Scientific Research Publishing

Journal of Software Engineering and Applications, 2023, 16, 443-482
https://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

FMS-Workflow Modeling Based on P-Timed Stochastic Petri Net

Walid Ben Mesmia[1], Kamel Barkaoui[2], Mohamed Escheikh[1]

[1]Sys'Com-ENIT, Tunisia, ENIT, Le Belvédère, Tunis, Tunisia
[2]CEDRIC-CNAM, Paris, France
Email: benmesmiawalid77@yahoo.fr, kamel.barkaoui@cnam.fr, mohamed.escheikh@enit.utm.tn

How to cite this paper: Mesmia, W.B., Barkaoui, K. and Escheikh, M. (2023) FMS-Workflow Modeling Based on P-Timed Stochastic Petri Net. *Journal of Software Engineering and Applications*, **16**, 443-482.
https://doi.org/10.4236/jsea.2023.169022

Received: July 11, 2023
Accepted: September 11, 2023
Published: September 14, 2023

Abstract

In this paper, we propose astochastic Petri net model P-timed Workflow (*WPTSPN*) to specify, verify, and analyze a business process (*BP*) of a Flexible Manufacturing System (*FMS*). After formalizing the semantics of our model, we illustrate how to verifysome of its properties (reachability, safety, boundedness, liveness, correctness, alive tokens, and security) in the P-Timed context. Next, we validate the relevance of the proposed model with *MATLAB* simulation through a specific *FMS* case study. Finally, we use a generalized truncated density function to predict the duration of a token's sojourn (residence) in a timed place with respect to the sequence states of the global *FMS* workflow.

Keywords

WPTSPN, SPN, Workflow, FMS, P-Timed, Specification, Verification, Prediction

1. Introduction

On the one hand, in thispaper, we suggest a Workflow P-Timed Stochastic Petri Net model (*WPTSPN*) for a Flexible Manufacturing System (*FMS*) management process to describe and decide how business activities are carried out. Next, we show the generic (P-timed) characteristicsrelating to reachability, safety, boredness, liveness, properness, and alive-tokens, followed by a description of our model semantics. On the other hand, we propose an extension of stochastic Petri nets (SPNs). Weverify the properties of this model using a new notation and expressiveness-specific features related to the randomresidences of tokens in timed places. This allows us to detectex ecution failures linked either to FMS steps or to interactions between resources used in the business process. First, to introduce the application field of this work, we present the complexities and complicated dynamism of resource allocation and execution in the industrial process. FMS [1] modeling has been introduced for the analysis, verification, and optimization of a manufacturing system [2] [3] according to various criteria. The objective is to evaluate and compare the architectures of production systems to select the most suitable according to previously established decision criteria [2]. This process involves the use of formal models ensuring specification, analysis, evaluation, and verification [4] [5].

So, production companies must develop a strategy that ensures control over their operations if they are to profit from the business process paradigm [1]. To achieve this, they must tend to totally or partially automate their processes because this method is apt to show the efficiency and tendency towards this objective [1] [5]. Several research works propose methodologies for modeling and automating business processes [4] [6] [7] [8].

Second, the need to model an FMS to achieve automation, control, analysis, and optimization of the elementary activities that make up its representative business process [2]. A variety of business process modeling methods [9] (specification and representation) are used to practically specify work procedures from a global perspective to describe their operational requirements. We can mention among these methods the activity diagram UML (Unified Modeling Language), Chain Event-Driven Process [10] (*EPC*) and Business Process Modeling and Scoring (*BPMN*) [11]. We consider *BPMN* [11] [12] [13] [14] as a standard for process modeling. It provides a graphical notation-based representation for business process modeling [15]. To build a standard annotation and expressiveness to make a business process representation understandable and readable for the different users involved in procedures' automation [16] [17].

Despite the reality that specification and verification methods are widely used, their application to the business domain remains difficult [8]. This is related to the reasons that business analysts strive to formalize simple approaches to modeling and simulating their processes over a long period of time [18]. These approaches' disadvantage, according to [8] [18] [19], comes down to the need for operational expertise (manipulation and simulation). Furthermore, the representative model may deviate from the actual simulated due to the significant increase in the number of tests (time and simulation cost ratio) [8] [18].

Third, the use of Petri Nets (*PNs*) to model an FMS management process [8] [18] [20] is justified as follows:

On one hand, industrial companies need a tool that simultaneously merges modeling and simulation processes. Also, this company's type has a fundamental objective of economic competitiveness, which summarizes the satisfaction of customers' tendencies with the changes in the environment [21]. Thus, this flexible and competitive aspect requires representation and control via the supply chain simulator and the use of modeling tools that guarantee an acceptable degree of comprehensibility and readability [21]. On second hand, it confirmed that (*PNs*) can represent a manufacturing system composed of simultaneous and synchronous activities set according to different levels of abstraction [22] [23] [24]. Any formalization based on the *PNs* [25] [26] [27] applied in the *FMS*, models the operational knowledge in the workshop, favoring a planning of the processes. So it provides a decision support tool [28] [29] [30] [31]. Additionally, *PNs* [22] [23] [24] [29] can model manufacturing systems at different levels of abstraction in a hierarchical manner identifying many strategic keys or control points [23]. Thus, *PNs* based models are used effectively to analyze control problems in real-time [23] [24]. These analyses are useful for understanding the behavioral aspect of real-time operations in an *FMS*. They effectively ensure a performance and systematic analysis by varying the parameters of the system represented.

We can list the works related to modeling *FMS* (Flexible Manufacturing System) with *PN* (Petri Nets) as follows:

- In [32], the author uses the *PNs* in modeling and controlling the *FMS*. The control function is divided into two levels: their proposed model based on *PN* [32] represents the lower level of control for each device in a cell and the upper level which deals with a decision framework about priority sequencing to avoid collisions.
- In [33] [34], the authors propose a model based on timed Petri nets (*TPN*) to model and analyze a production process. Also, they show an approach for translating TPN into linear algebraic equations. Evaluation and monitoring of production system performance can be done through the use of efficient algorithms to solve these last equations [33] [34].
- In [35] [36], a software package called Generalized Stochastic Petri-Net Analysis (*GSPNA*), is used to study the complex and dynamic behavior of a system. So, *GSPN* can achieve a more accurate and refined representation of a *FMS* than classical techniques such as network-based queuing [24] [35]-[39].

On the other hand, the interest in orienting our proposed model towards the *P-timed* semantics of PNs [40] [41] [42] is summarized in the fact that they effectively ensure the analysis as well as the verification of the dynamic behaviors of the systems they model. The latter (*P-timed PN*) combines two different approaches to analyze the behavior and study the properties of a model: the time interval approach and the tokens-age approach. That is to say that the first allows a better study of the behavior while it is preferable to use the second for the analysis of the properties of the model. Moreover, our proposed extension of P-timed SPN orientation [43] [44] [45] is based on the assumption that all model transitions must be fired. Therefore, the arrival times of the tokens in the places that are synchronization transition entry places must be compatible. However, another advantage of this extension is that it randomly models (specifies) the minimum residence times and the maximum residence times of the tokens in all the places of the model (even those that are upstream of the synchronization transitions) [41]. This advantage is not possible with other classic PN models [35] [36] [46] [47].

When we analyze the work related to modeling with Stochastic Petri Nets, we find:

- The authors [48] [49] [50] propose a model named *non-Markovian SPN*. They show that *Non-Markovian SPN* allows specifying the behavior of a system by representing the system more realistically. According to their modeling approach, each transition is characterized by a distributed random duration. This representation adopts bounded support laws with a predetermined probability density. Also, they use the maximum density method to estimate the firing time of the transitions [49] [50].
- The authors [7] [19] propose a second level (refined representation) described by the intermediate locations (micro locations), the micro-transitions, and their firing preconditions according to a micro-level life cycle (token consumption phase, execution phase, and token production phase).

So our model based on P-timed SPN captures the random time requirements in the structure of the representative model of an FMS. In this work, we propose an extension of SPN with a P-timed orientation in specification and verification. The model referred to as Workflows P-Timed Stochastic Petri Nets (*WPTSPN*) enables life cycle specification and control of tokens circulating in the timed places according to sequences of transitions. This is fulfilled through granting to each places a dynamic residence interval. This residence time allocated to a place follows a law of probability determined via three possible densities (exponential, normal, and log-normal). To determine the average residence time granted to each token, we chose the gravity center of the density method, for precision accuracy. Hence, this model adopts the formal verification tools to conform to the P-timed orientation, knowing that such an orientation brings verification of the token's life cycle in timed places. From where we redefine a formalism of verification according to this semantics Thus, the model proposes a prediction approach of the delay related to the time of residence granted to the tokens in the timed places. In an *FMS* context, it should be noted that the delay in residence (stay) of the tokens in a place can mean the expiration time of primary material or semi-finished material. So, our work focuses on the Workflow performance aspect through Stochastic Petri Nets (*SPN*) [46] [47] [51] use. Hence, we propose a SPN extension, named Workflow P-Timed Stochastic Petri Net (*WPTSPN*), able to specify, check *FMS*-workflow, also predict time of stay (residence) in activity lifecycle.

In other words, we provide a P-timed SPN extension that provides expressive specification, verification, and performance analysis of *FMS* business processes. So, the *WPTSPN* makes it possible to represent the expressiveness of a *FMS* business process, to manage and monitor its phases, then to identify any type of execution failure, and to build scenarios of cooperation between actors and their roles in case of failure. Also, this model' stochastic paradigm favors the delay prediction.

Knowing that several works related to the forecast of the risks of delay have elaborated:

- In the work orientation indicated in [16], one must emphasize the need to sensitively manage business operations according to risk. In [52] there is a link to a recent survey on the question. According to this study, the majority of prior studies focused on that topic from a qualitative perspective. Some quantitative research methods have recently been offered. The risk indicators for non-compliance with deadlines are described in works [27] [52] and [53]. Also, the work in [7] investigates and analyzes the models' goals, such as anomalous activity times, and forecasts a case of delay.
- The authors in [52] [53] provide a risk-based simulation prototype design that incorporates resource risks such as hardware failures and business process actor illnesses.
- The model in [7], which is assumed to be stochastically validated, must include a monitoring system that captures and documents the business process stages, just as the models in [53] and [54]. In this hypothesis, the occurrence earlier-time is applied as an information source to generate an exact accuracy for the representative dynamic system.

Ultimately, P-timed PNs capture time requirements in the model structure. On the one hand, according to the P-timed semantics, it is necessary to assign to each token a local clock (a stopwatch) which measures the time from the moment of arrival in the place. On the other hand, the addition of these counters is useful to check and control the state of non-occurrence of dead tokens (violation of residence time constraints). Moreover, this state of a timed PN is characterized by the current marking and by the value of the clocks associated with the tokens. The latter has very negative consequences for the complexity of the analysis. However, this drawback is only due to the effective complexity of the modeled problem. In our approach, the tools enriching the model make it possible to calculate the cycle time, the times of firing, and the operating margins of the system in mono-periodic operation in residence and firing times. Based on the work performed in [47] and [50], we extend our model with formal tools that ensure adaption to P-timed semantics. According to the splitted (division) mechanism, this model uses the analytical verification characteristics of the two-level transitions and their related micro-transitions. The density gravity center method is used to identify the firing duration and average random residency of a token in each model place, similar to the method used in [48]. Additionally, the current model adopts a strategy based on elapsed time randomly attached to tokens in timed places specified by randomly residence intervals. We note that randomly interpreting residency intervals is different from the approach in ff which interprets static stay intervals [47] [50] and [54].

So our *WPTSPN* accomplishes a verification method based on checking model properties that are related to elapsed time which is randomly stay assigned to tokens in timed places like as in [30] [47] [50] [54]. We apply our model and its adjacent tools to an *FMS* example which corresponds to the production chain sub-section presented in [4].

The remainder of this paper is divided into the following sections:

- The basic concepts of our proposed model are defined in Section 2;
- The suggested *WPTSPN* is depicted in Section 3;
- Section 4 focuses on the real-world implementation of our concept and related tools in the field of *FMS* Workflow;
- The final section summarizes our work and suggests future research directions.

2. Definitions and Basic Concepts

We define the concept set to introduce our paradigm and the associated formal tools to specify, verify and analyze a complex system such as a *FMS*-workflow.

2.1. Workflow

To present the strategic, organizational, and operational functions of the management process, we define *Workflow* [6] [7]. To understand and assess the existing parties in a socio-economic organization, it is crucial to understand the business processes in the organizational structure. They advocate for flexible information system design and deployment. These information systems' technical foundation is the realization of new products through adaptation to change and market demands. Given the context of managing and guiding businesses, business process management is innovative. According to [5] a business process is any set of actions that include one or more types of input and offer a customer-satisfactory outcome. This definition covers the relationships' semantics between the inputs and the outputs (pre-condition and post-condition) in a business process. So it is an abstraction of the concept of the process that becomes clear in this definition.

However, a workflow [12] [14] [55] ensures the automation of formal rules to restructure, simulate, and optimize the company's business processes. According to [56], the workflow optimization procedure, therefore, concerns the definition of the scheduling of fundamental interdependencies, the representation flexibility, at the time of the action, certain routes integrating the data specific to an instance particular process [18] [56].

2.2. Petri Nets

In 1962, Carl Adam Petri innovated a formal modeling tool called the Petri Net [57]. They emerged as a powerful graphical tool for the representation of complex system and sequential mechanisms [57]. They have effectively been used for the specification, verification, and analysis of the behavior of complex systems [58]. With a variety of *PN*s [59] [60] [61] [62] application fields and their efficient uses to specify and verify the dynamic behavior of complex systems, *PN*s several classes have appeared in the literature. Using the advantages of the SPN to complex systems analysis and verification, we suggest a model that covers the life-cycle of the manufacturing process workflow. To specify, validate, and predict the execution delay associated with the duration of the tokens propagating within firing transitions, we propose a P-timed oriented extension of the SPN [63] [64]. This proposed model, as well as the tools for specification, verification, and evaluation, is presented in the following section.

3. Workflow P-Timed Stochastic Petri Net Model (WPTSPN)

Our *WPTSPN* is an extension of the *SPN* that adds new components (timed places, tokens, and transitions) for the FMS-Workflow context. The basic formal concepts of the *WPTSPN* are covered in this section. The model's basic characteristics are formulated, interpreted, and illustrated after it has been defined. In addition, this proposed model incorporates *WFMS* notions [55] [65], with refinements attributed in [6] and [25].

3.1. Model Definition

The Workflow P-Timed Stochastic Petri Net, denoted *WPTSPN*, is the 8-uplets:

WPTSPN:

=<*P*, *Tr*, *A*, *Sac*, *Ac*, *Typ-tk*, *GTyp-P*, *CardM*>

- $P = P_b \cup P_w \cup P_r \cup P_f$, represents the finished set of places;

- P_b : all initial places,
- P_w : all workflow places,
- P_r : all resource places,
- P_f : all final places.

- $Tr = Tr_i \cup Tr_d \cup Tr_s$, represents the finished set of transitions (non-empty finite set);

- Tr_i : immediate transitions set,
- Tr_d : determinstic transitions set,
- Tr_s : stochastic transitions set.

- *A*: set of Workflow-actors is non-empty finished;
- *Ac*: workflow activities finished set;
- *Sac*: finished set of Sub-activities associated with a Workflow-activity;
- *Typ-tk*: the representation by our model distinguishes between four token types (activity, sub-activity, actor, and any encapsulation of tokens) then the *Typ-tk*-function is in charge of this classification.

$$Typ\text{-}tk : tk \to \{1,2,3,4\};\ tk \mapsto \begin{cases} 1,\ tk \in A \\ 2,\ tk \in Sac \\ 3,\ tr \in Ac \\ 4,\ tk \in uplet \end{cases} \tag{1}$$

- *GTyp-P*: the representation adopted by our model requires that we distinguish the ordinary places of a PN, those linked to the semantic workflow, and those endowed by a residence time. So, *GTyp-P*-function makes it possible to distinguish the model's places.

$$GTyp\text{-}P : P \to \{o, w, r, f\}$$

$$p \mapsto GTyp\text{-}P(P) = \begin{cases} o, \ (\text{initial place}) \\ w, \ (\text{workflow place}) \\ r, \ (\text{resource place}) \\ f, \ (\text{final place}) \end{cases} \tag{2}$$

- *CdM*: the marking cardinality of a *WPTSPN* is a family indexed by the set of places *P*, such that tokens (*tk*) in place *p*:*M*(*p*) is an element of *CdM*(*M*(*p*)).

The proposed *WPTSPN* aims to represent (specify) a flexible production business process according to a P-timed orientation, to then check (verify) the model properties that coincide with the domain of application, and finally to predict the delay relating to the execution of an activity (firing a non-immediate transition of the model). So to achieve these ends, we enrich our model with formal functions.

3.2. Formal Tools Enriching the *WPTSPN* Model

In this subsection, we enriched our model with the functions to increase the analytical coverage according to the workflow semantics, P-timed orientation, and the random residence time assigned to the tokens during a sequence of firing transitions. These <*GDensity*, *GDistrib*, *TFringT*, *IRP*, *MST*, *GTruncated*, *MIP*, *MPO*, *MIT*, *MTO*> formal tools are defined as follows:

- *GDensity*: the probability density function of the *WPTSPN* components (tokens, and stochastic transitions) represents a stochastic probability density (Normal, Log-Normal, or Exponential).

$$GDensity : (Ac \cup Sac \cup Tr_s) \times \{e, n, ln\} \to \mathrm{R}^+$$
$$typ = Typ\text{-}dist(k)$$

$$(k, typ) \mapsto GDensity(k) =$$
$$\begin{cases} typ = e, \lambda \cdot e^{-\lambda x} \ (\lambda \geq 0, x \geq 0); \\ typ = n, 1\big/\left(2\sqrt{2\pi}\right) \cdot e^{-1/2 \cdot x^2}, \left(x \geq a_p\right); \\ typ = ln, 1\big/ x\sigma\sqrt{2\pi} \cdot e^{-\left(\left(\ln x - \mu\right)^2 / 2\sigma^2\right)}, \left(x \geq a_p, \sigma \in \mathrm{R}^+, \mu \in \mathrm{R}\right). \\ a_p = \min\left(IRP_p\right) \ \left(\forall tr_j \in {}^{I}p\right) \text{and} \left(GTyp\text{-}P(p) \in \{o, w\}\right). \end{cases} \tag{3}$$

- *GDistrib*: each *WPTSPN* component (tokens, and stochastic transitions) is assigned an arbitrary probability distribution (Normal, Log-Normal, or Exponential). This assignment is performed by the function *GDistrib*.

$$GDistrib : (Ac \cup Sac \cup Tr_s) \times \{e, n, ln\} \to \mathrm{R}^+$$
$$typd = Typ\text{-}dist(k)$$
$$typ\text{-}k = Typ\text{-}tk(k)$$

$$\left(k,typ\text{-}d,typ\text{-}k\right)\mapsto GDistrib\left(k\right)=$$
$$\begin{cases} typ=e,1-e^{-\lambda x}\left(\lambda\geq 0,x\geq a_p\right); \\ typ=n,1/\left(2\sqrt{2\pi}\right)\cdot\int_{-\infty}^{t}e^{-1/2\cdot x^2}dx,\left(x\geq a_p,t\geq a_p\right); \\ typ=ln,1/2+1/2\cdot erf^{-\left(\left(\ln x-\mu\right)^2/2\sigma^2\right)}, \\ \left(x\geq a_p,\sigma\in\mathrm{R}^+,\mu\in\mathrm{R},erf\text{ is the Gauss error function}\right); \\ a_p=\min\left(IRP_p\right)\left(\forall tr_j\in{}^{I}p\right)\text{and}\left(GTyp\text{-}P\left(p\right)\in\left\{o,w\right\}\right). \end{cases} \tag{4}$$

- *TFringT*: Our model assigned three probability distributions (Normal, Log-Normal, or Exponential) to each stochastic transition [66]. This function makes it possible to determine each stochastic transition firing time, with *TFringT*-function, according to the distribution chosen. We use the gravity center method [48] which consists of taking the abscissa corresponding to the gravity center of the each stochastic density transition.

$$TFringT:Tr_s\times\left\{e,n,ln\right\}\to\mathrm{R}^+$$
$$typ=Typ\text{-}dist\left(trs\right)$$

$$\left(trs,typ\right)\mapsto TFringT\left(trs\right)=$$
$$\begin{cases} typ=e,\dfrac{\int_{-\infty}^{t}\left(\lambda\cdot e^{-\lambda x}\right)\cdot xdx}{\int_{-\infty}^{+\infty}\left(\lambda\cdot e^{-\lambda x}\right)dx}; \\ typ=n,\dfrac{\int_{-\infty}^{+\infty}\left(\left(1/\left(2\sqrt{2\pi}\right)\cdot e^{-1/2\cdot x^2}\right)\cdot x\right)dx}{\int_{-\infty}^{+\infty}\left(1/\left(2\sqrt{2\pi}\right)\cdot e^{-1/2\cdot x^2}\right)dx}; \\ typ=ln,\dfrac{\int_{-\infty}^{+\infty}\left(1/x\sigma\sqrt{2\pi}\cdot e^{-\left(\left(\ln x-\mu\right)^2/2\sigma^2\right)}\cdot x\right)dx}{\int_{-\infty}^{+\infty}\left(1/x\sigma\sqrt{2\pi}\cdot e^{-\left(\left(\ln x-\mu\right)^2/2\sigma^2\right)}\right)dx}; \\ \left(x\in\mathrm{R}^+,\lambda\in\mathrm{R}^+,\sigma\in\mathrm{R}^+,\mu\in\mathrm{R}\right). \end{cases} \tag{5}$$

- *IRP*: IRP_i function defines the static residence (stay) interval of the model's places (initial or workflow). ($\forall p_i\in\left(P_o\cup P_w\right)\setminus GTyp\text{-}P\left(p_i\right)\in\left\{O,w\right\}$). When it leaves the place p_i, the output transitions firing becomes equal to b_i. After this random time (b_i), the token will be "dead" and will no longer participate in the transitions validation. Formally *IRP* is expressed as follows:

$$IRP:P_o\cup P_w\to\mathbb{Q}^+\cup\left(\mathbb{Q}^+\times+\infty\right);$$

$$\forall p_i\in P_o\cup P_w\setminus GTyp\text{-}P\left(p_i\right)\in\left\{o,w\right\};$$

$$p_i\to IRP_i=\left[a_i,b_i\right]\text{ with }0\leq a_i\leq b_i;$$

$$\begin{cases} a_i=\min\left(a_i,b_i\right)\left(\forall\ tr\in{}^{I}p\right); \\ b_i=\max\left(a_i,b_i\right);\left(\forall\ tr\in{}^{I}p\right); \end{cases} \tag{6}$$

- *MST*: (Mean Stay Time) Taking into account the specificity of the token's type (activity or sub-activity) in places of our model. So, the *MST*-function calculates each token's average stay time in timed locations and which are of type (initial or workflow). We associate time value, framed by the $\min\left(TFringT\left(tr\right)\right)$ ($\forall tr \in {}^{I}p$) and the $\min\left(TfiringT\left(tr\right)\right)$ ($\forall tr \in p^{O} \setminus \left(GTyp\text{-}P\left(p\right)\right) \in \{O, w\}$, $Typ\text{-}tk\left(M\left(p\right)\right) \in \{2,3,4\}$), taking into account the random distribution assigned to each token in this place. Practically, we use the gravity center method [48] which determines the abscissa corresponding to the density attributed to the place in question.

$$
\begin{aligned}
&MST : \left(P \cup Ac \cup Sac \times Tr_s \times \{e, n, ln\}\right) \to \mathrm{R}^{+} \\
&ty = GTyp\text{-}P\left(p\right) \in \{o, w\} \\
&typ = GTyp\text{-}dist\left(ttk\right) \\
&ttk = Typ\text{-}tk\left(M\left(p\right)\right) \\
&m = \min\left(TFringT\left(tr\right)\right) \forall tr \in {}^{I}p \\
&M\ = \min\left(TFringT\left(tr\right)\right) \forall tr \in p^{O}
\end{aligned}
$$

$$
\left(p, ty, typ, ttk\right) \mapsto MST\left(p\right) =
\begin{cases}
ty \in \{o, w\}, typ = e, ttk \in \{1,2\}, \dfrac{\int_m^M \left(\lambda \cdot e^{-\lambda x}\right) \cdot x dx}{\int_m^M \left(\lambda \cdot e^{-\lambda x}\right) dx}; \\
ty \in \{o, w\}, typ = n, ttk \in \{1,2\}, \dfrac{\int_m^M \left(\left(1/2\sqrt{2\pi} \cdot e^{-1/2 \cdot x^2}\right) \cdot x\right) dx}{\int_m^M \left(1/2\sqrt{2\pi} \cdot e^{-1/2 \cdot x^2}\right) dx}; \\
ty \in \{o, w\}, typ = ln, ttk \in \{1,2\}, \dfrac{\int_m^M \left(1/x\sigma\sqrt{2\pi} \cdot e^{-\left(\left(\ln x - \mu\right)^2 / 2\sigma^2\right)} \cdot x\right) dx}{\int_m^M \left(1/x\sigma\sqrt{2\pi} \cdot e^{-\left(\left(\ln x - \mu\right)^2 / 2\sigma^2\right)}\right) dx}; \\
\left(x \in \left[m, M\right], \lambda \in \mathrm{R}^{+}, \sigma \in \mathrm{R}^{+}, \mu \in \mathrm{R}\right). \\
m < \min\left(IRP\left(p\right)\right) \text{ and } M < \max\left(IRP\left(p\right)\right).
\end{cases}
\tag{7}
$$

- *GTruncated*: We use a truncated density function to estimate and predict the delay related to the residence time of a token in a timed place. And for it to cover the densities allocated to the tokens' residence time in each timed place. We define the generalized truncated density sojourn probability law of a timed place function as follows:

$$
\begin{aligned}
&Truncated : \left(P \cup Tr_s\right) \to \mathrm{R}^{+} \\
&\left(t_0, t\right) \in TD \times TD \\
&typ \in \{e, n, ln\} \\
&ttk \in \{1,2,4\} \\
&\left(k, typ\right) \mapsto GDensity\left(k\right) \\
&\left(k, typ\right) \mapsto GDistrib\left(k\right)
\end{aligned}
$$

$$Truncated\left(k, ttk\right) = \begin{cases} 0,\ t \le t_0, GDistrib\left(t_0\right) \le 1; \\ GDensity\left(t\right)/1 - GDistrib\left(t_0\right), t > t_0, GDistrib\left(t_0\right) < 1; \\ GDensity_{\Delta}\left(t - t_0\right), GDistrib\left(t_0\right) = 1; \end{cases} \tag{8}$$

- Let MIP: the places input-transitions matrix:

$$\left(P \times Tr\right) \to \{0,1\} \tag{9}$$

Such that:

$MIP\left(P_i, tr_k\right) = 0$, if ${}^{I}P\left(P_i, tr_k\right) = \varnothing$, 1 if not

- Let MPO: the places output-transitions matrix :

$$\left(P \times Tr\right) \to \{0,1\} \tag{10}$$

Such that:

$MPO\left(P_i, tr_k\right) = 0$, if $P^{O}\left(P_i, tr_k\right) = \varnothing$, 1 if not.

- Let MIT: the transitions input-places matrix:

$$\left(Tr \times P\right) \to \{0,1\} \tag{11}$$

Such that:

$MIT\left(tr_k, P_i\right) = 0$, if ${}^{I}Tr\left(tr_k, P_i\right) = \varnothing$, 1 if not

- Let MTO: the transitions output-places matrix:

$$\left(Tr \times P\right) \to \{0,1\} \tag{12}$$

Such that:

$MTO\left(tr_k, P_i\right) = 0$, if $Tr^{O}\left(tr_k, P_i\right) = \varnothing$, 1 if not

The firing transitions semantics adopted by the $WPTSPN$ model are defined in the next sections.

3.3. Firing Transitions Semantics Adopted by the *WPTSPN*

The firing semantics adopted by the $WPTSPN$ depend on three factors: the static stay (residence) interval of the places at the transition input, the stochastic firing time, the average stay time of tokens in workflow entry places, and preconditions related to the interaction between the actors responsible for executing the activity represented by the transition. To explain this firing mechanism, we need to apply it to the context of the transition tr_6. According to Table 3, the transition Tr_6 which represents the operation below ("Machine run").

Figure 1 illustrates the tr_6 transition and the mechanism of division (splitted) into three micro-transitions: $\left\{tr_6^c, tr_6^w, tr_6^p\right\}$.

The transition tr_6 is stochastic according to the $WPTSPN$, so we assign a random value of firing time using $tydt = GTyp\text{-}dis\left(tr_6\right) \in \{e, n, ln\}$ and $TFringT\left(tydt, tr_6\right)$.

We note that according to the representation semantics of the WPTSPN: Let input places: ${}^{I}Tr_6 = \left\{P_0, P_7, P_8\right\}$ and output place: $Tr_6^{O} = \left\{P_{10}, P_0, P_8\right\}$.

- P_0: specifies the park of actors, $GTyp\text{-}P\left(P_0\right) = r$;
- P_7: specifies the both a machine bounded input and conveyor output buffer, $GTyp\text{-}P\left(P_7\right) = w$;

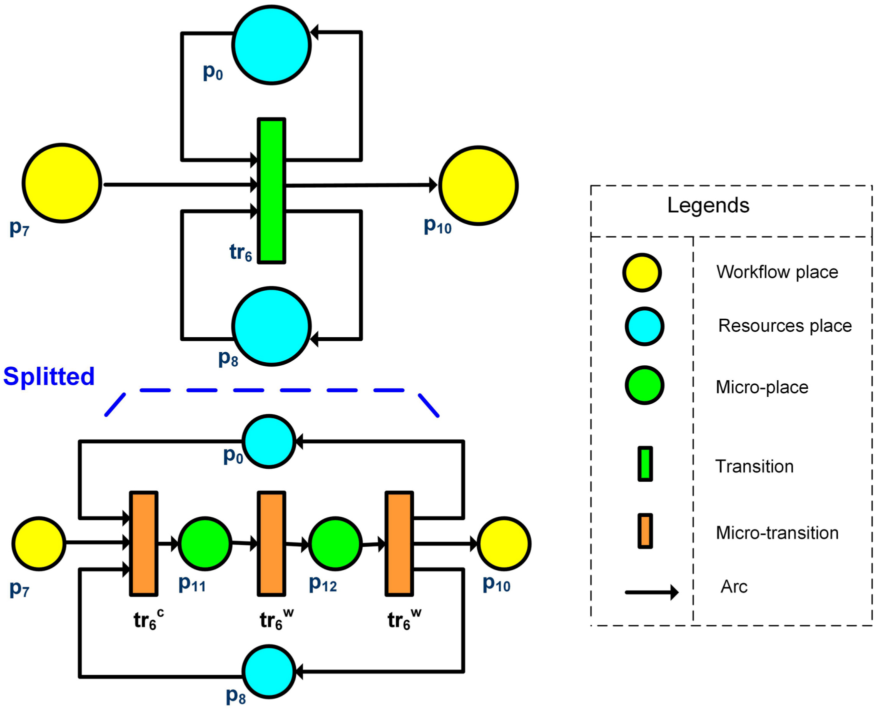


Figure 1. The tr_6 transition and their micro-transitions and micro-places according to the division (splitted) mechanism of the *WPTSPN*.

- P_8: specifies place of the maintenance sub-process, $GTyp\text{-}P(P_7)=r$;
- P_{10}: specifies the end of workflow process $GTyp\text{-}P(P_7)=f$;.

It is assumed that non-workflow places (P_0, P_8), *i.e.*, resources, are not timed, so the tokens they store are always available, such as actors, primary materials, or semi-finished products. The timed place P_7 is a workflow place dedicated to storing an activity or sub-activity type token (noted *tk*: $Typ\text{-}tk(tk)=\in\{2,3\}$). Then we assign a static interval of stay (residence): $IRP(P_7)=[a_7,b_7]$.

At the moment when the token named *tk* such that *Gtype*(*tk*) = 2 (*i.e.* of type *Sac*) instead of p_7, we assign it a stochastic value of stay using the density function ($\tau_{tk}=Gdensity(tk,ty)$) where *ty* varies between the three possible distributions of the model $\{e,n,ln\}$. If there are several (*Gtype*(*tk*) = 2) type tokens, an average residence time is used using the $MST(P_7)$ function.

The firing semantics of a stochastic transition with timed places assumes that a time counter is triggered when a token arrives. Let t be the value of this time counter. We associate a token-life-cycle time named (τ_{tk},t). We discuss the following three cases:

- $\tau_{tk}<a_i(P_7)$; so the *tk* token is not available.
- $t\in[a_7,a_7+\tau_{tk}]$ and ($a_7+\tau_{tk}<TFiring(Tr_6)<b_7$); During Tr_6 firing, *WPTSPN* detects the transition type via the function calls *Distinct*(Tr_6) = 3 and *GTyp-dist*(Tr_6) = *ln*. We know that the overall firing time value is determined by the function call *TFiringT*(Tr_6, *ln*). The transition firing Tr_6, which is in the sub activity executing charge Sac_1, consists in sequentially firing the three micro-transitions Tr_6^c, Tr_6^w and Tr_6^p. The transition pre-condition

Tr_6^c is the *uplet* <11, *time*, 1> and the stochastic value ($0.33*\varphi_3$). The micro-transition firing Tr_6^w is influenced by the stochastic value ($0.67*\varphi_{11}$). Once the function activity call is done, T_6^w micro-transition firing is achieved. The T_6^p firing is immediate; *i.e.*, the token Sac_1 is moving towards the place P_{10} and the two tokens A_1 and A_3 will be released to the place P_0 (the original place before firing Tr_6), also a token in place P_8.

- $\tau_k > b_7$; so the *tk* token is dead. We note that the tokens which circulate in the three micro-transitions have the same behavior according to the static residence time. Note that each micro transition from the split mechanism inherits a behavior similar to the transition (splitted).

The main properties that generally studied the *WPTSPN* are verification and checking. First, verification consists in determining the techniques for deciding on each of them. Second, checking, which is based on the dynamic timed evolution to analayze and verify this model properties, is then presented.

3.4. *WPTSPN* Checking Formal Tools

The *WPTSPN* model can be used to qualitatively analyze the basic *FMS* properties to provide the dynamic behavioral aspects of complex systems. The *CdM* (marking cardinalities graph) associated with the model at any time in time represents the number and position of tokens concerning the places of the model. Firing a transition moves the model into a new marking cardinality. The marking cardinality of an *WPTSPN* helps to define which places are occupied and which are free.

The model adopts two verification approaches:

- Checking based on the elapsed time following a firing transitions sequence:

- Reachability (Accessibility): A *CdM'* marking cardinality is said to be accessible from *CdM* if there are sequences of markings cardinalities (CdM_0 to CdM_n) and transitions (tr_0 to tr_n , $\forall(tr_0, tr_n) \in (Tr \times Tr)$) such as CdM_{i+1} is immediately accessible from *CdM'* ($i \in [0, n-1]$) with $CdM_i = CdM_0$ and $CdM' = CdM_n$. The set of each markings cardinalities accessible from *CdM* is noted the set of reachability.

  So reachability seems to be a tool for finding erroneous states, for practical problems the constructed graph usually has far too many states to compute. This allows us to detect execution failures of a business process (FMS) modeled by a *WPTSPN*.

- Safeness: An *WPTSPN* is safe if each of its places ($\forall p \in P$) is safe. So, a place p is safe if it does not contain more than one token. The model transitions if it can fire infinitely often, so if there is always a fixed (necessarily infinite) firing sequence that ensures the execution of a workflow activity.

- Boundeness: A *WPTSPN* is said to be bounded if it is composed only of bounded places. A place ($\forall p \in P$) is bounded if there is a limit on the tokens maximum number ($CdM(p) \le K \backslash(\forall p \in P, K > 1)$).

  This property indicates that the system represented by a *WPTSPN* never risks exceeding the storage limits of semi-finished material, finished products, and workers… This property indicates that the system represented by a *WPTSPN* never risks exceeding the storage limits of semi-finished material, finished products, and workers…

- Conservativeness: A *WPTSPN* is said to be conservative if the tokens-number in the $\forall CdM \in CdM$ remains constant.

  It is obvious that conservativeness is a special case of structural delimitation and that coherence is a case of repetitiveness. Then this property can be partial, consistency and repetitiveness are the relaxations of the circulation of tokens in the places of the *WPTSPN*.

- Liveness: A transition is said to be alive if for each ($\forall CdM \in (CdM, WPTSPN)$) there is a firing sequence which brings the *WPTSPN* to a *CdM* in which this *tr* transition is fired. An *WPTSPN* is active if each transitions ($\forall tr \in Tr$) are active. If the *WPTSPN* is activated and correct, this indicates that there are no deadlocks in the *FMS* operations.

- Properness: if the initial marking CdM_0 is accessible from each marking ($\forall CdM \in (CdM, u)$) in the accessibility set, the *WPTSPN* is said to be proper.

  This property verifies that the tokens which represent the actors (worker) return to their place of origin before the each firing model's transition for which they will always be available at the future firing (excution of an activity workflow FMS).

- Checking is based on randomly residence intervals associated with the token in workflow-places and the token life-cycle:

We keep the verification properties related to the firing transitions context, and we add other definitions related to the *P-timed SPN* contexts that we adopt in this work. Let *DT* is a finite time domain; Let *t*: time in finished time domain ($\forall t \in DT$); Let state characterizing the *WPTSPN* graph situation be defined by a doublet *SE* = <*CdM*, *Q*>, where:

- *CdM* is the same as the previous definition of the *WPTSPN* marking function,
- *Q* is a time-of-stay application which associates with each token *k* ($\forall k \in \{SAC \cup AC\}$) in place p_i ($\forall p_i \in P \setminus Typ\text{-}tk(M(p_i)) \in \{1,2,4\}$) a real number q_i^k where q^k is the this token lifetime (the time elapsed since its arrival in place p_i). The q_i^k associated with a token *k* in the place p_i must be less than or equal to bi where $[a_i, b_i]$ is the Randomly residence interval associated with the place p_i. This token *k* in the place p_i is involved in the activation and validation of its p^O transitions when q_i^k *age* great. The token is considered *dead* when its lifespan (*age*) is strictly greater than b_i.

Note that the lifetime q_i^k of a token $k \in (Ac \times Sac)$ (given by a local clock associated with it) is relative to this token arrival time in the place $p_i \setminus Typ\text{-}tk(M(p_i)) \in \{1,2,4\}$. We suppose that the token *k* arrives in the place p_i at the instant τ (given by a global clock for example), this token time duration is equal to zero. At the absolute instant τ' its lifetime is $q_i^k = \tau' - \tau$. It participates in the validation of the output transitions of the place ($\forall tr \in p_i^O$ which contains it, only from the instant $\tau' = \tau + a_i$, and it is dead from the instant $\tau' > \tau + b_i$. Then, a token is dead if its lifetime becomes strictly greater than the upper limit of the dynamic interval associated with its host place and if none of its exit transitions is validated at this time. Therefore, a transition can be activated and validated in the semantics of autonomous *PNs* and not in the *WPTSPN*'s semantics because of time constraints.

- A transition ($tr \in T$) is potentially firable (validated in the semantics of model) from the state $SE(CdM, Q)$ if and only if:

- It is validated within the meaning *WPTSPN*s in this state, that is to say $\forall p_i \in {}^I tr; CdM(p_i) \ge Card(I(p_i, tr)); Typ\text{-}tk(M(p)) \in \{1,2,4\}$ (*Card*: function which counts the tokens number in input flow or respectively output flow) $\forall p_i \in {}^I tr$, there is at least $I(p_i, tr)$ tokens in the place such as: $\min(b_i - q_i^k) - \max(0, \max(a_i - q_i^k)) \ge 0$ $(\forall k \in \{SAC \cup AC\})$ and $[a_i, b_i]$ is the dynamic interval associated with the place p_i. In addition, there are no *j* tokens (which do not participate in firing the *tr* transition) such as: $(b_i - q_i^k) \le \max(0, \max(a_i - q_i^k))$.
- Otherwise, this token is *dead*. It is, then, associated with this place interval: $\left[\max(0, \max(a_i - q_i^k)), \min(b_i - q_i^k)\right]$.

Figure 2 illustrates the token life-cycle according to the firing semantics of the *WPTSPN*.

When we perform in the intersection of all these intervals ($\forall p \in {}^I tr$, we associate an interval), we select the interval in which the transition remains potentially firable. According to the state *SE* notion, from a given state, an infinite number of states is generally accessible. There are, therefore, two possibilities of having this new state from an earlier state:

- the time fluidity (time passage);
- the transition firing (the potential firing interval lower limit is equal to zero from this new state).

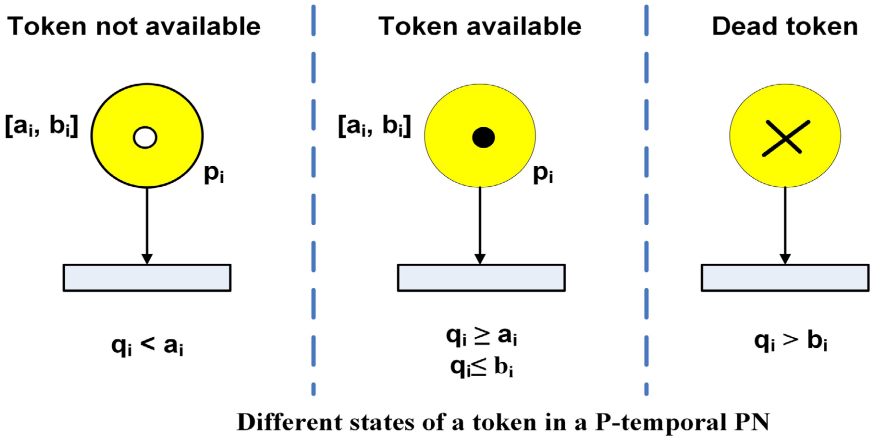


Figure 2. Token life-cycle according to the firing semantics of the *WPTSPN* model.

- The state *e*' accessibility from a previous state *e* ($(e,e')\in SE\times SE$):

- by a τ time-flow: we note $e(CdM,Q)\rightarrow e'(CdM',Q')$, is verifiable if: $CdM\neq CdM'$, $\forall k\in(SAC\cup Ac)$ in the $p_i\setminus Typ\text{-}tk(M(p_i))\in\{1,2,4\}$ place, $q_i'^k=q_i^k+\tau$; where q_i^k (respectively $q_i'^k$) is the token k lifetime in state *e* (respectively in state *e*') and b_i is the dynamic interval upper bound associated with the place p_i.
- by firing a transition tr_i if and only if: tr_i can be firing directly (the lifetimes of all the marks which validate tr_i are greater than or equal to the dynamic intervals lower bounds associated with their places) from *e*, $\forall p\in P_w$, $CdM'(p)=CdM(p)-CdM(I(p,tr_i))+CdM(O(tr_i,p))$. When the tokens do not flow and keep the same lifetime in *e* and *e*' (this is assumed to be an immediate transition), then we assume those tokens that are flowed or created have a lifetime (age) equal to zero. We note that the preceding firing rule allows us to determine the states and the state-space accessibility relations. The set of firing transitions' sequences from the initial state characterizes the dynamic behavior of the *WPTSPN*.

According to the state analysis, either by time fluidity or by firing a transitions sequence, the *WPTSPN* verifiable properties are:

- An *WPTSPN* model's state $e(CdM,Q)$ is tokens-alive if all of the tokens in *CdM* are alive.
- A *WPTSPN* is token-alive for an initial CdM_0 marking (the initial state e_0) if all the markings accessible from CdM_0 are token-alive states.

We adopt the hypothesis that if a token, in a cardinality marking a state accessible from e_0, is verified dead, the *WPTSPN* is dead tokens. Consequently, this dead token no longer participates in the transitions validation. There are two objectives through which we can verify the token death upstream of a synchronization transition *tr*:

- the tokens on circuits containing it do not arrive at a compatible time in its entry places ($\forall p\in {}^{I}tr$),
- the tokens, on the transition elementary oriented paths from parallelism to *tr*, do not arrive at compatible instants upstream of the transition *tr*.

- A transitions firing sequence (made up of the transitions and the associated firing time $((SE,tr);\forall tr\in(Tr_s\cup Tr_d))$ is called a token-liveness sequence (k-v-sequence) if and only if each of the states accessible ($\forall e\in SE$) by this transition sequence is token-alive states.
- A transition firing sequence (*SE*, *u*) is repetitive if, for a state $\exists e\in SE$ reachable from e_0 (the initial state of graph states), *e* is accessible from *e* by the (*SE*, *u*) sequence transitions firing.
- A transition firing sequence (*SE*, *u*) is complete if it contains each of the transitions ($u\subseteq Tr$) composing the *WPTSPN* model. The *WPTSPN* model checking P-timed properties definitions help to adopt the tow following assumptions in [30]:

- If an *WPTSPN* has a repetitive and complete (*tok-v-sequence*), then this model has at least one operation which results in a steady-state after a finite time.

- If an *WPTSPN* is boundless, alive, and alive-tokens then each operation of the model results in a steady-state after a finite time.

After verifying the *WPTSPN* model's probabilities according to two P- and T-timed orientations, we show our formal mechanism for predicting of the delay relative to the residence time of sac-token in timed places.

3.5. Delay Prediction Relative to the Randomly Residence Time in Model Place

In this subsection, we demonstrate the specificity in the delay prediction that is related to the token' randomly residence time assigned in the model' timed places. We include the *Event-log* related to *WPTSPN* model to provide delay semantics in the activities' description. Let *TD* represent the temporal domain associated with the state firing sequence *SE*.

So, we define an *Event Log* by *LEv* = (*SE*, *C*, *D*):

- *SE*: states' firing sequence set,
- *C*: cases finite set where each case specifies an *FMS* business process instance,
- *D*: $SE \rightarrow TD$. A function to assign each event to time-space and classify it as a time function.

Each deterministic and stochastic transitions in the *WPTSPN* ($tr \in \left(Tr_d \cup Tr_s\right)$) are covered by the set *SE*. And, each *SE*/*CdM* states in the event log contain the *C* set. We can utilize a time capture approach based on a probabilistic aspect to assure the prediction of the delay associated to task execution in the business processes represented by our model. As a result, this model was integrated with an event-log (*Eventlog* : *LEv*) to predict remaining durations as well as the probability of execution delays. To induce the elapsed time required by the model, we use the same definition of a sequence of delays $\left\{d_0, d_1, d_2, \cdots, d_n\right\}$, where $d \in TD$ is the time elapsed before the occurrence of se_0 and d_i is the elapsed time between occurrences of se_i and ($\forall se_i \in SE$) The prediction method is based on:

- The enrichment of places with probabilistic delays in order to determine the duration of an activity and the associated stay time.
- To determine the probability of a decision outcome, assign weights to places.
- The probability density function associated with stochastic places determines the randomly residence time distribution.
- Using an adjacent algorithm in the *WPTSPN* to predict the residence delay based on the elapsed time.

A monitoring system that collects and records the *FMS* business process steps is required for our *WPTSPN* to be stochastically verified. According to this assumption, the time since the previous occurrence represents an information source that benefits the preacher in acquiring precise accuracy for the model. With the duration distribution, it is possible to visualize the conditional impacts on the remaining time. The formal representation is as follows:

- $t \in TD$: time space,

- $tr \in (Tr_d \cup Tr_s)$: timed transition,
- $P_i \in P_w$: timed place,
- $tk \in Ac \cup Sac$: token,
- $GDistrib(k) k \in (P \cup Tr_d \cup tr_s)$: the duration distribution function,
- $GDensity(k) k \in (P \cup Tr_d \cup tr_s)$: The differential function $GDistrib(k)$ becomes the generalized density function.
- $Gtruncated_P$: the generalized truncated density function.

In a formal way, we denote that ($t \in TD$) represents time, *tk* token ($Typ\text{-}tk(tk) \in \{2,3,4\}$) and ($P_i \in P_w$) is a timed place with the assigned static residence. Let ($t_0 \in TD$; $t_0 > 0$) be *tr* the current place time sensor. Let $Gtruncated_\Delta$ be the *truncate delta function* which captures the entire probability mass at a single point.

We aim to build an algorithmic prediction solution based on the immediate detection of events. It is assumed that the duration of each stay place is purely isolated from other independent activities. This hypothesis allows us to simplify the forecasting process while remaining within the norms of other prediction approaches.

To perform the forecast on a state case, we build the forecast algorithm with six entries:

- business process model, its *states Graph* representative or markings cardinalities graph,
- current moment noted t_0,
- minimum time limit: $t_m = Min(TFiringT(tr))(\forall tr \in {}^I P_i)$,
- maximum time limit: $t_M = Min(TFiring(tr))(\forall tr \in P_i^O)$,
- *Nbt*: activity Cases total number,
- activity current trace *c* which represents all the events observed up to the moment t_0.

The organizational chart illustrates the prediction process, which begins with a case activity search taking place in the model or one of its representative *SE* states. It's a *SE* graph in the *WPTSPN* that specifies all of the observed events in this scenario.

Notation: we use the same concept which is an alignment method for localizing the current state in the business process model by adjusting the (place/transitions). The next step is to arrange the results of the probabilistic simulation (t_m, t_M, *MST*(*P*), *se*, *Nbr*: iterations number). Based on the *WPTSPN* firing sequence states, the simulated case probability distribution type is established for each iteration. Despite the fact that their represented place's probability density is the same, all cases are thought to be independent. The simulation operation is responsible for simulating the trace sequences in the restricted sampling model using conditioned truncated distribution densities without analyzing the transition source *GDistrib*(*P*). Then, the conditioned truncated distributions must be raised one more $GTruncated(t)$ ($\forall t \geq MST(P)$) samples.

We must notice that the remaining durations are estimated using the empirical estimation method.

4. Study Case: Flexible Manufacturing Systems

We show in this section the potential of a modeling approach based on the *WPTSPN* model via an application of industrial modeling. This approach is composed of the specification, the verification, and the delay prediction. In this case study, we use our *WPTSPN* to specify, verify, analyze, and simulate the production process.

In addition, the modeling of flexible manufacturing systems (*FMS*) in order to specify, analyze, verify, and optimize a manufacturing system according to different criteria. So it's a performance study to detect any source of failure. This objective is summarized in a study to evaluate and compare several models of manufacturing systems by selecting the most suitable according to specific pre-selection criteria [64]. Our modeling process combines new formal tools specific to *WPTSPN*, with those of *SPN* [63] [64] to formulate a clear and expressive approach to specification, analysis, evaluation, and the checking process. Also, most *FMS'* stochastic modeling studies [67] [68] have focused on steady-state analysis based on metrics such as throughput, productivity, and lifetime [4] [63] [64] [69].

We note that the application of our model and its adjacent tools to an *FMS* example corresponds to the production chain section represented in [4]. In the present work, we rely on mixed specification, analysis, and verification between timed transitions, timed places (random residence time), and lifecycle tokens to detect delay sources. Our approach to modeling a flexible production chain based on the WPTSPN covers concepts related to delays, the temporal properties of transitions, and the circulation of tokens endowed with randomly timed life cycles (activities, sub-activities, actors, etc.).

The *FMS* system architecture [4] (See (**Table 1**)), modeled by *WPTSPN*, is composed of machines served by conveyor belts [4]. The raw material automatically comes from an unlimited buffer representing the raw material stockyard. The time between the arrival of these materials is distributed randomly. Knowing that the raw materials flow into the conveyor belt according to a strategy based on predictions of breakdowns. It is possible to assume that these two operations specify the movements on the conveyor chain. Firstly, the first operation is deterministic (with Delta Time) and designates the transport time between the pallet and the input buffer of the production machine, denoted (location 7) [4]. Suppose that the conveyor belt injects the raw materials into (slot 7) which specifies both a machine-limited input buffer and a conveyor output buffer. Thus, to synchronize logistics, the production system adopts a rule that blocks the conveyor belt when the location is saturated (two rooms in location 7). Secondly, the machine execution service is specified by operation 6. In parallel, the system estimates the lifetime according to a stochastic law. To simplify the calculations, we consider that the service is always available [4]. The control system assumes a breakdown/repair sub-process of the production chain: The time between two successive breakdowns distributed according to a normal distribution [4].

Figure 3 illustrates the production chain [4] represented by Bizagi Modeler [70] adopting the BPMN 2.0 notification. Table 1 explains the business process steps (operations) represented by the *FSM* workflow. The Workflow Nets (*WF-Net*) notion [8] [71], based on a Petri Net modeling, was introduced by Van Der Aalest [72] [73]. Also, the refinements made Van Der Aalest [6] [63] exploit the Petri Net criteria by behaving like an intuitive graphical language which results in Workflow models whose definition is clear and precise. We cannot leave the profile of an end-user in charge of generating Petri nets which are rather the expert's language in the field of mathematical modeling. The conversion [72] [73] from a *WF-Net* to a Petri Net necessitates the interpretation of states (of a Role, a Resource, or a Tool) as "places" and operations (of the same model), usually represented by squares, as "transitions" [71] [72] [73] [74].

4.1. *FMS* Specification by *WPTSPN*

We adopt a conversion manual method of the *WF-Net* (represented by *BPMN*) to *WPTSPN* using the terms adapted by [71] [72] [73] [74].

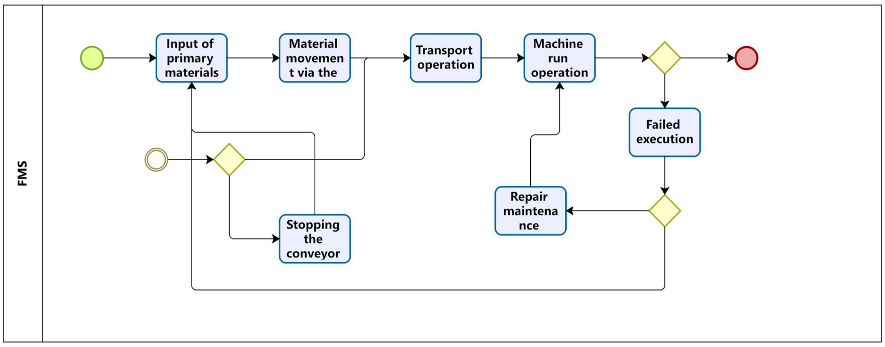


Figure 3. Flexible manufacturing systems modeled based on *BPMN* 2.1 notation.

Table 1. Description of the *FMS*-business process steps.

Operations-Id	Signification
Op_1	Input of primary materials.
Op_2	Material movement via the conveyor belt.
Op_3	Stopping the conveyor belt.
Op_4	Starting the conveyor belt.
Op_5	Transport operation between the pallet and the input buffer of the production machine.
Op_6	Machine run operation.
Op_7	Failed execution.
Op_8	Repair maintenance operation.

- The application field is an *FMS*-Workflow.
- The set of $A=\{A_1, A_2, A_3, A_4, A_5\}$: the actors set representing human resources allocated to a workflow.
- The workflow activities set named AC represents the finished product manufacture via an FMS business process.
- The set $SAC=\{sac_1, sac_2, \cdots, sac_n\}$ the sub-activities to be executed sequentially or in parallel according to a work scheduling strategy on the production multiple lines.
- Places Specification:

The set $$\begin{aligned} P=&\{P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10}\} \\ &\cup\{P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{19}, P_{20}, P_{21}, P_{22}\} \end{aligned}.$$

The model places, without taking into account the intermediate places, are explained by the Table 2.

The $\{P_{11}, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}, P_{17}, P_{18}, P_{19}, P_{20}, P_{21}, P_{22}\}$ places set represents intermediate places set represents the intermediate places (micro-places) when we adopt the splitting approach for non-immediate transitions (like in [7]). Such micro-places' role is to avoid the loss of tokens during the transitions splitting as explained in [7]. In specifying the *FSM*, P_1 is deemed as an initial place and P_{10} is a final place.

- Transitions Specification:

Like in [7], the *WPTSPN* is liable to distinguish between the three transition types (immediate, deterministic and stochastic). Each stochastic or deterministic model transition is divided into three micro-transitions (Tr^c, Tr^w, and Tr^p). The sub-model transition specification and the associated probability values are well described in Table 2.

Each model transition, whether stochastic or deterministic, is divided into

Table 2. Places specification.

Places	*Signification*	${}^{I}P$	P^{O}
P_0	Actor park (execution and maintenance worker).	$\varnothing$	tr_1
P_1	The place of Workflow begins.	tr_6, tr_8	tr_6, tr_8
P_2	A primary material conveyor.	$\varnothing$	tr_1, tr_5
P_3	Unlimited conveyor belt input buffer.	tr_1	tr_2
P_4	A conveyor belt reading place.	tr_3	tr_2, tr_4
P_5	A remote position and conveyor belt element.	tr_2	tr_5
P_6	A waiting place.	tr_2, tr_4	tr_3
P_7	Specifies both a machine bounded input buffer and a conveyor output buffer.	tr_5	tr_6
P_8	Place of the maintenance sub-process start.	tr_6, tr_9	tr_6, tr_7
P_9	End of the maintenance sub-process.	tr_7	tr_8
P_{10}	The end of the workflow.	tr_6	$\varnothing$

three micro-transitions (Tr^c, Tr^w, and Tr^p). **Table 3** and **Table 4** contain a detailed description of the sub-model transition specification as well as the corresponding probability values. **Table 3** describes the transition Tr_6 (which corresponds to Op_6 in **Table 1**) as follows: ("Machine run operation").

- Tokens Specification:

Three different token types are adopted by our *WPTSPN*, which is an *FMS* business process representation:

- A firing sequence triggered by the presence of a *SAC-type* token named sac_1 specifies the finished product cycle, from its primary form to its packaging. In other words, the FMS workflow represented by the WPTSPN, started by P_1 is considered as begin place contains sac_1.
- To ensure the execution of the semi-automatic operations of the *FMS* workflow modeled by *WPTSPN*, we represent five types of tokens *A* having the execution agents' role in the production section or the maintenance workflow. Place P_0 is considered the workers' park.
- An encapsulation tokens type comes from the places-entries of a non-immediate transition. This new token role guaranteed the non-loss of the tokens-entries during the division and the sequential micro-transitions firing.

4.2. *FMS* Model Represented by WPTSPN

In this subsection, with the development of the model *WPTSPN* representative

Table 3. The business process *FMS* transitions specification.

Tr	Op	Type	Rate	${}^I tr$	tr^O
Tr_1	1	Stochastic	φ_1	P_1	P_3, P_2
Tr_2	2	Stochastic	φ_2	P_3, P_4	P_5, P_6
Tr_3	3	Immediate		P_6	P_4
Tr_4	4	Immediate		P_4	P_6
Tr_5	5	Deterministic		P_2, P_5	P_7
Tr_6	6	Stochastic	φ_3	P_0, P_7, P_8	P_{10}, P_0, P_8
Tr_7	7	Stochastic	φ_4	P_8	P_9
Tr_8	8	Stochastic	φ_5	P_0, P_9	P_0, P_8

Table 4. Rates met description.

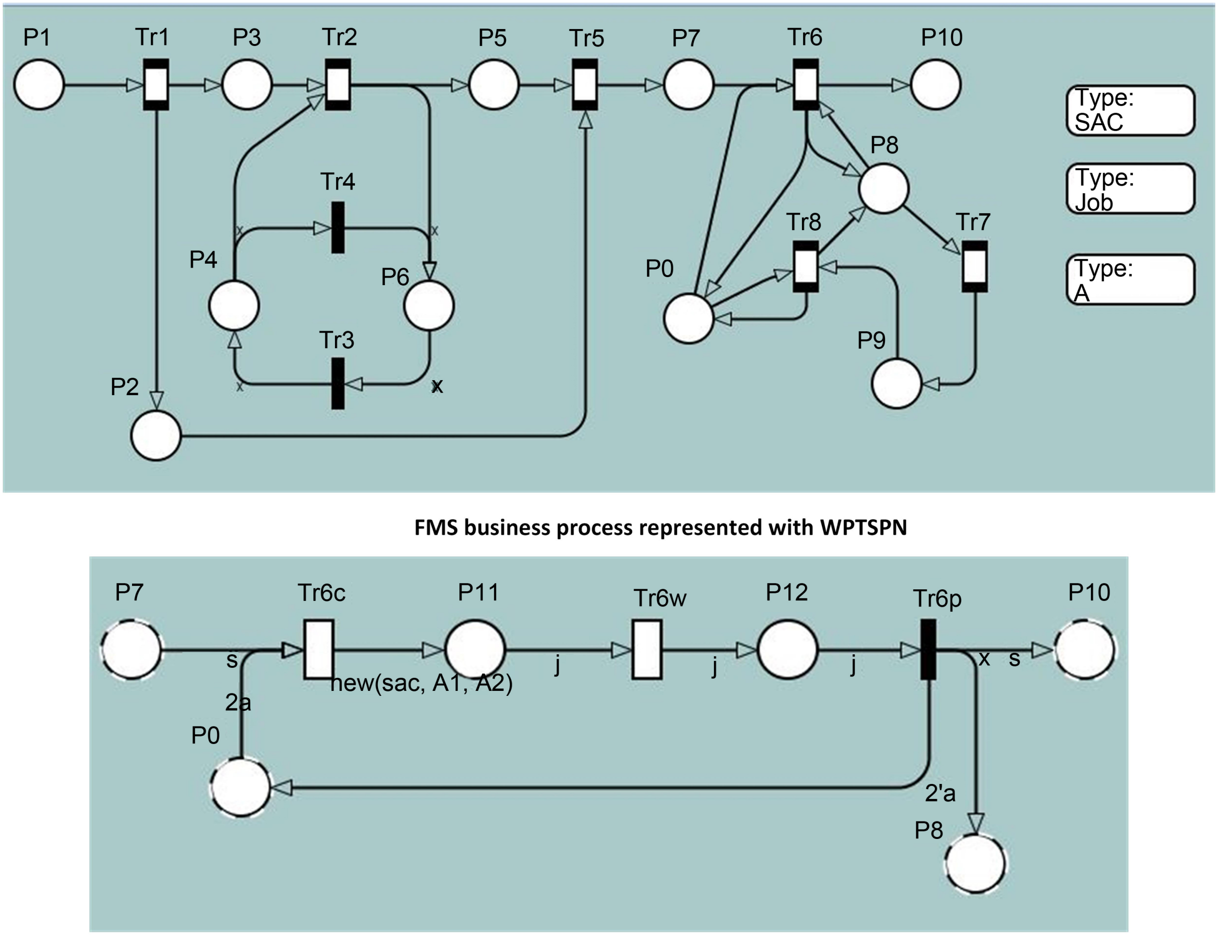
Rate	Mets signification
φ_1	The raw material capture success rate
φ_2	Success rate of material movement operation
φ_3	Machine runtime service success rate
φ_4	Execution failure rate
φ_5	Repair rate maintenance operation

of the *FMS* business process. The specification phase is completed by this step which summarizes the previous subsection steps.

From the *FMS* workflow specified by our *WPTSPN* model representing an extension of model in [7], we consider an abstraction with two levels:

- The first level presents the graphical *PN* representation via places, tokens randomly residence (stay), transitions, arc, tokens, and firing pre-conditions (see **Figure 4(a)**).
- According to a micro-level life cycle, the second level explains the micro places, micro-transitions, and their firing pre-conditions (tokens consumption phase, execution phase, and tokens production phase) (liken in [7]).

To illustrate the micro-places and the micro-transitions, we present the abstraction-second level applied to the transition Tr_6 in **Figure 4(b)**. The Cardinality Marking Graph (**Figure 5**), State Graph based (**Figure 6**) on the timed places, Randomly residence (stay) time tokens, and tokens' life-cycle shown earlier are helpful tools to interpret and verify the P-timed properties following a transitions firing sequence.



FMS business process represented with WPTSPN

Tr6 splitted into tree micro-transitions

Figure 4. *FMS* business process represented with *WPTSPN* and Tr_6 splitted into tree micro-transitions.

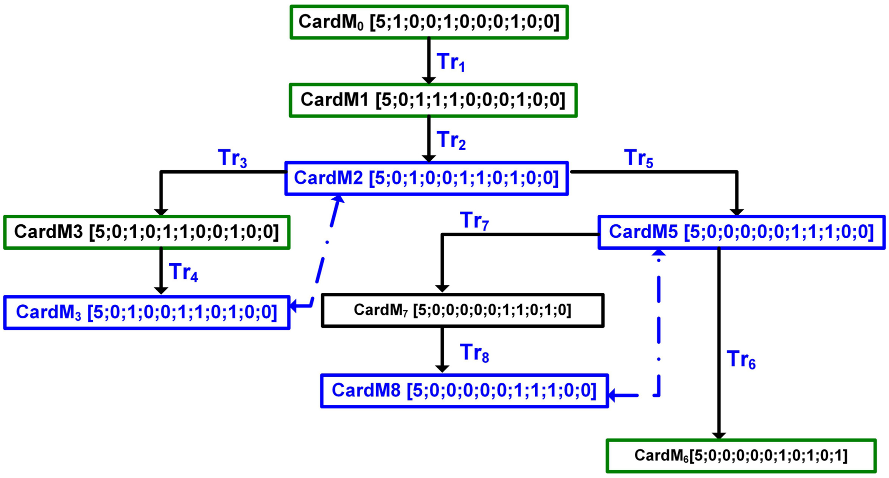

CardM₀ [5;1;0;0;1;0;0;0;1;0;0]
Tr₁
CardM1 [5;0;1;1;1;0;0;0;1;0;0]
Tr₂
Tr₃
CardM2 [5;0;1;0;0;1;1;0;1;0;0]
Tr₅
CardM3 [5;0;1;0;1;1;0;0;1;0;0]
Tr₇
CardM5 [5;0;0;0;0;0;1;1;1;0;0]
Tr₄
CardM₃ [5;0;1;0;0;1;1;0;1;0;0]
CardM₇ [5;0;0;0;0;0;1;1;0;1;0]
Tr₈
Tr₆
CardM8 [5;0;0;0;0;0;1;1;1;0;0]
CardM₆[5;0;0;0;0;0;1;0;1;0;1]

Figure 5. *FMS* represented by *WPTSPN* model cardinality marking graph.

4.3. FMS Model Checked Based on WPTSPN

The rules for conversion from a markings cardinalities graph *CdM* to a state graph *G-SE* are:

- each *CdM* is transformed into a state *e*;
- in case cdM_i comes from a cdM_{i-1} by firing an immediate transition of the *WPTSPN*, then a new state will not be considered and we associate a reversible arc in state e_{i-1}.

Table 5 illustrates the correspondence between *SE* states and *cdM* markings and their generating transitions.

According to the markings cardinalities graph and the states graph associated with the *WPTSPN*, we notice:

- According to the workflow semantics of the FMS system modeled by WPTSPN, the arrival of a sac_1 type token in the place considered initial. Then the state e_0 and cdM_0 are interpreted as the start of the *FMS* management process.
- The state e_4 and cdM_6 are interpreted as the end of the *FMS* workflow.
- According to the hypothesis of state stability during the firing of an immediate transition [63], we observe that two tr_3 and tr_4 generate a state of temporal stability despite the change in the marking cardinality based on ordinary circulation tokens.
- The firing of the tr_7 transition generates, according to the semantics of the workflow [6], a change in the state of the sub-activity sac_1 from in-execution to suspended.
- Each firing sequence comprising the two immediate transitions tr_7 and tr_8 generate a return to the (cdM_5, e_2), so it is a firing circuit.
- The study of markings cardinalities graph allows us to notice that it is possible to fire three transition sequences: $s_1 = \{tr_1, tr_2, tr_3, tr_4, tr_5, tr_7, tr_8, tr_5, tr_6\}$; $s_2 = \{tr_1, tr_2, tr_5, tr_7, tr_8, tr_5, tr_6\}$; $s_3 = \{tr_1, tr_2, tr_5, tr_6\}$. The s_3 firing sequence ends with a transition providing the finished product (the sac_1 token instead

Table 5. Markings cardinalities, states, and generating transitions in *FMS* management process.

Steps (*CdM*)	States	Transition
$cdM_0\left[5,1,0,0,1,0,0,0,1,0,0\right]$	e_0	-
$cdM_1\left[5,0,1,1,1,0,0,0,1,0,0\right]$	e_1	tr_1
$cdM_2\left[5,0,1,0,0,1,1,0,1,0,0\right]$	e_2	tr_2
$cdM_3\left[5,0,1,0,1,1,0,0,1,0,0\right]$	e_2	tr_3
$cdM_4\left[5,0,1,0,0,1,1,0,1,0,0\right]$	e_2	tr_4
$cdM_5\left[5,0,0,0,0,0,1,1,1,0,0\right]$	e_3	tr_5
$cdM_6\left[5,0,0,0,0,0,1,0,1,0,1\right]$	e_4	tr_6
$cdM_7\left[5,0,0,0,0,0,1,1,0,1,0\right]$	e_5	tr_7
$cdM_8\left[5,0,0,0,0,0,1,1,1,0,0\right]$	$e_6 \rightleftharpoons e_3$	tr_8

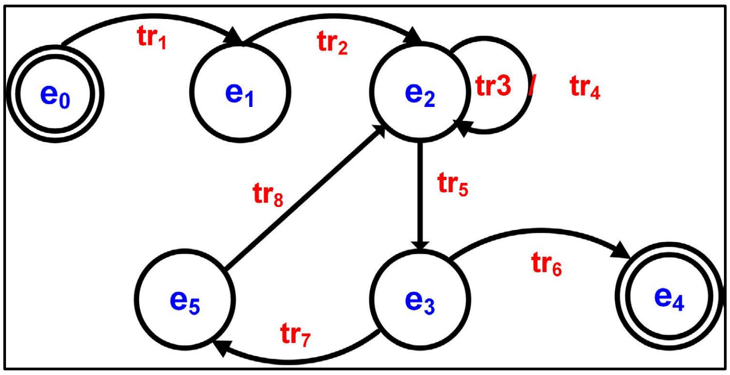


Figure 6. *FMS* business process represented by States Graph.

P_{10}), which contains no transition circuit, so its firing is assumed to be purely sequential.

The model's properties are checked according to verification semantics based on *T-timed*:

- The *WPTSPN* associated with the *FMS* workflow for the firing sequence transition s_3 is 9-*bounded*.
- The *WPTSPN* is not pure.
- The *WPTSPN* has no source transition.
- The transitions are *quasi-live* so the *WPTSPN* model is almost alive.
- Note that the *WPTSPN* has an accessibility problem when it reaches cdM_6.
- We see that cdM_5 and cdM_8 are equivalent to what a firing circuit shows.
- The marking cardinality cdM_5 has no successor. So it's a deadlock.
- The *WPTSPN* is *quasi-live*, non-conservativeness, not proper, not balanced, and not sound.

The model properties specificity verifying based on the *P-timed* semantics and the tokens cycle life carried following a behavioral simulation of the *FMS* business process. So, the following subsection is dedicated to simulating the tokens' randomly timed behavior in the places at the *WPTSPN* sequence s_3.

4.4. *FMS* Model Simulation

A cardinality marking graph/state graph interpretation verifies that cdM_6/e_4 is the final state of the *FMS* business process. We select the transitions in three sequences $s_1 = \{tr_1, tr_2, tr_3, tr_4, tr_5, tr_7, tr_8, tr_5, tr_6\}$, $s_2 = \{tr_1, tr_2, tr_5, tr_7, tr_8, tr_5, tr_6\}$, and $s_3 = \{tr_1, tr_2, tr_5, tr_6\}$ lead to the final state.

The s_3 sequence is preferred to be simulated since it does not contain circuits. In our firing and randomly residence time analysis, we try to apply the method adopted by the *WPTSPN*.

- Let $\{e_0, e_1, e_2, e_3, e_4\}$: the states' firing sequence corresponding to s_3.
- Let $\{P_0, P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_{10}\}$: the places' subset is involved in the s_3 firing sequence.
- Let MIP_3: the matrix of the places input-transitions involved in the transitions sequence s_3.
- Let MPO_3: the matrix of the places output-transitions involved in the transitions sequence s_3.

$$MIP_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad MPO_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Let MIT_3: the matrix of the transitions input-places involved in the transitions sequence s_3. Let MTO_3: the matrix of the transitions output-places involved in the transitions sequence s_3.

$$MIT_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$MTO_3 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Thus, the simulated system evolution depends on different situations according to the densities types (Exponential, Normal, and Log-Normal). To calculate the firing time of each transition as well as the tokens' random stay in each timed place, we target the transitions firing sequence S_3. A data collection specifying the parameters of our simulation solution must be provided as an input {*places*, MIP_3, MPO_3}, {*transitions*, MIT_3, MTO_3, *times at the latest*, *and firing rates*}, and the parameters related to the probability distribution type ($\lambda = 0.6$, $\mu = 0.6$ and $\sigma = 0.7$). **Table 6** illustrates our simulation transitions basic data.

After retrieving the input data, our simulation process follows the following steps:

- Step 1: The gravity center method [48] [50] which is used by the $TFiringT\left(tr_j\right)$ ($\forall tr_j \in s_3$) function, is used to determine the firing time for each transition in the firing sequence (s_3/se_3). This function returns three-time values, one for each density type (exponential, normal, and log-normal). The outcomes of this stage are presented in **Table 7**: (sorting time with the gravity center method according to the three types of distribution and their cumulative values).

We note:

- *EGFT*: Exponential Gravity Center Transition Firing Time;
- *NGFT*: Normal Gravity Center Transition Firing Time;
- *L-NGFT*: Log-Normal Gravity Center Transition Firing Time;
- *CEGFT*: Cumulative Exponential Center Transition Firing Time;
- *CNGFT*: Cumulative Normal Center Transition Firing Time;
- *CL-NGFT*: Cumulative Log-Normal Gravity Center Transition Firing Time.

- Step 2: In this simulation phase, we calculate the cumulative firing time ($SFT = \sum\left(TFiringT\left(tr_j\right)\right)$) for each transition tr_j in its firing sequence order (s_3/se_3). Also, for each transition, the three cumulative firing time values (see (**Table 7**)) are the basis for determining token dwell times. Then, for each timed place applied to the sequence, the lower bounds of the static stay intervals are determined from the matrix MIP_3. For each density type, the $MEFIP_3$, $MNFIP_3$, and $ML\text{-}NFIP_3$ matrix are constructed to select each

Table 6. Simulation transitions basic input data.

Tr	Type	Rate value	Time at the latest (expressed in minutes)
Tr_1	Stochastic	0.6	10
Tr_2	Stochastic	0.7	15
Tr_5	Deterministic		20
Tr_6	Stochastic	0.5	25

Table 7. Each value of transitions gravity center firing time is expressed in minutes.

Transition	*EGFT*	*NGFT*	*L-NGFT*	*CEGFT*	*CNGFT*	*CL-NGFT*
tr_1	1.12	0.79	4.99	1.12	0.79	4.99
tr_2	3.89	2.37	9.99	5.02	3.17	14.98
tr_5	11.69	0.79	4.99	16.71	18.17	29.98
tr_6	35.08	37.13	37.48	51.80	55.30	67.47

min-value row corresponding to the lower sac-token boundary at the current timed place (random residence).

$$MEFIP_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 51.80 \\ 0 & 0 & 0 & 0 \\ 1.12 & 0 & 0 & 0 \\ 0 & 0 & 16.71 & 0 \\ 0 & 5.02 & 0 & 0 \\ 0 & 5.02 & 0 & 0 \\ 0 & 0 & 16.72 & 0 \\ 0 & 0 & 0 & 51.80 \\ 0 & 0 & 0 & 51.80 \end{pmatrix}$$

$$MNFIP_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 55.30 \\ 0 & 0 & 0 & 0 \\ 0.79 & 0 & 0 & 0 \\ 0 & 0 & 18.17 & 0 \\ 0 & 3.17 & 0 & 0 \\ 0 & 3.17 & 0 & 0 \\ 0 & 0 & 18.17 & 0 \\ 0 & 0 & 0 & 55.30 \\ 0 & 0 & 0 & 55.30 \end{pmatrix}$$

$$ML\text{-}NFIP_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 67.47 \\ 0 & 0 & 0 & 0 \\ 4.99 & 0 & 0 & 0 \\ 0 & 0 & 29.98 & 0 \\ 0 & 14.98 & 0 & 0 \\ 0 & 14.98 & 0 & 0 \\ 0 & 0 & 29.98 & 0 \\ 0 & 0 & 0 & 67.47 \\ 0 & 0 & 0 & 67.47 \end{pmatrix}$$

- Step 3: In this third phase, we fix a random residence interval upper limit, which is assigned to a token in each timed place occurring in the sequence s s_3/se_3. For each density type, the $MEFIT_3$, $MNFIT_3$, and $ML\text{-}NFIT_3$ matrices are elaborated to select the minimum value of each column corresponding to the upper limit of the token's stay.

$$MEIT_3 = \begin{pmatrix} 0 & 1.12 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5.02 & 5.02 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 16.71 & 0 & 0 & 16.71 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 51.80 & 51.80 & 0 \end{pmatrix}$$

$$MNIT_3 = \begin{pmatrix} 0 & 0.79 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.07 & 3.17 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18.17 & 0 & 0 & 18.17 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 55.30 & 55.30 & 0 \end{pmatrix}$$

$$MNLIT_3 = \begin{pmatrix} 0 & 4.99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 14.99 & 14.99 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 29.98 & 0 & 0 & 29.98 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 67.47 & 67.47 & 0 \end{pmatrix}$$

After determining the randomly residence limits for each sac-token in *WPTSPN*' timed place, we clarify that stochastically using the density functions according to the three distributions. Figure 7 represents the sac-token' randomly residence distributions in the timed place P_7.

- The last step is summed up in determining for each place its three means of the stay time according to the gravity center method using the function $MST(P_i)$, $(\forall P_i \in s_3)$.

Table 8 illustrates the three randomly residence of all sac-token in the place P_7, its average residence times according to the three densities (exponential, Normal, and Log-normal), and a token sac_1 cycle-life according to the time horizon.

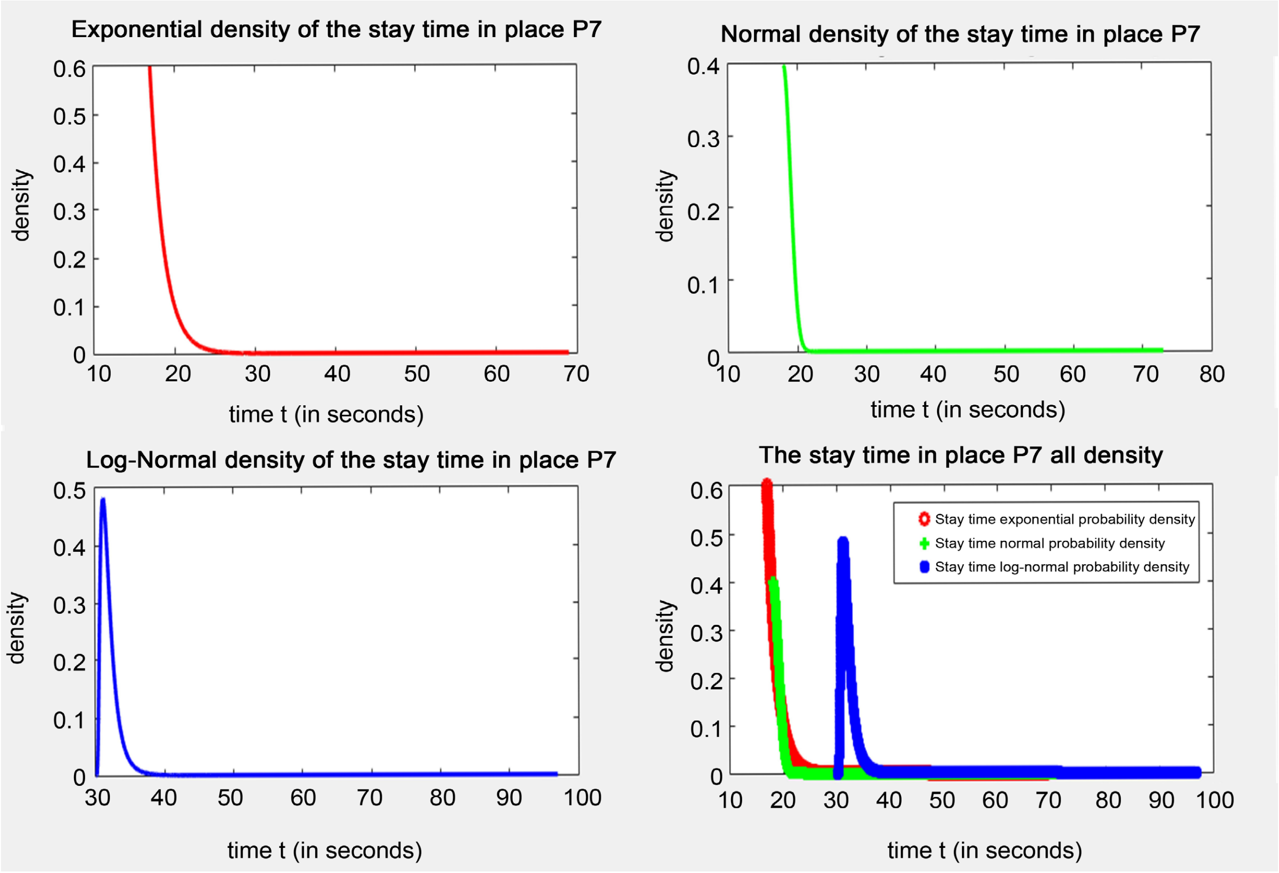


Figure 7. Token random residence time in the place P_7 according to the three densities.

Figure 8 specifies the process of determining the stay dynamic interval according to the place P_7 and the decision about the token's life-cycle, according to its time sensor value. The token sac_1 arriving in the place P_7 following the transition tr_5 firing is equipped with an automatic time sensor. This time sensor can activate a perception operation of the Randomly residence and compare its value, triggered from the first moment in the place, with the $MST(P_7)$.

The following subsection shows our delay detection and prediction process related to the randomly residence time in a firing sequence.

4.5. *FMS* Business Process Prediction Delay

The verification and simulation, allow us to detect a delay problem of the execution time relating to the operation op6 of the FMS business process, knowing that the latter is specified by the transition tr_6. So, we determine the cumulative

Table 8. Three randomly residence intervals of the token sac_1 and their life cycle in the place P_7.

Density type	Lower limits	Upper limit	MST	$\tau < LL$	$\tau \in [MST, UL]$	$\tau > UL$
Exp	16.71	51.80	18.65	not available	available with risk of delay	dead
N	18.17	55.30	18.79	not available	available with risk of delay	dead
L_N	29.98	67.47	33.43	not available	available with risk of delay	dead

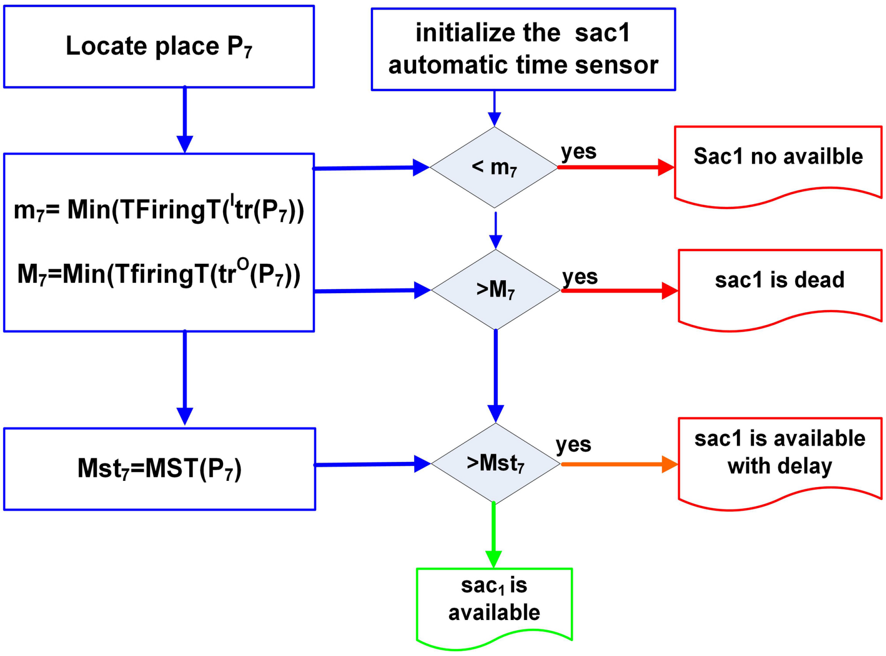

Locate place P_7
initialize the sac1 automatic time sensor
m_7= Min(TFiringT(^{I}tr(P_7))
M_7=Min(TfiringT(tr^O(P_7))
Mst_7=MST(P_7)
< m_7
yes
Sac1 no availble
>M_7
yes
sac1 is dead
>Mst_7
yes
sac1 is available with delay
sac_1 is available

Figure 8. Process of determining the stay dynamic interval and the decision about the token's life-cycle.

transition sequence firing time (s_3/se_3), after that, we check this delay indicated in the **Figure 9**. This figure illustrates the cumulative firing time, according to the three densities, for each transition making up the sequence. The delay detection step launches the prediction process linked to the tokens random residence in timed places. We apply the same prediction process to the micro-places, which are obtained following the division mechanism adopted by the model *WPTSPN*).

The model prediction approach requires converting the cardinality markings graph and/or the state sequence graph to *Event-log*: *Lev*, matching each *cdM*/*se* in Event. According to our *FMS* workflow simulation and verification results, firing sequence transitions denoted s_3/se_3 create a delay risk in FMS execution. Therefore, we propose a delay prediction procedure based on the algorithmic aspect illustrated in **Figure 8**. Knowing that the delay causes the token death in timed places during the transitions firing sequence. So, we use a prediction process based on a density truncation with the three possible distributions. This last process is illustrated by **Algorithm 1**.

```
Algorithm 1 GTruncated
  $t_m \leftarrow \min(MFIP(id_p,\ :))$
  $t_M \leftarrow \min(MFPO(id_p,\ :))$
  $t_c \leftarrow MST(id_p,\ t_m,\ t_M)$
  $t \leftarrow [t_m : 0.016 : t_M]$
  $/*\ times\ interval\ graduated\ in\ seconds\ */$
  $i \leftarrow 1$
  while $i \leq tength(t)$ do
    if $t[i] < t_m$ then
      $dtp[i] \leftarrow Gdistrib(t_m)$
    end if
    if $t[i] \geq t_c$ then
      $dtp[i] \leftarrow \frac{Gdensity(t[i])}{1-Gdistrib(t_c)}$
    else
      $dtp[i] \leftarrow GDensity_{\triangle}(t[i]-t_c)$
    end if
  end while
  return $dtp$
```

An illustrative example of the truncated randomly residence densities is applied to sac-token in P_7 and shown in **Figure 10**. According to our model of transitions division approach (like in [7]), it is possible to apply the truncating process at the second level (micro-places), as illustrated in **Figure 11** and **Figure 12**.

Knowing that the two places (P_{17}, P_{18}) are considered as two micro-places preventing the splitting mechanism of the tr_6 transition. As our model based on the no interpretation hypothesis of immediate transitions, then tr_6^P does not temporally generate interpretable micro-states.

So, the immediate micro-transition Tr_6^P densities' truncation is not performed (like in [7] [48]). The delay margin calculating the operation is based on subtraction between the time corresponding to the gravity center stay of the

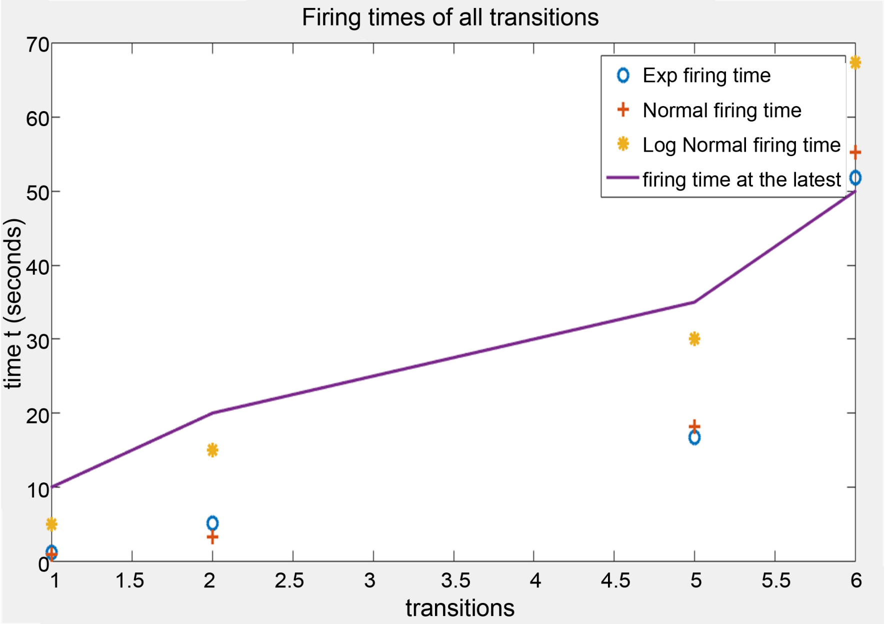


Figure 9. The firing time of all transitions in s_3/se_3 according to all densities adopted by the *WPTSPN* model.

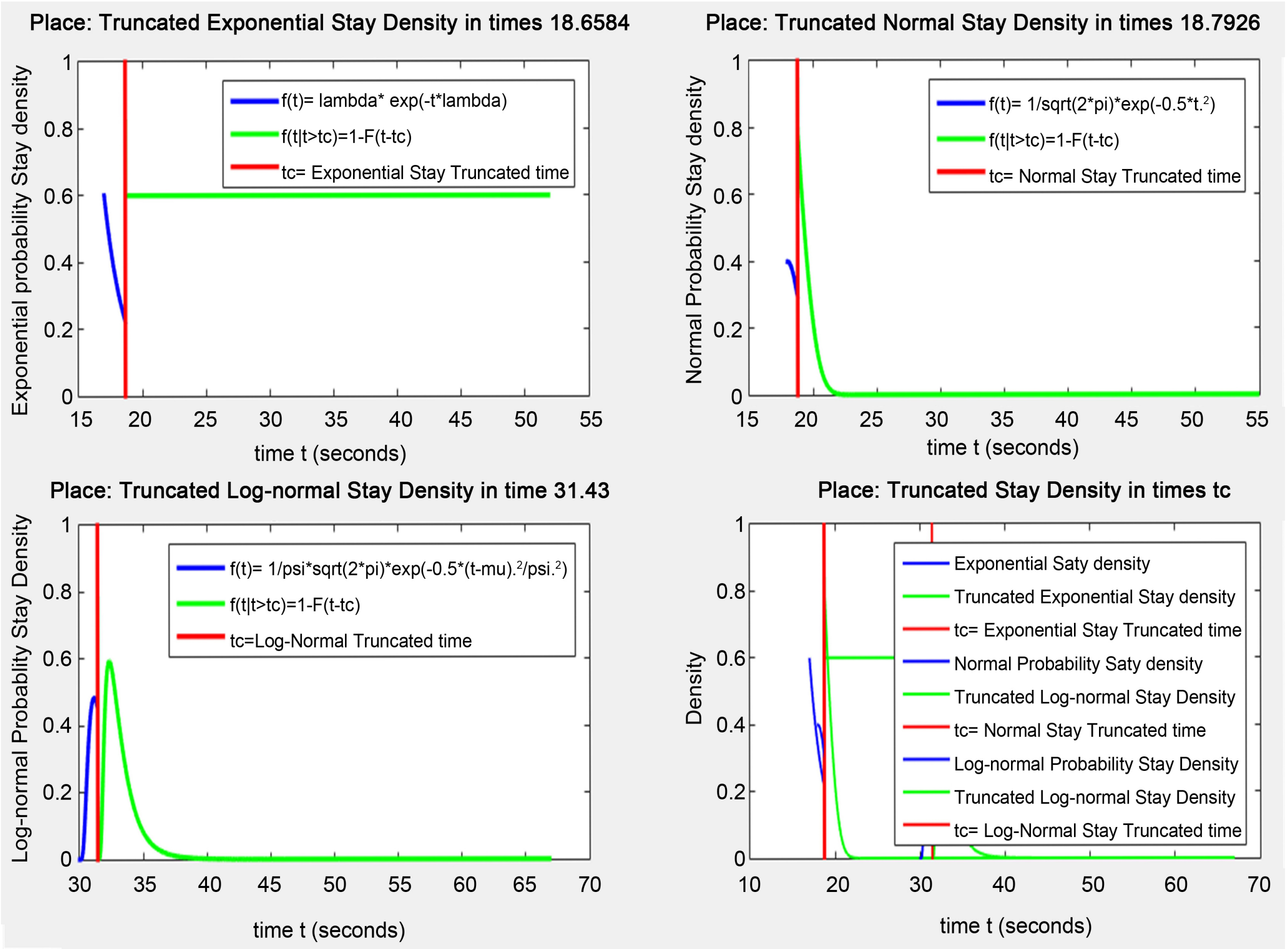


Figure 10. The firing times of all transitions in s_3/se_3 according to all truncated densities adopted by the *WPTSPN* model.

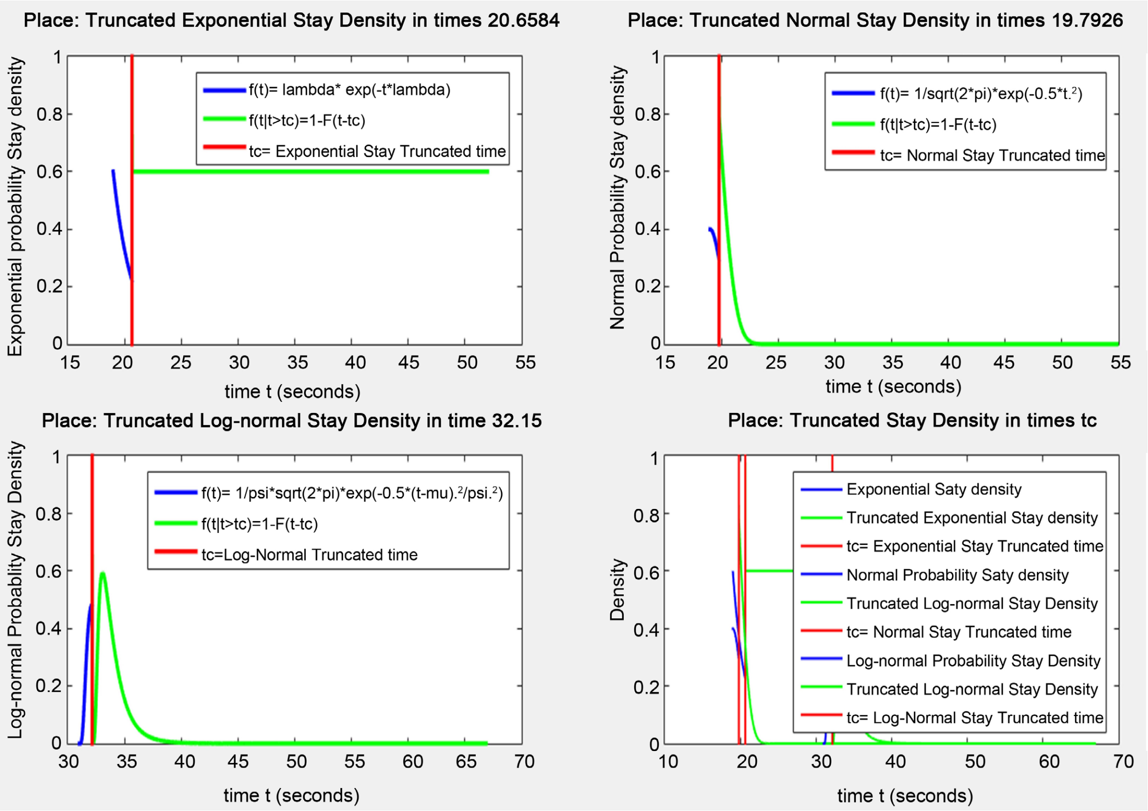


Figure 11. All the randomly residence densities in the micro-place P_{17} truncated.

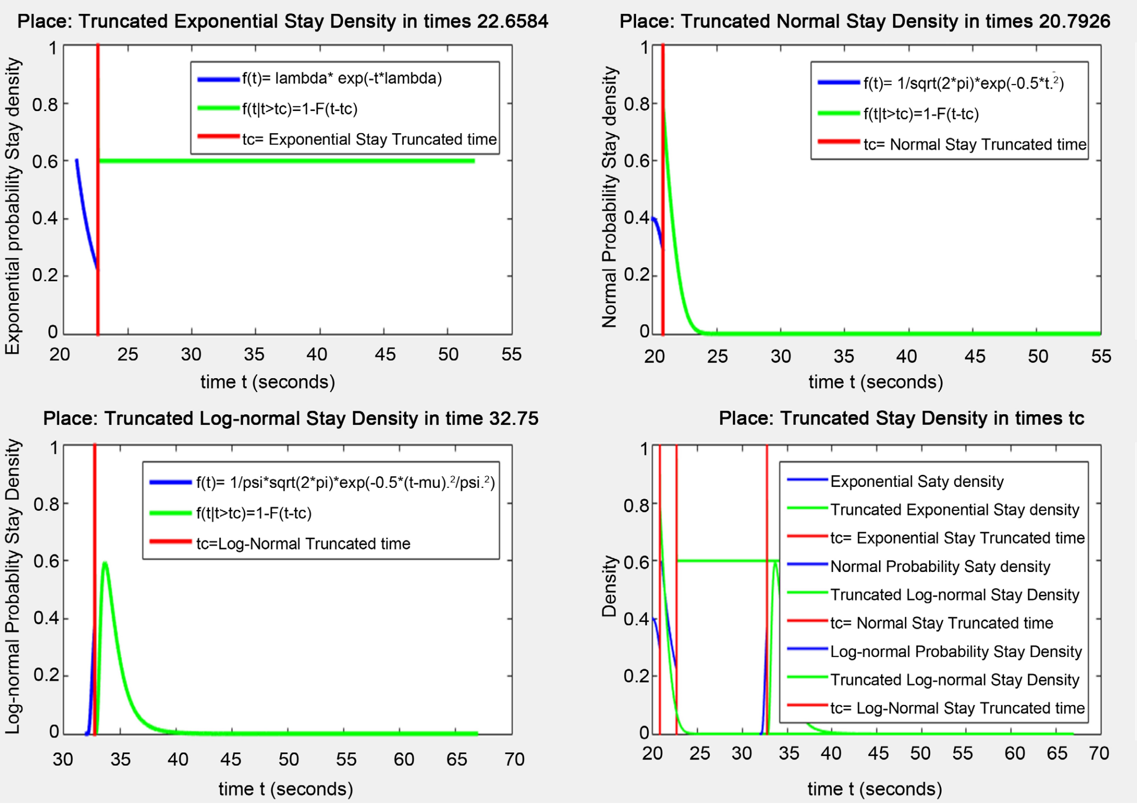


Figure 12. All the randomly residence densities in the micro-place P_{18} truncated.

truncated density and the truncation time. After that, a statistical set is generated by observing the truncation operations on the densities. The empirical mean and empirical variance are determined for each density, with the confidence interval forming the aim around the truncation times delayed estimations. Since the statistical population number is reduced, we adopt the empirical estimate method in [7] and [75].

Table 9 illustrates the prediction results based on our truncation process applied to P_7, P_{17} and P_{18} micro-places. Thus, we determine the delay margins for truncated density. To explore the numerical values meaning in **Table 9**, we note:

- *LST*: Lower Limit of Stay Interval;
- *TT*: Truncated Time or Gravity Center Stay Time;
- *EM*: Empirical Mean of Truncated Densities;
- *EMS*: Empirical Margin of Stay;
- *EL*: Empirical Likelihood;
- *CM*: Empirical Confidence Margin.

The temporal analysis of tokens' random residence in timed places adopted by our *WPTSPN*, shows that if we use an exponential density, then the stay-delay tends towards stability. So, **Figure 10** shows a uniformly distributed truncated stay density on the time horizon. According to **Table 9**, we notice that the stay margin distributed with the log-normal density gratitude is compared to that distributed with the normal one. Also, in the *FMS* business process evolution, the tr_6 transition favors the end of production because of a maintenance subprocess. The proposed model applies the prediction tool to the micro-places which result from the transitions division mechanism. We note that the prediction process, oriented towards the P-timed context, requires retrospection on the past input transitions firing time, and a future vision (estimation) of the transitions firing time output.

Table 9. Three randomly time residence intervals in the places P_7, P_{17}, and P_{18} truncated.

Places	Densitytype	*LST*	*TT*	*EM*	*EMS*	10^2**EL*	10^2**CM*
P_7	*Exp*	17	18.65	0.60	±1.65	0.01	±0.04
	N	18	18.79	0.01	±0.79	0.88	±0.33
	L_N	30	31.43	0.03	±1.43	1.16	±2.23
P_{17}	*Exp*	19	20.65	0.60	±1.65	0.01	±0.04
	N	19	19.79	0.02	±0.79	0.09	±0.40
	L_N	31	32.15	0.03	±1.15	1.18	±2.51
P_{18}	*Exp*	21	22.65	0.60	±1.65	0.01	±0.04
	N	20	20.79	0.02	±0.79	0.093	±0.40
	L_N	32	32.75	0.03	±0.75	1.20	±2.7

5. Conclusions

In this article, we propose an extension of stochastic Petri nets [63] named *WPTSPN* and which combined with *Workflow-Nets* [6]. This extension endowedwith an orientation *P-Timed* [30] [58] with (sojoun) residence times of the tokens determined randomly using the centers of gravity assigned to the density functions [31]. We apply their specification and verification approach in the context of the FMS Workflow [76]. In addition, the random residence time of tokens in timed places and the mechanism for triggering transitions are determined not only by the stochastic and deterministic aspects but also by other conditions adapted to the FMS context. The micro-steps are formally represented respectively by micro-places (micro-transitions [7]) and sequentially concern the consumption, the execution of the task (sub-activity), and the production of tokens. The latter allows us to distinguish and locate, in a precisemanner, any type of execution failure. Our verification process adopted by the *WPTSPN* can bejustified by a state space based on the state graph, which represents firing sequences to control and localize shared resource conflicts.

We note that this model is enriched by formal tools based on the elapsed time to predict the remaining time and estimate the residence time (residence) of the tokens in the timed places during a firing sequence of a state. The residence time (stay) of the tokens in each model timed place, representative of an FMS, also indicates the expiration time of the primary or semi-finished material.

On the one hand, in our future work, we want to expand the expressiveness of the *WPTSPN* by incorporating the sharing of resource restrictions [76], the interdependence between the time of activities [77], and the performance evaluation of synchronous operations [30] [31]. This may be accomplished utilizing model transformation approaches, fluid Stochastic Petri Net models [26], or *RecatNets* models [78], which can be used to drive the tool to automatically change the development of the *BPMN* model into a *WPTSPN* like in [74] and [71].

The *WPTSPN*, on the other hand, can be enriched with formal tools to cover a dynamic workflow context, ensuring the dynamic display of real-time performance and checking the dynamic workflow criteria to effectively assess the system's strengths and weaknesses according to personalized criteria.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Alcacer, V. and Cruz-Machado, V. (2019) Scanning the Industry 4.0: A Literature Review on Technologies for Manufacturing Systems. *Engineering Science and Technology, an International Journal*, **22**, 899-919. https://doi.org/10.1016/j.jestch.2019.01.006

[2] (2020) CGP Gestion de Production.

https://www.cours-gratuit.com/cours-gestion/cours-sur-les-outils-de-la-gestion-de-production

[3] Courtois, A., Martin-Bonnefous, C. and Pillet, M. (2003) Gestion de production. 4 eme Edition, Editions d'Organization.

[4] Ballarini, P., Barbot, B., Duflot, M., Haddad, S. and Pekergin, N. (2015) HASL: A New Approach for Performance Evaluation and Model Checking from Concepts to Experimentation. *Performance Evaluation*, **90**, 53-77. https://doi.org/10.1016/j.peva.2015.04.003

[5] Weske, M. (2019) Business Process Management: Concepts, Languages, Architectures. Springer, Berlin. https://doi.org/10.1007/978-3-662-59432-2_1

[6] Van der Aalst, W.M.P. and Van Hee, K. (2002) Workflow Management: Models, Methods, and System. The MIL Press, Cambridge.

[7] Ben Mesmia, W., Escheikh, M. and Barkaoui, K. (2020) DevOps Workflow Verification and Duration Prediction Using Non-Markovian Stochastic Petri Nets. *Journal of Software*: *Evolution and Process*, **33**, e2329. https://doi.org/10.1002/smr.2329

[8] Sbai, Z. and Barkaoui, K. (2012) Verification Formelle des Processus Workflow Collaboratifs. *Conference francophone sur les Systemes Collaboratifs*, (*SysCo*'12), **5**, 197-200.

[9] D'Ambrogio, A. and Zacharewicz, G. (2016) Resource-Based Modeling and Simulation of Business Processes. *Proceedings of the* 2016 *Summer Computer Simulation Conference* (*SCSC* 2016), Montreal, 24-27 July 2016, page.

[10] (2020) CEDP Chain Event-Driven Process. https://www.wu.ac.at/erp/webtrainer/epc-webtrainer/theory

[11] (2020) Object Management Group Business Process Model and Notation. https://www.omg.org/spec/BPMN/2.0

[12] (2020) BPMN 1.0 Business Process Model and Notation (BPMN)—Version 1.0 2020 URL. https://www.omg.org/spec/BPMN/1.1/PDF

[13] Holzmann, G.J. (1997) The model checker SPIN. *IEEE Transactions on Software Engineering*, **23**, 279-295. https://doi.org/10.1109/32.588521

[14] (2020) BPMN 2.0 Business Process Modeling Notation. https://www.omg.org/spec/BPMN/2.0/

[15] Kheldoun, A., Barkaoui, K. and Ioualalen, M. (2017) Formal Verification of Complex Business Processes Based on High-Level Petri Nets. *Information Sciences*, **385-386**, 39-54. https://doi.org/10.1016/j.ins.2016.12.044

[16] Ciardo, G. (1995) Discrete-Time Markovian Stochastic Petri Nets. In: Stewart, W.J., Ed., *Computations with Markov Chains*, Springer, Boston, 339-358. https://doi.org/10.1007/978-1-4615-2241-6_20

[17] Bocciarelli, P., D'Ambrogio, A., Giglio, A., Paglia, E. and Gianni, D. (2014) Empowering Business Process Simulation through Automated Model Transformations. 2014 *Proceedings of the Symposium on Theory of Modeling and Simulation—DEVS Integrative M&S Symposium*, Tampa, 13-16 April 2014, 39-45.

[18] Boucheneb, H. and Barkaoui, K. (2012) Parametric Verification of Time Workflow Nets. *Proceeding of the* 24*th International Conference on Software Engineering and Knowledge Engineering SEKEC*, San Francisco, 1-3 July 2012, 352-361.

[19] Ben Mesmia, W., Escheikh, M. and Barkaoui, K. (2019) DevOps Workow Specification Based on Non-Markovian Stochastic Petri Nets with Enhanced Expressiveness. *COSI*' 2019, Tizi-Ouzou, 24-26 June 2019, 2-12.

[20] Yamaguchi, S., Yamaguchi, M. and Tanaka, M. (2008) A Soundness Verification Tool Based on the SPIN Model Checker for Acyclic Workflow Nets. *The* 23*rd International Technical Conference on Circuits*/*Systems*, *Computers and Communications* (*ITC*-*CSCC* 2008), Japan, 6-9 July 2008, 285-288.

[21] Dumas, M., La Rosa, M., Mendling, J. and Reijers, H.A. (2018) Fundamentals of Business Process Management. Springer, Berlin. https://doi.org/10.1007/978-3-662-56509-4

[22] Favrel, J. and Lee, K.H. (1985) Modelling, Analysing, Scheduling and Control of Flexible Manufacturing Systems. In: Falster, P. and Maxumder, R.B., Eds., *Modelling Production Management Systems*, Elsevier, Amsterdam, 279-288.

[23] Cecil, J.A., Srihari, K. and Emerson, C.R. (1992) A Review of Petri-Net Applications in Manufacturing. *The International Journal of Advanced Manufacturing Technology*, **7**, 168-177. https://doi.org/10.1007/BF02601620

[24] Archetti, F. and Sciomachen, A. (1988) Representation, Analysis and Simulation of Manufacturing Systems by Petri-Net Based Models. In: Varaiya, P. and Kurzhanski, A.B., Eds., *Discrete Event Systems*: *Models and Applications*, Springer, Berlin, 162-178.

[25] Cox, D.R. (1995) The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables. *Mathematical Proceedings of the Cambridge Philosophical Society*, **51**, 433-440. https://doi.org/10.1017/S0305004100030437

[26] Trivedi, K.S. and Kulkarn, V.G. (1993) FSPNs: Fluid Stochastic Petri Nets. In: Ajmone Marsan, M., Ed., *ICATPN* 1993: *Application and Theory of Petri Nets* 1993, Springer, Berlin, 24-31. https://doi.org/10.1007/3-540-56863-8_38

[27] Ciardo, G., German, R. and Lindemann, C. (1994) A Characterization of the Stochastic Process Underlying a Stochastic Petri Net. *IEEE Transactions on Software Engineering*, **20**, 506-515. https://doi.org/10.1109/32.297939

[28] Krogh, B.H. and Srecnivas, R.S. (1987) Essenlially Decision Free Petri Nets for Real-Time Resource Allocation. *Proceedings of* 1987 *IEEE International Conference on Robotics and Automation*, Raleigh, 31 March-3 April 1987, 1005-1011.

[29] Colter, S.M., Woodward, A.T., fluck, A.A., Smith, R. and White, D.M. (1986) Extended Petri-Nets: The Language of Factory Automation? *Proceedings of ISATA* 1986: 15*th International Symposium on Advances in Technology Education*, *Croydon*, 1-20.

[30] Khansa, W., Aygalinc, P. and Denat, J.P. (1996) Structural Analysis of P-Time Petri Nets Computational Engineering in Systems Applications. *CESA*'96 *IMACS Multiconference*, *Computational Engineering in Systems Applications*, Lille, 9-12 July 1996, 127-136.

[31] Calvez, S., Aygalinc, P. and Khansa, W. (1997) P-Time Petri Nets for Manufacturing Systems with Staying Times Constraints. *IFAC Proceedings Volumes*, **30**, 1487-1492. https://doi.org/10.1016/S1474-6670(17)43571-3

[32] Capkovic, F. (1995) A Petri Net-Based Approach to Synthesis of Control Systems for Discrete Event Dynamic Systems New Trends in Design of Control Systems 1994. Elsevier, 345-350.

[33] DiLeva, A., Vernadat, F. and Bizier, D. (1987) Information Systems Analysis and Conceptual Database Design on Production Environments with M*. *Computers in Industry*, **9**, 183-217. https://doi.org/10.1016/0166-3615(87)90037-6

[34] Bruno, G. and Biglia, P. (1985) Performance Evaluation and Validation of Tool Handling in Flexible Manufacturing Systems Using Petri Nets. *Proceedings of International Workshop on Timed Petri*-*Nets*, Turin, July 1985, 64-71.

[35] Balbo, G., Chiola, G., Franceschinis, G. and Roet, G.M. (1987) Generalised Stochastic Petri Nets for the Performance Evaluation of FMS. *Proceedings of IEEE International Conference on Robotics and Automation*, Raleigh, 31 March-3 April 1987, 1013-1018.

[36] Narahari, Y. and Viswanadham, N. (1987) Performance Evaluation Using a Class of Petri-Nets with Deterministic and Exponential Firing Times. *TENCON* 87, 1987 *IEEE Region* 10 *Conference*: "*Computers and Communications Technology toward* 2000", Seoul, 25-28 August 1987, 482-486.

[37] Wadbwa, S. and Thrownc, J. (1988) Analysis of Collision Avoidance in Multirobot Cells Using Petri Nets. *Robotersysteme*, **4**, 107-115. https://dblp.org/rec/journals/robotersysteme/WadhwaB88.bib

[38] Martincz, J. and Silva, M. (1984) A Language for the Description of Concurrent System Modeled by Colored Petri Nets: Application to the Control of Flexible Manufacturing Systems. In: Chang, S.K., Ed., *Languages for Automation*, Springer, Boston, 369-388. https://doi.org/10.1007/978-1-4757-1388-6_18

[39] Castelain, E., Corbel, D. and Gentina, J.C. (1985) Comparative Simulations of Control Processes Described by Petri-Nets. *Proceedings of COMPINT* 1985: *Computer Aided Technologies*, Quebec, 8-12 September 1985, 215-226.

[40] Ammour, R., Leclercq, E., Sanlaville, E. and Lefebvre, D. (2017) Faults Prognosis Using Partially Observed Stochastic Petri Nets: An Incremental Approach. Discrete Event Dynamic Systems, **28**, 247-267. https://doi.org/10.1109/WODES.2016.7497890

[41] Ajmone Marsan, M., Balbo, G. and Conte, G. (1984) A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Transaction Computer Systems*, **2**, 93-122. https://doi.org/10.1145/190.191

[42] Choi, I., Song, M., Park, C. and Park, N. (2003) An XML-Based Process Definition Language for Integrated Process Management. *Computers in Industry*, **50**, 85-102. https://doi.org/10.1016/S0166-3615(02)00139-2

[43] Lefebvre, D. (2017) Detection of Temporal Anomalies for Partially Observed Timed PNs. *Mathematical Problems in Engineering*, **2017**, Article ID: 2821078. https://doi.org/10.1155/2017/2821078

[44] Lefebvre, D. (2017) Probability of Faults for Partially Observed Timed PNs with Temporal Constraints. 2017 *IEEE* 14*th International Conference on Networking, Sensing and Control* (*ICNSC*), Calabria, 16-18 May 2017, 103-108. https://doi.org/10.1109/ICNSC.2017.8000075

[45] Berthomieu, B. and Diaz, M. (1991) Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Transactions on Software Engineering*, **17**, 259-273. https://doi.org/10.1109/32.75415

[46] Choi, H. (1993) Performance and Reliability Modeling Using Markov Regenerative Stochastic Petri Nets. Ph.D. Thesis, Graduate School of Duke University, Durham.

[47] Ciardo, G. and Lindemann, C. (1993) Analysis of Deterministic and Stochastic Petri Nets. *Proceedings of* 5*th International Workshop on Petri Nets and Performance Models*, Toulouse, 19-22 October 1993, 160-169.

[48] Rachidi, S. (2019) Diagnostic des défauts dans les systèmes à évènements discrets soumis à des contraintes temporelles. Automatique/Robotique. Thèse de Doctorat, Normandie Université, Normandie, 26-30. https://tel.archives-ouvertes.fr/tel-02441760

[49] Lefebvre, D., Rachidi, S., Leclercq, E. and Pigné, Y. (2018) Diagnosis of Structural

and Temporal Faults for *k* Bounded Non-Markovian Stochastic Petri Nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, **50**, 3369-3381.

[50] Lefebvre, D., Rachidi, S., Leclercq, E. and Pigné, Y. (2018) Temporal Fault Diagnosis for *k*-Bounded Non-Markovian SPN. 2018 5*th International Conference on Control, Decision and Information Technologies* (*CoDIT*), Thessaloniki, 10-13 April 2018, 271-276. https://doi.org/10.1109/CoDIT.2018.8394843

[51] Dugan, J., Trivedi, K., Geist, R. and Nicola, V.F. (1984) Extended Stochastic Petri Nets: Applications and Analysis. *Performance*' 84: *Proceedings of the Tenth International Symposium on Computer Performance Modeling, Measurement and Evaluation*, Paris, 19-21 December 1984, 507-519.

[52] Ajmone Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G. and Cumani, A. (1989) The Effect of Execution Policies on the Semantics and Analysis of Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, **15**, 832-846. https://doi.org/10.1109/32.29483

[53] Van Dongen, B.F., Crooy, R.A. and Van der Aalst, W.M.P. (2008) Cycle Time Prediction: When Will This Case Finally Be Finished? In: Meersman, R. and Tari, Z., Eds., *On the Move to Meaningful Internet Systems: OTM* 2008, Springer, Berlin, 319-336. https://doi.org/10.1007/978-3-540-88871-0_22

[54] Andreas Rogge, S. and Mathias, W. (2015) Prediction of Business Process Durations Using Non-Markovian Stochastic Petri Nets. *Information Systems*, **54**, 1-14. https://doi.org/10.1016/j.is.2015.04.004

[55] WFMS (2020) Workflow Management System (WFMS). https://www.igi-global.com/dictionary/workflow-management-system-wfms/32777

[56] MP (2020) Processus-Management. https://www.pyx4.com/blog/3-familles-processus-management-realisation-support/

[57] Petri, C.A. (1962) Kommunikation mit Automaten (Communication with Automata). Ph.D. Thesis, University of Bonn, Bonn.

[58] Muratat, T. (1989) Petri Nets: Properties, Analysis and Application. *Proceedings of the IEEE*, **77**, 541-580. https://doi.org/10.1109/5.24143

[59] Ramamoorthy, C.V. and Ho, G.S. (1980) Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets. *IEEE Transactions on Software Engineering*, **SE-6**, 440-449. https://doi.org/10.1109/TSE.1980.230492

[60] Laftit, S. (1991) Graphes d'événements deterministes et stochastiques: Application aux systemes de production. Thèse de Doctorat, Université de Paris, Paris.

[61] David, R. and Alla, H. (1992) Du grafcet aux reseaux de Petri. Hermes Science Publications, Paris.

[62] Dicesare, F., Harhalakis, G., Proth, J.M., Silva, M. and Vernadat, F.B. (1993) Practice of Petri Nets in Manufacturing. Chapman et Hall, London. https://doi.org/10.1007/978-94-011-6955-4

[63] Balbo, G. (2000) Introduction to Stochastic Petri Nets. In: Brinksma, E., Hermanns, H. and Katoen, J.P., Eds., *EEF School* 2000: *Lectures on Formal Methods and PerformanceAnalysis*, Springer, Berlin, 84-155. https://doi.org/10.1007/3-540-44667-2_3

[64] Ballarini, P., Gallet, E., Le Gall, P. and Manceny, M. (2019) Formal Analysis of the Wnt/*β*-Catenin through Statistical Model Checking. In: Margaria, T. and Steffen, B., Eds., *ISoLA* 2014: *Leveraging Applications of Formal Methods, Verification and Validation. Specialized Techniques and Applications*, Springer, Berlin, 193-207. https://doi.org/10.1007/978-3-662-45231-8_14

[65] Thé Workflow Management Coalition Spécification (1999) Workflow Management Coalition Terminology and Glossary. Technical Report WFMC-TC-1011, Workflow Management Coalition, Brussels.

[66] TimeNet (2020). https://timenet.tu-ilmenau.de/

[67] Ballarini, P., Djafri, H., Duflot, M., Haddad, S. and Pekergin, N. (2011) Petri Nets Compositional Modeling and Verification of Flexible Manufacturing Systems. 2011 *IEEE International Conference on Automation Science and Engineering*, Trieste, 24-27 August 2011, 588-593. https://doi.org/10.1109/CASE.2011.6042488

[68] Narahari, Y. and Viswanadham, N. (1994) Transient Analysis of Manufacturing Systems Performance. *IEEE Transactions on Robotics and Automation*, **10**, 230-244. https://doi.org/10.1109/70.282547

[69] Buzacoott, J.A. and Shantikumar, J.G. (1993) Stochastic Models of Manufacturing Systems. Prentice-Hall, New Jersey.

[70] Bizagi Modeler (2020) http://www.bizagi.com/en/products/bpm-suite/modeler/

[71] Mutarraf, U., Barkaoui, K., Li, Z.W., Wu, N. and Qu, T. (2018) Transformation of Business Process Model and Notation Models Onto Petri Nets and Their Analysis. *Advances in Mechanical Engineering*, **10**, 168781401880817. https://hal.science/hal-02475788

[72] van der Aalst, W.M.P. (1996) Three Good Reasons for Using a Petri-Net-Based Workflow Management System. In: Wakayama, T., Kannapan, S., Khoong, C.M., Navathe, S. and Yates, J., Eds., *Information and Process Integration in Enterprises*, Springer, Boston, 161-182.

[73] van der Aalst, W.M.P. (1997) Verification of Workflow Nets. In: Azéma, P. and Balbo, G., Eds., *ICATPN* 1997: *Application and Theory of Petri Nets* 1997, Springer, Berlin, 407-426. https://doi.org/10.1007/3-540-63139-9_48

[74] Meghzili, S., Chaoui, A., Strecker, M. and Kerkouche, E. (2016) Transformation and Validation of BPMN Models to Petri Nets Models Using GROOVE. 2016 *International Conference on Advanced Aspects of Software Engineering* (*ICAASE*), Constantine, 29-30 October 2016, 22-29. https://ieeexplore.ieee.org/abstract/document/7843859/authors

[75] (2020) Statistique: Estimation. https://www.math.u-bordeaux.fr/~mchabano/Agreg/ProbaAgreg1213-COURS2-Stat1.pdf

[76] Barkaoui, K. and Petrucci, L. (1998) Structural Analysis of Workflow Nets with Shared Resources. In: van der Aalst, W.M.P., De Michelis, G. and Ellis, C.A., Eds., *Workflow Management*: *Net-Based Concepts*, *Models*, *Techniques*, *and Tools* (*WFM*'98): *Proceedings of the Workshop*, *June* 22, 1998, *Lisbon*, *Portugal*: *WFM*'98, *Proceedings of the Workshop on Workflow Management*. https://api.semanticscholar.org/CorpusID:60056523

[77] Barkaoui, K., Boucheneb, H. and Hicheur, A. (2009) Modelling and Analysis of Time-Constrained Flexible Workflows with Time Recursive ECATNets. In: Bruni, R. and Wolf, K., Eds., *WS-FM* 2008: *Web Services and Formal Methods*, Springer, Berlin, 19-36. https://doi.org/10.1007/978-3-642-01364-5_2

[78] Barkaoui, K. and Hicheur, A. (2012) Towards Analysis of Flexible and Collaborative Workflow Using Recursive ECATNets. In: ter Hofstede, A., Benatallah, B. and Paik, H.Y., Eds., *BPM* 2007: *Business Process Management Workshops*, Springer, Berlin, 232-244. https://doi.org/10.1007/978-3-540-78238-4_24