2007). Online fault diagnosis of discrete event systems: A Petri net-based approach, *IEEE Transactions on Automation Science and Engineering* **4**(1): 31–39.

Ran, N., Hao, J., He, Z. and Seatzu, C. (2018a). Diagnosability analysis of bounded Petri nets, *IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), Turin, Italy*, Vol. 1, pp. 1145–1148.

Ran, N., Su, H., Giua, A. and Seatzu, C. (2018b). Codiagnosability analysis of bounded Petri nets, *IEEE Transactions on Automatic Control* **63**(4): 1192–1199.

Ran, N., Su, H. and Wang, S. (2017). An improved approach to test diagnosability of bounded petri nets, *IEEE/CAA Journal of Automatica Sinica* **4**(2): 297–303.

Shih, C.-S., Chou, J.-J., Reijers, N. and Kuo, T.-W. (2016). Designing CPS/IoT applications for smart buildings and cities, *IET Cyber-Physical Systems: Theory Applications* **1**(1): 3–12.

Szpyrka, M. and Jasiul, B. (2017). Evaluation of cyber security and modelling of risk propagation with Petri nets, *Symmetry* **9**(3): 32.

White, A., Karimoddini, A. and Karimadini, M. (2020). Resilient fault diagnosis under imperfect observations—A need for Industry 4.0 era, *IEEE/CAA Journal of Automatica Sinica* **7**(5): 1279–1288.

Wiśniewski, R. (2017). *Prototyping of Concurrent Control Systems Implemented in FPGA Devices*, Springer, Cham.

Wiśniewski, R., Barkalov, A., Titarenko, L. and Halang, W. (2011). Design of microprogrammed controllers to be implemented in FPGAs, *International Journal of Applied Mathematics and Computer Science* **21**(2): 401–412, DOI: 10.2478/v10006-011-0030-1.

Wiśniewski, R., Bazydło, G., Szcześniak, P. and Wojnakowski, M. (2019a). Petri net-based specification of cyber-physical systems oriented to control direct matrix converters with space vector modulation, *IEEE Access* **7**: 23407–23420.

Wiśniewski, R., Karatkevich, A., Adamski, M., Costa, A. and Gomes, L. (2018). Prototyping of concurrent control systems with application of Petri nets and comparability graphs, *IEEE Transactions on Control Systems Technology* **26**(2): 575–586.

Wisniewski, R., Grobelna, I. and Karatkevich, A. (2020). Determinism in cyber-physical systems specified by interpreted Petri nets, *Sensors* **20**(19): 1–22, Article no. 5565.

Wiśniewski, R., Wiśniewska, M. and Jarnut, M. (2019b). C-exact hypergraphs in concurrency and sequentiality analyses of cyber-physical systems specified by safe Petri nets, *IEEE Access* **7**: 13510–13522.

Xia, C. and Li, C. (2021). Property preservation of Petri synthesis net based representation for embedded systems, *IEEE/CAA Journal of Automatica Sinica* **8**(4): 905–915.

Yang, F., Wu, N., Qiao, Y., Zhou, M., Su, R. and Qu, T. (2018). Petri net-based efficient determination of optimal schedules for transport-dominant single-arm multi-cluster tools, *IEEE Access* **6**: 355–365.

Zaitsev, D.A. (2006). Compositional analysis of Petri nets, *Cybernetics and Systems Analysis* **42**(1): 126–136.

Zaitsev, D.A. (2016). Sleptsov nets run fast, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **46**(5): 682–693.

Zaitsev, D.A., Shmeleva, T.R. and Groote, J.F. (2019). Verification of hypertorus communication grids by infinite Petri nets and process algebra, *IEEE/CAA Journal of Automatica Sinica* **6**(3): 733–742.

Zhang, Y., Qiu, M., Tsai, C.-W., Hassan, M.M. and Alamri, A. (2017). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data, *IEEE Systems Journal* **11**(1): 88–95.

Zhou, M. and Wu, N. (2009). *System Modeling and Control with Resource-Oriented Petri Nets*, 1st Edn, CRC Press, Boca Raton.

Zhu, Q., Zhou, M., Qiao, Y. and Wu, N. (2018). Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **48**(3): 389–400.

Zurawski, R. and Zhou, M. (1994). Petri nets and industrial applications: A tutorial, *IEEE Transactions on Industrial Electronics* **41**(6): 567–583.

Marcin Wojnakowski received his BSc and MSc degrees in computer science from the University of Zielona Góra in 2017 and 2018. Since 2018, he has been a PhD student at the Faculty of Computer, Electrical and Control Engineering of the University of Zielona Góra. His research interests include Petri nets, modelling, analysis and decomposition of concurrent systems. He is a member of the research project *Hippo* (`www.hippo.issi.uz.zgora.pl`).

Remigiusz Wiśniewski is a professor and the head of the Division of Information Systems and Cybersecurity (Institute of Control & Computation Engineering) at the University of Zielona Góra, Poland. His research interests include design and analysis of the control part of cyber-physical systems, concurrent control systems, Petri nets, programmable devices, field programmable gate arrays (FPGAs), partial reconfiguration of FPGAs, perfect graph and hypergraph theories, and cryptography. He is an author of over 100 peer-reviewed research papers and books. He is a co-founder and the coordinator of the research project *Hippo* (`www.hippo.issi.uz.zgora.pl`).

Grzegorz Bazydło obtained his MSc and PhD degrees in computer science from the University of Zielona Góra in 2003 and 2010, respectively. Since 2012, he has been an assistant professor, now with the Institute of Control and Computation Engineering, Faculty of Computer, Electrical and Control Engineering, University of Zielona Góra (Poland). His current research interests include graphical methods (especially UML, Petri nets) in design, synthesis and verification of cyber-physical systems implemented in FPGA devices.

Mateusz Popławski obtained his BSc and MSc degrees in computer engineering from the University of Zielona Góra in 2019 and 2020, respectively. Since 2020, he has been a PhD student in computer science at the University of Zielona Góra. His current research interests include methods for the analysis of Petri nets as well as design and verification of cyber-physical systems.

Received: 27 April 2021
Revised: 15 July 2021
Re-revised: 1 September 2021
Accepted: 4 September 2021