Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P635-cd

A new approach to time Petri nets modelling with an example of transportation system performance analysis

Pawel Skrobanek

Department of Electronics and Computer Science, Koszalin University of Technology, Poland.
E-mail: pawel.skrobanek@tu.koszalin.pl

Sylwia Werbińska-Wojciechowska

Faculty of Mechanical Engineering, Wroclaw University of Science and Technology, Poland.
E-mail: sylwia.werbinska@pwr.edu.pl

In any real system, every event takes some amount of time, no matter how small. Therefore, conducting analyses related to evaluating the level of any system's performance often refers to investigations of given systems' timing parameters. In practice, the timing analysis can be typically performed with the use of, among others, Petri nets with time extensions approach. The main types of Petri nets models that allow the analysis of temporal aspects include time Petri nets (TPNs), timed Petri nets, stochastic timed Petri nets or coloured timed Petri nets. In the presented paper, the authors introduce an alternative time Petri net model with dynamic time intervals. In the proposed approach a dynamic firing time interval is assigned to tokens. In addition, the application example for the transportation system is described. Petri net with the timed token model for the presented example and its analysis are discussed. The future research directions are also defined.

Keywords: time Petri nets, dynamic time interval, timed token, a minimal set of enabling tokens, time parameters analysis, transportation system.

1. Introduction

For complex systems, where the problem of modelling the relationship between two distinct subsystems that affect the overall system availability becomes important, many studies propose the consideration of time-based reserving as well as timing analyses as a method for improving overall reliability (Werbińska-Wojciechowska 2013). The timing analysis was typically performed with the usage of such techniques as timed automata (Alur and Dill 1994; Bengtsson and Yi 2004), timed state charts (Kesten and Pnueli 1991; Eshuis 2005) and Petri nets with time extensions (Popova-Zeugmann 2013; Wang 1998). Comparison and brief survey of these approaches may be found, e.g., in (D'Aprile, et al. 2007; Bérard, et al. 2006; 2008; Bouyer, et al. 2006; Shingo 1992). A survey of time roles and methods for time analysis in Petri nets is provided in (Bowden 2000).

In the context of transportation systems, Petri nets (PNs) have recently attracted significant attention for modelling and studying their risk/reliability evaluation issues. This is mainly due to the many different types of Petri nets that have unique properties to model specific applications. One of the literature reviews on Petri net applications for transportation systems is presented in (Cavone, et al. 2018; Ng, et al. 2013). In (Cavone, et al. 2018) the authors focused on the real-life applications of Petri nets in the area of freight logistics and transportation systems. There were investigated modelling aspects of water transport, road transport, air transport, rail transport, pipeline transport, as well as intermodal and multimodal transport. Issues on urban traffic transport performance management based on Petri nets use are surveyed in work (Ng, et al. 2013). The obtained results in both studies underline the PN-based modelling and simulation potential for transportation systems performance also in the aspect of time analyses conducting.

The types of Petri nets models that allow the analysis of time aspects are reviewed, e.g. in (Diaz 2009). We can distinguish time Petri nets

(TPNs), timed Petri nets, stochastic timed Petri nets or coloured timed Petri nets. In general, the time parameters express the delay between the time when the transition is ready to fire and its firing (Berthomieu and Menasche 1982; Merlin 1974), or the (Berthomieu and Diaz 1991)duration of firing the transition (Ramchandani 1973), or are connected with places, tokens (Sifakis 1977). A detailed description of the basic classes of Petri nets extended by time aspects can be found in (Popova-Zeugmann 2013; Berthomieu and Diaz 1991).

In Petri nets they are two basic type of elements: places and transitions. Places represents for example resources, system components, channels, transitions – events, actions, executions etc. The state of a Petri net is defined by the sets of token distributed in Places. The classic solutions are based on an approach, let's call it the "system thinking" approach. We see the Petri net states as a model of a system states. The changes in Petri net states are associated with the so-called transition firing, resulting in token distribution.

The presented paper introduce an alternative time Petri net model, based on an approach, let’s call it the „object in system thinking” approach. The main differences between time Petri net (Berthomieu and Menasche 1982) and presented solution is that a dynamic firing time interval is assigned to tokens, as will be shown.

As a result, the presented paper is focused on the introduction of an alternative time Petri net model with dynamic time intervals. In the proposed approach a dynamic firing time interval is assigned to tokens. Following this, In the Introduction section, the authors present a short literature review of the investigated research area. Later, section 2 introduces the new time Petri nets with time tokens. The proposed approach is presented based on the case example of transportation system performance. The paper ends with conclusions and a definition of future research directions.

2. Time Petri Nets with Timed Token (TPNwTT)

In this paper, a method for modelling and analysis the time Petri net with time assigned to tokens is given. The main differences is dynamic time interval. In TPN the dynamic time interval is associated with transitions. In the presented solution the dynamic time interval is associated with token. Hence, in TPN for the transition to be enable it is required (as part of the condition) “*each input place contains a sufficient number of tokens*”, in TPNwTT “*each input place contains a sufficient number of tokens with the appropriate time parameters*” as will be shown.

The TPNwTT is defined as an ordered 6-tuple $<P,T,B,F,O_0, SI>$, where:

- $P=\{p_1,...,p_m\}$ is the set of places.
- $T=\{t_1,...,t_n\}$ is the set of transitions.
- B: Px$T \rightarrow N$ – backward transition function.
- F: Tx$P \rightarrow N$ – forward transition function.
- $O_0=\{\{o_1\{pi_1, <\alpha_1, \beta_1>\}, ..., o_k\{pi_k, <\alpha_k, \beta_k>\}\} \cup \varnothing$ is the initial set of tokens, where o_k – is the token identifier (it has an ordinal meaning), p_{ik} – is the place indicator for token o_k ($p_{ik} \in P$), α_k, β_k – denotes time interval in which the transition can be fired using that particular token, details will be explained later,
- SI: $TxP \rightarrow Q_+$ x $(Q_+ \cup \{\infty\})$ – function assigning the static interval of firing time to each forward transition. Let $<\alpha^S_{k,p}, \beta^S_{k,p}>$ be the static interval for a given arc from transition k to place p. **Static intervals are used to determine intervals for token or tokens created after the transition is firing.**

2.1. *Marking and tokens in TPN vs TPNwTT*

In a Petri net without time point of view, TPNwTT still has an initial marking which assigns a natural number of tokens to each place (but in different way). This marking is still graphically represented by the corresponding number of tokens (points) on the places, but the tokens are distinguishable and have assign dynamic time interval as was shown in Fig. 1.

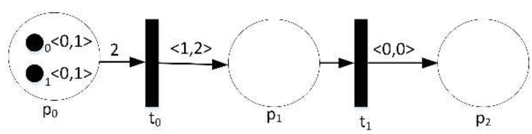


Fig. 1. The TPNwTT.

For TPNwTT given in Fig. 1:

$$\begin{cases} P = \{p_0, p_1, p_2\} \\ T = \{t_0, t_1\} \\ B = \{\{p_0, t_0\} \to 2, \{p_1, t_1\} \to 1\} \\ F = \{\{t_0, p_1\} \to 1, \{t_1, p_2\} \to 1\} \\ O_0 = \{o_0\{0, \langle 0,1 \rangle\}, o_1\{0, \langle 0,1 \rangle\}\} \\ SI = \{\{t_0, p_1\} \to \langle 1,2 \rangle, \{t_1, p_2\} \to \langle 0,0 \rangle\} \end{cases} \quad (1)$$

2.2. ***States in a TPNwTT***

According to (Berthomieu and Menasche 1982), a state $Si = \{M_k, I_l\}$ of TPN is described by the pair M_k, I_l. In presented solution, the vector I is not necessary – the time dependencies are expressed (indirectly) by set of timed tokens O_i. Therefore, in presented solution, "the transition is ready to fire" should be considered in the context of specific subsets of tokens and the time parameters of such sub-sets, which will be presented now.

Definition 1. (minimal set of enabling tokens)

The *minimal set of enabling tokens* for the transition t_i (denotes as $MSET_{i,j}$, where i – the transition number, $j \in N$ – denotes ordinal number of MSET for t_i) is the set consisting of the smallest combinations of tokens that result: each input place of t_i contains a sufficient number of tokens. $MSET_{i,j} \neq MSET_{i,k}$ for each k≠j.

Let us the $M_{MSETi,j}(p)$ denotes numbers of tokens from $MSET_{i,j}$ in place p. Therefore, for the transition t_i condition (2) must be fulfilled.

$$(\forall p) \left(M_{MSET_{i,j}}(p) = B(p, t_i) \right) \quad (2)$$

Definition 1 is similar to condition "*each input place contains a sufficient number of tokens*" (see (Berthomieu and Menasche 1982)), but due to the fact that time is assigned to tokens and tokens are distinguishable, the subsets of tokens are taken into consideration (instead the number of tokens).

Definition 2. (minimal and maximal time of $MSET_{i,j}$)

Let us the minimal and maximal time of $MSET_{i,j}$ denotes respectively as $\alpha_{MSi,j}$, $\beta_{MSi,j}$, are equal:

$$\begin{cases} \alpha_{MSij} = \max\{\alpha_l\}, \\ \beta_{MSij} = \max\{\alpha_{MSij}, \min\{\beta_l\}\} \end{cases} \quad (3)$$

where: o_l – token in $MSET_{i,j}$ with β_l, l – ranges over the $MSET_{i,j}$.

According to the Definition 2., intuitively, the time parameters α_i, β_i assign to a token o_i denotes, respectively: the minimal time that must elapse to use the token, the maximal time after which the token must be used as soon as possible. The main difference is that in the TPN, first the appropriate number of tokens is collected, and then we count the time to fire the transition (after firing, the time is reset). In the presented solution, we are waiting for a sufficient number of tokens which fulfil the time criteria.

Definition 3. (enabled)

The transition t_i is *enabled* if and only if:

- is enabled for marking - have at least one $MSET_{i,j}$,
- is enabled for time - all other enabled for marking transitions can be firing later (the condition (14) must be fulfilled).

$$\alpha_{MSij} \leq \theta \leq \beta_{MSk} \quad (4)$$

where: θ - relative (to "now") time of firing the transitions t_i, β_{MSk} – other minimal set of enabling tokens, k – ranges over the set of *MSET*.

Definition 4. (firing)

Let $N = \{P, T, B, F, O_0, SI\}$ be a Petri net. Let O_m be a marking in N. A transition $t_f \in T$ can fire in O_m (notation: $O_m \xrightarrow{t_f / MSET_{f,j} /}$), if t_f is enabled in O_m. After the firing of t_f the Petri net is in the new marking O_n (notation: $O_m \xrightarrow{t_f / MSET_{f,j} /} O_n$).

Firing of the transition t_f consists of three steps (analogically to steps in TPN):

(i) removal from set O of $B(p,t_f)$ tokens from each input places **according to $MSET_{f,j}$**,
(ii) calculating a new time parameters to tokens,
(iii) addition to set O of $F(t_f,p)$ tokens (**whit time interval equal to static time interval $SI(t_f,p)$**) to each output place.

These operations are given by three steps as follows:

STEP 1. Eliminations of tokens in $MSET_{f,j}$ from set O_m.

$$SuperSuper\,O_m \backslash MSET_{fj} \quad (5)$$

STEP 2. Shift times of all remaining firing intervals by the value θ.

$$\alpha_k = \max\{0, \alpha_k - \theta\}, \beta_k = \max\{0, \beta_k - \theta\} \quad (6)$$

STEP 3. Introducing to the set O_m new tokens o_n, where n is the next (not used) ordinal number for token. For each p to which the arc leads from t_f and $F(t_f,p)$ times introducing new tokens o_n will be performed as follows:

1. $O_m \cup \{o_n\{p, <\alpha^S_{tf,p}, \beta^S_{tf,p}>\}$

2. calculating new *MSETs* with o_n (if possible).

3. n=n+1

Analogically to the TPN, the number of states is infinite and the concept of state classes in TPNwTT will be introduced.

2.3. *State classes and reachability graph in TPNwTT*

The difference between condition and class is qualitative. The dependencies described by the domain in TPN (Berthomieu and Menasche 1982; Berthomieu and Diaz 1991), result in TPNwTT (indirectly) from the time parameters assigned to tokens. Hence, the notation of the state S_0 and the class C_0 does not differ, but the next classes (as opposed to states) are determined based on the calculation of the time intervals (not based on the value of θ), what will be shown.

Definition 5. (state classes)
State classes C_i is a single: $C_i = (M_i)$, where: M_i – is the marking (all state in the class have the same marking).

In comparison with definition given in (Berthomieu and Diaz 1991), the domain D_i is omitted, due to fact, that dependencies follow from the parameters assigned to tokens.

Let us denote $\tau(t_i)$ to be the time instant of firing of the transition t_i.

Initial class $C_0 = (M_0)$ obtain as follows: the initial marking is the initial marking of the Petri net. Let t_f be the firing transition in class C_i. The class C_j is computing from class C_i (notation $C_i \xrightarrow{t_f} C_j$) by performing the following operations:

STEP 1. Let t_f be the firing transition with $MSET_{f,j}$ by marking O_m. The operations *(O1)* for all tokens outside $MSET_{f,j}$ are performed.

$$^{(O1)}\begin{cases}\beta_f = \min\{\beta_{MSk}\} \\ \alpha_i = \max\{0, \alpha_i - \beta_f\} \\ \beta_i = \max\{0, \beta_i - \alpha_{MSfj}\}\end{cases} \quad (7)$$

where: k – ranges over the set of *MSET* (including $MSET_{f,j}$)

STEP 2. Eliminations of tokens in $MSET_{f,j}$ from set O_m and $MSET_{f,j}$ elimination.

$$^{(O2)}O_m \backslash MSET_{fj} \quad (8)$$

STEP 3. Introducing to the set O_m new tokens o_n.

for each p to which the arc leads from tf and $F(t_f,p)$ times introducing new tokens will be performed as follows:

$$^{(O3.1)}O_m \cup \{o_n\{p, < \alpha^S_{tf,p}, \beta^S_{tf,p} >\} \quad (9)$$

where n is the next (incremented) number assign to the introducing into the O_m set token.

$^{(O3.1)}$ calculating new *MSETs* with o_n (if possible).

Using the firing rule a reachability graph can be built, as was shown in Fig. 2.

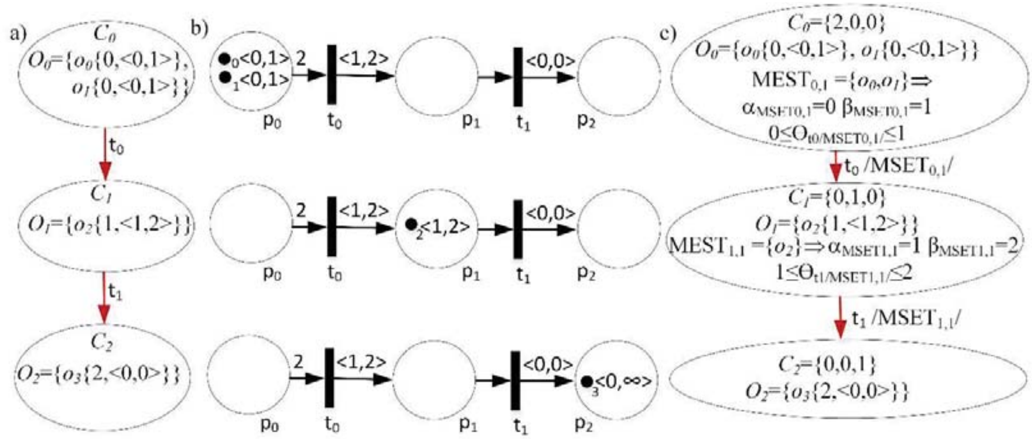


Fig. 2. a) The state classes diagram (standard notation), b) Corresponding TPNwTT models c) The state classes diagram with additional information

4. Case study

Let us assume that our company has two subdivisions and a central storage facility, and that when the work at the subdivisions is completed (the conventional '0' moment), the products are forwarded to the warehouse (Fig. 3).

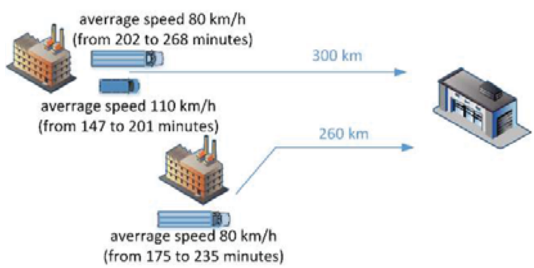


Fig. 3. A simplified model for forwarding goods to the warehouse

In the case of the first branch, we have two vehicles; in the case of the second, we have one (as shown in Fig. 3). In order to simplify the analysis, let us assume that the transport of goods takes place mainly on the highway and that it takes place in the time interval given in the Fig. 3. The values were determined by taking the value of the time calculated for the given distance and the average speed of the vehicle reduced (for the lower limit) or increased (for the upper limit) by 10% of this value. In addition, twenty minutes were added to the upper limit due to the possible delay of the vehicle departure from the conventional "0" moment.

Let us assume that the warehouse in the afternoon and at night is operated by one vehicle unloading station and that the vehicles are unloaded in the order of their arrival. Let for each vehicle it takes no less than t_{unl_min} and no more than t_{unl_max} to unload, and given that the unloading procedure involves properly parking the vehicle, unloading, completing the documentation and leaving with the vehicle, that the minimum unloading time satisfies the inequality $t_{unl\_min} \geq 10$. A model of a temporal Petri net for the described transport system is shown in Fig. 4. In contrast to classical temporal Petri nets, by assigning time to markers, we can simplify the model to a single location modelling "vehicles on the road". Building the model using classical TPNs would require modelling with three transitions - due to the different travel times of the vehicles.

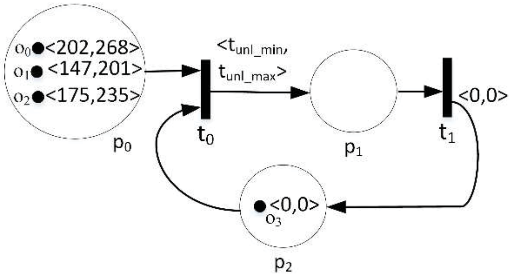
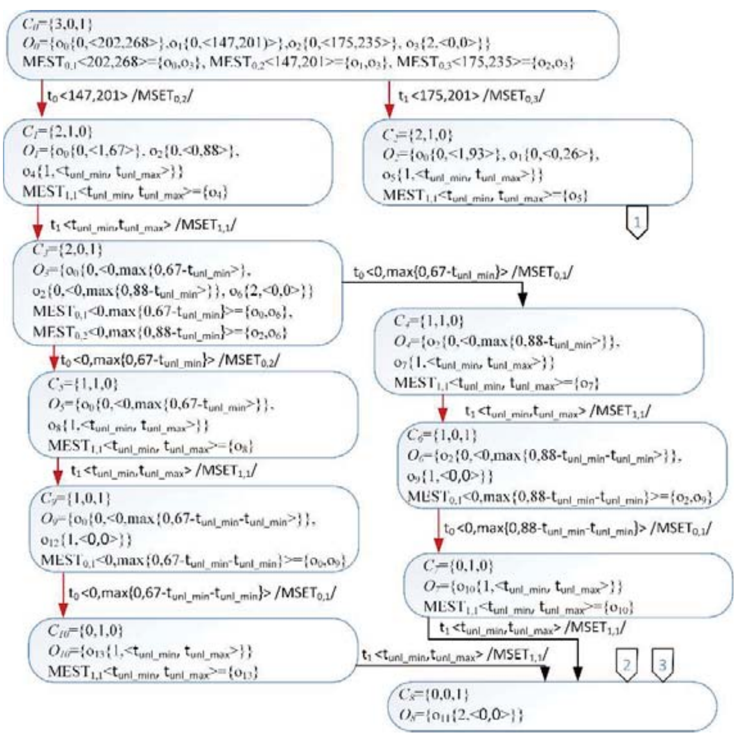


Fig. 4. Timed Petri Net with time assigned to tokens, that represents the transportation system from Fig. 3.

The analysis of the timed Petri net allows for the examination of all possibilities related to, among other things, the different order of arrival of delivery vehicles. The results of the analysis covering these possibilities are shown in Figure 5. Analysing in the class diagram the paths leading to class C_8 modelling the situation when all vehicles have been unloaded, we obtain relationships modelling the shortest and longest unloading times of all vehicles, denoted $t_{unl\_all\_min}$, $t_{unl\_all\_max}$ respectively. The shortest unloading time for all vehicles is given by equation (10) and corresponds to the unloading starting from the faster vehicle from branch one, while the longest unloading starts from the vehicle from branch two and is given by equation (11), whereby, the last two components of the sum describe the waiting time or lack thereof when the minimum unloading time for a single vehicle t_{unl_min} is large enough. Such knowledge can be used, for example, to assess whether vehicles will wait to be unloaded and, if so, how much and whether, for example, it is worth running an additional stand to unload one or more vehicles.

$$t_{unl\_all\_min} = 147 + 3 \cdot t_{unl\_min} \quad (10)$$

$$t_{unl\_all\_max} = 201 + 3 \cdot t_{unl\_max} + \max\{0{,}67 - t_{unl\_min}\} + \max\{0{,}88 - 2 \cdot t_{unl\_min}\} \quad (11)$$



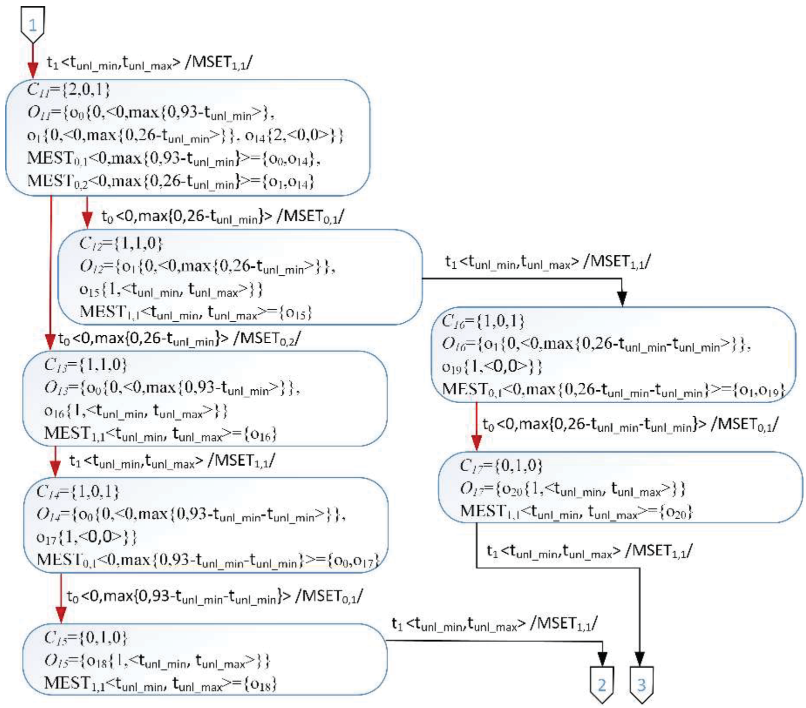


Fig. 5. The results for timed Petri net

5. Conclusions

Petri nets with time assigned to tokens allow easier modeling and analysis of object-oriented systems than classical time Petri nets. The present work aims to demonstrate the new model of timed Petri nets along with outlining the area of its applicability. In the later perspective, it is planned to develop an automatic conversion of such a network model into software that monitors the location of objects by, for example, analyzing their location data in the real world and comparing it with criteria derived from the time Petri net model. This should make it possible, among other things, to detect situations involving failures or delays, and ultimately, to take timely action to prevent undesirable situations from occurring.

References

Alur, R., and Dill, D. L. (1994). A Theory of Timed Automata. *Theoretical Computer Science* 126 (2): 183–235. https://doi.org/10.1016/0304-3975(94)90010-8.

Bengtsson, J., and Yi, W. (2004). Timed Automata: Semantics, Algorithms and Tools. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3098: 87–124. https://doi.org/10.1007/978-3-540-27755-2_3.

Bérard, B., Cassez, F., Haddad, S., Lime, D., and Roux, O. H. (2006). Comparison of Expressiveness for Timed Automata and Time Petri Nets. *ANNALES Du LAMSADEn4-5*, no. hal-00115535: 55–106. https://doi.org/10.1002/9780470611098.ch4.

Bérard, B., Cassez, F., Haddad, S., Lime, D., and Roux, O. H. (2008). When Are Timed Automata Weakly Timed Bisimilar to Time Petri Nets? *Theoretical Computer Science* 403 (2–3): 202–20. https://doi.org/10.1016/j.tcs.2008.03.030.

Berthomieu, B., and Diaz, M. (1991). Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Transactions on Software Engineering* 17 (3): 259–73. https://doi.org/10.1109/32.75415.

Berthomieu, B., and Menasche, M. (1982). A State Enumeration Approach for Analyzing Time Petri Nets. *Proceedings of the 3rd European Workshop on Applications and Theory of Petri Nets, Varenna, Italy.*, no. 1982: 27–65.

Bouyer, P., Haddad, S., and Reynier, P. A. (2006). Extended Timed Automata and Time Petri Nets.

Proceedings - International Conference on Application of Concurrency to System Design, ACSD, 91–100. https://doi.org/10.1109/ACSD.2006.6.

Bowden, F. D. J. (2000). A Brief Survey and Synthesis of the Roles of Time in Petri Nets. *Mathematical and Computer Modelling* 31 (10–12): 55–68. https://doi.org/10.1016/S0895-7177(00)00072-8.

Cavone, G., Dotoli, M., and Seatzu, C. (2018). A Survey on Petri Net Models for Freight Logistics and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems* 19 (6): 1795–1813. https://doi.org/10.1109/TITS.2017.2737788.

D'Aprile, D., Donatelli, S., Sangnier, A., and Sproston, J. (2007). From Time Petri Nets to Timed Automata: An Untimed Approach. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4424 LNCS: 216–30. https://doi.org/10.1007/978-3-540-71209-1_18.

Diaz, M. (2009). *Petri Nets. Fundamental Models, Verification and Applications*. Wiley.

Eshuis, R. (2005). Statecharting Petri Nets. *Research School for Operations Management and Logistics* 153: 1–37. http://www.researchgate.net/publication/228573000_Statecharting_petri_nets/file/60b7d516537e993df7.pdf%5Cnpapers3://publication/uuid/EE99A7BC-C138-484E-8FF3-6758496D2ADB.

Kesten, Y., and Pnueli, A. (1991). Timed and Hybrid Statecharts and Their Textual Representation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 571 LNCS: 591–620. https://doi.org/10.1007/3-540-55092-5_32.

Merlin, P. M. (1974). A Study of the Recoverability of Computing Systems. University of California, Irvine.

Ng, K. M., Reaz, M. B. I., and Ali, M. A. M. (2013). A Review on the Applications of Petri Nets in Modeling, Analysis, and Control of Urban Traffic. *IEEE Transactions on Intelligent Transportation Systems* 14 (2): 858–70. https://doi.org/10.1109/TITS.2013.2246153.

Popova-Zeugmann, L. (2013). *Time and Petri Nets*. *Time and Petri Nets*. Vol. 9783642411. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-41115-1.

Ramchandani, C. (1973). Analysis of Asynchronous Concurrent Systems by Timed Petri Nets. Massachusetts Institute of Technology.

Shingo, T. (1992). Equivalence between a Finite Automaton and a Petri Net. *Complex Systems with Decision Makers* 809 (1): 130–42.

Sifakis, J. (1977). Use of Petri Nets for Performance Evaluation. In *Measuring, Modelling and Evaluating Computer Systems, Proceedings of the Third International Symposium, Bonn - Bad Godesberg, Germany*, 185–201. https://doi.org/10.1016/b978-0-444-81660-3.50018-6.

Wang, J. (1998). *Timed Petri Nets. Theory and Application*. Springer Science + Business Media LCC.

Werbińska-Wojciechowska, S. (2013). Time Resource Problem in Logistics Systems Dependability Modelling. *Eksploatacja i Niezawodnosc* 15 (4): 427–33.