

Research Article

Role of Stochastic Petri Net (SPN) in Process Discovery for Modelling and Analysis

Shabnam Shahzadi,¹ Xianwen Fang,¹ and David Anekeya Alilah ²

¹Department of Mathematics and Big Data, Anhui University of Science and Technology, Huainan, China

²Department of Mathematics, Masinde Muliro University of Science and Technology, Kakamega, Kenya

Correspondence should be addressed to David Anekeya Alilah; aliladavid2010@gmail.com

Received 3 June 2021; Accepted 18 June 2021; Published 30 June 2021

Academic Editor: Ishfaq Ahmad

Copyright © 2021 Shabnam Shahzadi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For exploitation and extraction of an event's data that has vital information which is related to the process from the event log, process mining is used. There are three main basic types of process mining as explained in relation to input and output. These are process discovery, conformance checking, and enhancement. Process discovery is one of the most challenging process mining activities based on the event log. Business processes or system performance plays a vital role in modelling, analysis, and prediction. Recently, a memoryless model such as exponential distribution of the stochastic Petri net (SPN) has gained much attention in research and industry. This paper uses time perspective for modelling and analysis and uses stochastic Petri net to check the performance, evolution, stability, and reliability of the model. To assess the effect of time delay in firing the transition, stochastic reward net (SRN) model is used. The model can also be used in checking the reliability of the model, whereas the generalized stochastic Petri net (GSPN) is used for evaluation and checking the performance of the model. SPN is used to analyze the probability of state transition and the stability from one state to another. However, in process mining, logs are used by linking log sequence with the state and, by this, modelling can be done, and its relation with stability of the model can be established.

1. Introduction

Process mining is an analytical method for finding, monitoring, and improving actual processes by extracting information from event logs that are freely available in modern information systems. It provides targeted facts based on event logs that help in doing research, analysis, and improving the existing business processes.

Presently, the process mining has been observed as an important technology and has been used in the application of business processes and hence applied successfully in many organizations. It is a process that is centric (not-data-centric), truly intelligent (read from historical data), and based on facts (event data rather than theory). In addition to process discovery, the mining process allows automatic discovery of a process model from event logs which provides insight into it and enables various types of model-based analysis. Process

acquisition models can be extended with information to predict the elimination of functional conditions. In particular, capturing of activities and waiting times in the business process are required for the understanding of the process's efficiency. In addition, these enrich types used as a source for the prediction algorithms, which are used for predicting time until the process is completed. To provide an explanation of the time forecast, a set order that allows for a good balance between “overfitting” and “underfitting” is used. It is worth noting that the words “overfitting” and “underfitting” are orthogonal to nonfitting. The model is not valid if visual cues do not occur according to that model. Establishing the remaining run time of the business process and its function is an important administrative function that allows for improved resource allocation. This also improves the quality of results when clients inquire about the status and expected completion of a given business process.

The three main types of process mining are process discovery, conformance checking, and enhancement. In this research work, attention is placed on the process discovery. In process discovery it is known that, without using any a priori information, event logs produce a process model. In case the event logs have any information about the resource, a model-related resource, for example, social networking, can be discovered, which can be able to show how different people worked in an organization collectively. On the other hand, the process model can be used for analyzing cost, resource utilization, or process performance and automation. Models are used for redesigning a process for planning and controlling in order to make decisions in a process. There are two types of process model: (i) formal model used for discussion and documentation and (ii) informal model used for analysis or enactment. In this research, informal modelling is used. There are certain errors that occur during modelling: (i) when a model defines a prepared version or a fact, (ii) failure to properly do human conduct, and (iii) when the model is at the wrong abstraction level.

For information system, analytical evaluation has become an integral part for the whole process design. Many diverse model specification techniques have been proposed, for example, Petri net, BPMN, UML function diagrams, BPEL, and EPC. Petri net is widely used in business processes either as the first model language or as a basis for validation. The Petri net was first introduced in 1962 by Carl Adam Petri. It can be described as a graphical method of the formal definition of logical interaction between components or the flow of activities in a complex system. PN is particularly well suited for modelling concurrency and conflict, sequencing, conditional branch and looping, synchronization, limited resource allocation, and mutual exclusion. It enables the study of logical properties of modelled system for each individual finite state machine, which can be seen as possible and yet shows an interaction whenever they occur when using idea behind the Petri net given by Professor Petri. Petri net is categorized as being application and theory of Petri net (ATPN) and Petri net and performance modelling (PNPM) which includes stochastic Petri net. Original Petri net did not have the notation of time and was therefore used for studying logical properties. In order to introduce time duration in Petri net, we link event data with transition that is important for availability, reliability, and performance quantification. Hence, that is the main reason why Petri net was extended to have time linked with event occurrence, giving rise to the stochastic Petri net.

Stochastic Petri net was introduced earlier in 1980. It had a graphical representation that was used for modelling of discrete events and the bipartite graph of the transitions and places in SPN is used for knowing event mechanism. It is also worth noting that the time of firing tokens is considered as a random variable and is exponentially distributed. Reachability graph in stochastic Petri net is a continuous firing rate that is linked with each transition and may be marking-dependent. If the history of a given process is known, for example, event log with time information, it is possible to extract stochastic performance data and insert it into the model. This research paper is therefore based on generalized

stochastic Petri net (GSPN) which does not restrict distributions in a particular way. GSPN is very useful in the case where some events happen in extremely small time. Whereas SPN model handles this situation by introducing immediate transition in the model which has zero firing time, the other transitions are timed transitions which are exponentially distributed. In this research, stochastic reward net is used in order to check survivability and reliability. SRN is basically an extension of GSPN and it was introduced in 1989. SRN is extensively marking-dependent which is used for firing rates and probabilities by giving priority to the transitions. Therefore SPN has an advantage in that sense, and it analyzes the probability of state transition and the stability of one state to another. For the process with logs, the log sequence is based on the results of the transition execution and not the complete state results. Hence, this paper links the log sequence with the state for stability of the model and analysis.

The remainder of the paper is organized as follows: in Section 2, related work is discussed, in Section 3, preliminary definitions are presented, and Section 4 discusses generalized distributed transition stochastic Petri net model with example. Section 5 elaborates about the stochastic reward net, survivability, and reliability model. Conclusion is presented in Section 6.

2. Related Work

Some related literature associated with Petri net model and stochastic overall performance for feature information are presented in the literature review. Hu et al. [1] proposed a technique that is primarily based on SPN model which is exponentially distributed for workflow log and depends on transition rate of firing. Anastasiou et al. [2] proposed completely unique strategies, whereby they centered at the location data for generalized stochastic Petri net model for customer modelling flows. In their research work for transition intervals, they used constant hyper-Erlang distributions which shows waiting and run times and also used GSPN to upgrade the other corresponding transitions subnet depicting the same features which the hyper-Erlang distribution has. They observed at all the transitions independently which did not cause problems in the sequence, but the similarities within the method, especially the many similar transitions, were not considered in their technique.

According to Leclercq et al. [3], non-Markovian stochastic Petri nets and attempts at eliciting exist. They were looking at a way of removing a model of normally distributed data. In their work, they focused entirely on the anticipation algorithm prepared for maximization convergence. Compared to the method used in this study, they are unable to manage lost data and different performance guidelines. The reconstruction of the model parameters of stochastic structures was also investigated in a study by Buchholz et al. [4], where they dealt with the problem of adjusting the model parameters of the basic stochastic system. Contrary to this study, the distribution of transition system was targeted in advance, although the main purpose was to make GDT_SPN version transitions which were comparable to, for example, incomplete Wombacher and

Jacob's [5] statistical estimates' distribution does not have initial process times. Rozinat et al. [6] checked out how to acquire records for simulation models in an attempt to discover data dependencies which are mostly taken into consideration for discovering optimal standard alignment between model and log, which are doing manual replies that make decisions, mean durations, and trendy deviations because before that these were not taken under consideration. The technique proposed in this paper has successfully dealt with noise in a much far better way through building the notion of alignments, which is able to pick out the finest direction through the model for a noisy trace. According to van der Aalst [7], the available process mining techniques are used to consider the noise and probability for creating control flow, and the importance of business process modelling is recognized. The best way to represent these methods is stochastic Petri net, which is the main research task of the process mining in business modelling. Rogge-Solti et al. [8] proposed an algorithm in their research work for process discovery of each execution and also used different raw event data for discovering various classes of SPN. In their study, they used notation alignment which is based on plug-in in process mining framework.

3. Preliminaries

Here concepts and some techniques are presented which are used throughout the paper. Primarily, our main focus is towards event logs and PN, SPN, GSPN, SRN and log sequence.

3.1. Event Log. It is a set of collected cases $L \subseteq C$ so that each event performs once in the whole log; i.e., for any $c_1, c_2 \in L$ such that $c_1 \neq c_2: \partial_{\text{set}}(\widehat{c}_1) \cap \partial_{\text{set}}(\widehat{c}_2) = \emptyset$. If the event log contains time stamps, tracing order must respect these time stamps. That is, for any $c \in L, i$ and j such that $1 \leq i < j \leq |\widehat{c}|: \#_{\text{time}}(\widehat{c}(i)) \leq \#_{\text{time}}(\widehat{c}(j))$.

3.2. State. It means a particular condition that a system uses for some specific purpose at specific time. Let $c = (a, a_i, a_0, d, i, o)$ be c - net. $s = b(a \times a)$ represents state space of c . $s \in S$ is a state which shows a multiset of pending obligations.

3.3. Sequence. The most natural way to present the traces in the event log is by sequence. At a point when we need to explain the functional semantics of PN and transition of how a system is made, and the performance is also modelled in terms of sequences, some of the useful operators on sequences are presented. If A is a set, then A^* represents a set of all finite sequences over A . A finite sequence over A with length n is a mapping $\sigma \in \{1, 2, \dots, n\} \rightarrow A$. In this case, the sequence is represented in string form; that is, $\sigma \in \{a_1, a_2, \dots, a_n\}$, where $a_i = \sigma(i)$ for $1 \leq i \leq n$. $|\sigma|$ shows length of the sequence; that is, $|\sigma| = n$. $\sigma \oplus a' = \langle a_1, a_2, \dots, a_n, a' \rangle$ represents sequence with element a' . Similarly, $\sigma_1 \oplus \sigma_2$ adds sequences σ_1 and σ_2 causing a sequence of length $|\sigma_1| + |\sigma_2|$.

3.4. Petri Net. Petri nets were initially presented by C. A. Petri in 1962. Formally, a Petri net (PN) is a 5-tuple $\text{PN} = (P, T, F, W, M)$, where

- (i) $P = \{p_1, p_2, \dots, p_n\}$ is a finite set of places (as circles)
- (ii) $T = \{t_1, t_2, \dots, t_n\}$ is a finite set of transitions (as bars)
- (iii) $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs connecting P and T
- (iv) $W: F \rightarrow \{1, 2, 3, \dots\}$ is a weighted function
- (v) $M: P \rightarrow \{0, 1, 2, 3, \dots\}$ marking represents number of tokens (black dots) in place P and initial marking is represented as M_0 .

PN is a graphical representation of nodes based on places and transitions. The arc that connects places by transition is called the input transition arc, and the arc connected from transition to places is called the output transition arc, and the positive number is connected to each arc. Places connected to the transition with the input arc are called input places and vice versa. Each place may have zero or more tokens. Transitions are enabled if each of the input places has at least as many tokens corresponding to the input arc. Enabled transitions can be fired. When we have fired token, the number of tokens that are equal to the input arcs multiplicity is transferred from each input place, whereas the number of tokens which are equal to the output arc multiplicity is deposited in the output places. Petri net transformation of one marking into another which is done by using firing transitions, in reachability set M_0 , shows the initial marking and is defined as the sequences of the firing transitions followed by marking and always starts through initial marking. It is known that a PN is a 5-tuple $\text{PN} = (P, T, F, W, M)$.

Figure 1 describes a Petri net model as a marking vector that calculates the number of tokens in each net:

$$(n_1, n_2, \dots, n_p), \quad \text{with } p \text{ representing the number of places.} \quad (1)$$

Transition is enabled in case the input places have the required number of tokens.

Figure 2 shows that the enabled transitions can be fired by taking a certain number of tokens from the input places and then placing them in the output places, for example.

In real world, Petri net can be used in many situations based on sequences, concurrency, synchronization, and conflict. We present in Figure 3 concurrency/parallelism of stochastic Petri net model, in Figure 4 we present the independency, and Figure 5 shows the synchronization of the stochastic Petri net model. Reachability analysis of stochastic Petri net is present in Figure 6. However, in computer networking, Petri net describes communication protocols. In original definition of PN, concept of time is not included for the performance evaluation of dynamic system; therefore, it is important to present time delay that is linked with each transition in Petri net model. This concept of time delay in transition firing emerges in the SPN.

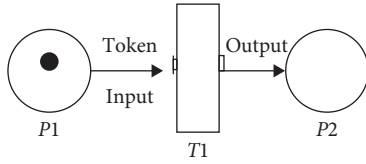


FIGURE 1: General firing sequence of Petri net.

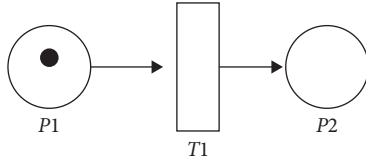


FIGURE 2: Enabled transitions.

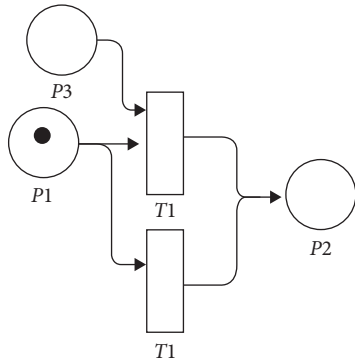


FIGURE 3: Concurrency/parallelism.

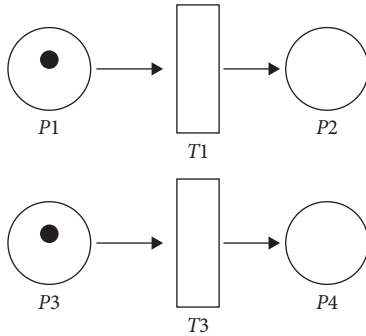


FIGURE 4: Independency.

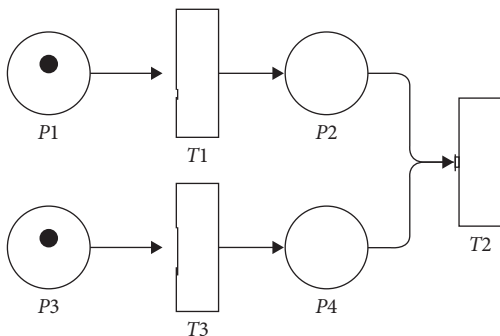


FIGURE 5: Synchronization.

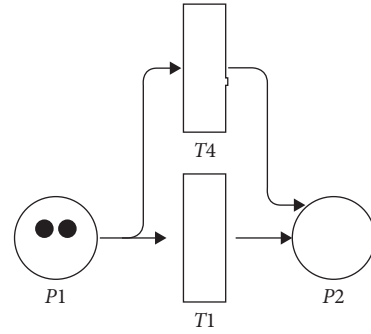


FIGURE 6: Reachability analysis.

3.5. *Stochastic Petri Net.* In SPN, data perspective is used because flow of the information between the tasks can be described by it and each transition is exponentially distributed with firing time. SPN can be described as a workflow net if (i) there is only one place to start and (ii) one place to end and (iii) every node is on track from the start to the end of the task. For a generalized stochastic Petri net (GSPN), the transitions are either timed (firing time represented by rectangular box) or immediate (zero firing time, represented by black bar). Priority is always given to the immediate transition over timed transition for firing. Usually, we use probability mass function in order to complete the firing and break the tie between immediate transitions. In GSPN marking is completed when at least an immediate transition enables the marking to be visible. Inhibitor arcs are also introduced in the GSPN which are used to connect places to transition. At the terminating point of arrow there is a small hollow circle which indicates that it is an inhibitor arc. If input places of the inhibitor arcs contain more tokens than the multiplicity arcs, then at that point transition cannot be fired with the inhibitor arc.

It has been proven that the condition of limited number of transitions is related to continuous time Markov chain (CTMC) with GSPN that can be fired in a limited time with nonzero probability. When stochastic Petri net is used in computer network for performance evaluation, places are denoted with packets or cell in buffer or active user or flow in the system, whereas their arrival and departure are represented with transitions.

4. Generalized Distributed Transition Stochastic Petri Net

It is based on seven-tuple $\{p, \tau, P, W, F, M_0, D\}$, whereas the basic Petri net has $\{P, T, F, M_0\}$, where

- (i) T is a set of transitions which is equal to $T_i \cup T_t$ based on immediate and rimed transitions
- (ii) $P: T \Rightarrow N_0^+$ is the priority to transition, where $\forall_t \in T_i: P(t) \geq 1$ and $\forall_t \in T_t: P(t) = 0$
- (iii) $W: T_i \Rightarrow R^+$ represents the probability weight to immediate transition

- (iv) $D: T_t \Rightarrow D$ is an arbitrary probability distribution D to timed transition, which reflects duration of corresponding activities

In Figure 7 we present the generalized distributed transition stochastic Petri net model for two parallel branches, and the above model shows the conflict between transitions T_c and T_d .

4.1. Case-Based Alignment. Case-based alignment method is much more powerful than the naïve replays of logs in the model because it ensures that global availability best penalizes asynchronous part of replay. Since there are two execution traces tr_1 and tr_2 of the above model, it is assumed that immediate transitions are invisible, whereas the timed transitions are visible. In this study, the focus is geared towards the invisible transitions in the model alignment which is denoted by τ , whereas tr_2 is not fitted in the model. So, to overcome this, optimal alignment between model and log is found out using the method proposed by Adriansyah et al. [7], which gives sequence of replay movements in the trace and model.

Therefore, these movements can be either log moves, synchronous moves, or model moves. Table 1 shows the extension traces of the model which are the matching subscript of each event in the transition net. Table 2 gives us perfect alignment for tr_1 based on synchronous or invisible model moves. Also, multiple alignments for trace tr_2 are presented in Table 3. Hence, Table 4 shows the event logs which are based on the optimal alignment of the model and log. The symbol \gg shows no progress on either side. Cost-based alignment provides the unnecessary moves which are penalized, and the high costs are not included in the optimal alignments.

5. Stochastic Reward Net

(SRN): basically, SRN is an extension of the GSPN. Reward rate, in stochastic reward net, is linked with each visible marking. Therefore, there are many ways of measuring performance. It also gives us many other ways that make specification convenient, which are as follows:

- (i) Every transition can have enabling function (also referred to as security guard), which only makes the transitions enabled if their marking dependency function is correct. This specification gives a superior way to enhance the graphical representation and makes SRN easy to understand.
- (ii) Repeated arcs marking is permitted. This characteristic can be used in a case when transformation of total number of tokens depends on the present marking.
- (iii) Marking of dependent firing rate is allowed. This specification enables the firing value of the transition to be specified as a token number in any Petri net.
- (iv) Transitions can be provided with different priorities, and transitions are enabled only if no other high-value transition is enabled.

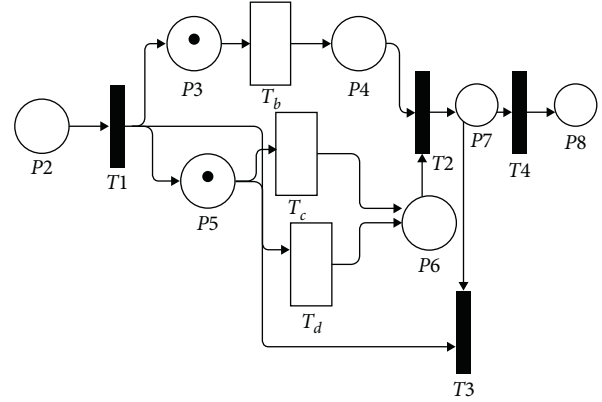


FIGURE 7: Generalized distributed transition stochastic Petri net model for two parallel branches; conflict between T_c and T_d .

TABLE 1: Small log.

tr_1 :	(a,	b,	d,	c,	b)
tr_2 :	(b,	d,	d)		

TABLE 2: tr_1 : perfect alignment.

Log	a	\gg	b	d	\gg	\gg	\gg	c	b	\gg	\gg
Model	a	τ	b	d	T	τ	τ	c	b	τ	τ
	t_a	t_1	t_b	t_d	t_2	t_4	t_1	t_c	t_b	t_2	t_3

TABLE 3: tr_2 : two possible alignments.

Log	\gg	\gg	B	d	d	\gg	\gg
Model	A	τ	B	d	\gg	τ	τ
	t_a	t_1	t_b	t_d		t_2	t_3

TABLE 4: Event log-based alignment.

Log	\gg	\gg	b	d	\gg	\gg	\gg	c	\gg	\gg	\gg
Model	A	τ	b	d	τ	τ	τ	c	b	τ	τ
	t_d	t_1	t_b	t_d	t_2	t_4	t_1	t_c	t_b	t_2	t_3

- (v) The expectation of traditional withdrawal measures found in GSPN as the inclusion of a transition and the mean number of tokens in a place, the most difficult reward work can now be defined.

Primarily availability of assessment approaches depends on modelling and measurement methods. System availability model-based approach is more effective and inexpensive for analysis and comparison than measurement-based approach. Discrete-event simulation can be used for system modelling, whereas, for analytical modelling, both approaches can be used. Analytical modelling can be categorized into four main parts:

- (1) Nonstate-space model (reliability graph)
- (2) Reliability block diagram (RBD)
- (3) Fault tree (FT)
- (4) State-space models (Markov chain, stochastic Petri net, among others)

The hierarchical models, fixed iterative point models, and nonstate paradigms provide a quick overview of the system basic metrics (reliability, availability, and MTTF) with their proper scanning and the architecture of the system. However, state-space models capture complex functionality and system performance. This approach can also be able to manage failure or fix dependencies and complex connection between the system components. In order to avoid a wide range of problems at the point state level, modelling approach for nonstate-space and state-space models for the other points can also be used.

Availability $a(t)$ shows the probability of correct state at any instant t in operating system without taking into consideration the interval $(0, t)$. The instantaneous availability $a(t)$ related to system reliability is computed by

$$a(t) = r(t) + \int_0^t r(t-x)q(x)dx. \quad (2)$$

$r(t)$ shows instantaneous reliability at time t and can be defined as follows:

$$r(t) = \int_t^{\infty} f(x)dx. \quad (3)$$

$f(x)$ is the probability density function for random variable x which represents the system lifetime or time to failure; $q(x)$ represents the renewal process rate in the interval $(0, t)$.

$$q(x) = f(x) + \int_0^x q(x-v)f(v)dv. \quad (4)$$

$w(x)dx$ represents probability of renewal process cycle that will be completed in the time interval $(x, x+dx)$. $r(t-x)$ shows the probability of the system which works properly in the remaining time interval (t, x) . $r(t-x)w(x)dx$ shows the probability of the case where fault has occupied, and repair or renewal will reassume functioning with no further faults. The concept of $a(t)$ is similar to the reliability $r(t)$ in the case where the system is not repairable.

For a long running time, we have steady-state availability (SSA), where at initial state we have limiting value of $a(t)$ decreasing by 1. Therefore,

$$a = a(t) = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}, \quad (5)$$

$$a = a(t) = \frac{\mu}{\mu + \lambda},$$

where λ represents the time failure rate and system and μ , which is the repair time, is determined as average of number of repairs over maintenance tie.

MTTF (mean time to failure) represents the expected time in which a system functions correctly before its first failure. Mean time to repair (MTTR) shows the expected

time in which a system is used for repair. At the point where both mean time to failure (MTTF) and mean time to repair (MTTR) are exponentially distributed, the arithmetic inversion of failure and the repair rate of the system are as follows:

$$\text{MTTF} = \frac{1}{\lambda},$$

$$\text{MTTR} = \frac{1}{\mu}. \quad (6)$$

In hardware and software of industrial process, stochastic reward net is an appropriate modelling tool. In Figure 8, we present SRN availability model which can specify the system operations by using transitions, arcs, and places, which are the main parts. Stochastic reward net for availability framework is based on three stages:

- (1) Requirements specifications
- (2) SRN-based system modelling
- (3) System analysis

5.1. Survivability and Reliability Model for Performance Checking. Survivability and reliability model can be used to check the performance of transient behavior after the occurrence of failure, attack, and disaster among others. It can be seen that a generalization of recovery time means how much effect can be put during the recovery time. Survivability can check out the performance of the model by using “system average interpretation duration index” in a short form called SAIDI model.

The predictive model of SAIDI is

$$\text{SAIDI} = \sum_{i=1}^c \varnothing_i E(X_i) \frac{E[M_i(X_i)]}{E[D_i(X_i)]}, \quad (7)$$

where $E(X_i)$ shows the availability model and $E[M_i(X_i)]/E[D_i(X_i)]$ shows survivability. In short, it can be said that SAIDI is combination of availability and survivability model.

Hence, \varnothing_i means number of failures at section “ i ,” after failure of section “ i ,” we have

- (i) X_i : time up to full recovery
- (ii) $D_i(X_i)$: energy demanded up to full recovery
- (iii) $M_i(X_i)$: energy not supplied up to full recovery

It can also be used to predict the recovery matrix; after coupling with the availability model, the generalized SAIDI form is as follows:

$$\text{SAIDI} = \frac{\text{Customer interruption duration}}{\text{Total number of customers}}. \quad (8)$$

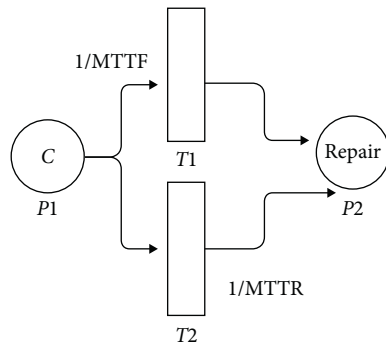


FIGURE 8: Stochastic Petri net availability model.

6. Conclusion

In conclusion, it is observed that the performance of the system or business process plays an important role in modelling, analysis, and prediction. Nowadays, memoryless model such as exponentially distributed stochastic Petri net (SPN) has gained much attention in research and industry. This paper is based on time perspective for modelling, analysis, and use of stochastic Petri net to check the performance, evolution, stability, and reliability of the model. To know the effect of time delay in firing the transition, we use stochastic reward net (SRN) model. Stochastic reward net (SRN) model can also be used to check the reliability of the model. Generalized stochastic Petri net (GSPN) is used for evaluation and checking of the performance of the model. It is known that stochastic Petri net is used to analyze the probability of state transition and the stability from one state to another, whereas, in mining process, logs are used by linking log sequence with the state which enables modelling to be done and relates it with stability of the model. Generalized distributed transition stochastic Petri net model is used for checking the performance of the model. Case-based alignment is used to check the optimal alignment between the traces. SAIDI model is used to check the survivability and reliability of the generalized stochastic Petri net model. This paper deals with mathematical and theoretical work that shows how checking the importance of stochastic Petri is done in order to know the performance of the discovery process model and analysis. Further work can be done on its practical application.

Data Availability

The data used to support the findings of the study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

- [1] H. Hu, J. Xie, and H. Hu, "A novel approach for mining stochastic process model from workflow logs," *Journal of Computational Information Systems*, vol. 7, no. 9, pp. 3113–3126, 2011.
- [2] N. Anastasiou, T. Hornig, and W. Knottenbelt, "Deriving generalized stochastic petri net performance models from high-precision location tracking data," in *Proceedings of the 5th International ICST Conference on Performance Evaluation Methodologies and Tools*, pp. 91–100, Paris, France, May 2011.
- [3] E. Leclercq, D. Lefebvre, and S. Ould El Mehdi, "Identification of timed stochastic petri net models with normal distributions of firing periods," *IFAC Proceedings Volumes*, vol. 42, pp. 948–953, 2009.
- [4] R. Buchholz, C. Krull, and G. Horton, "Reconstructing model parameters in partially-observable discrete stochastic systems," in *Analytical and Stochastic Modeling Techniques and Applications*, pp. 159–174, Springer, Berlin, Germany, 2011.
- [5] A. Wombacher and M. E. Iacob, "Start time and duration distribution estimation in semi-structured processes," Technical report, Centre for Telematics and Information Technology, University of Twente, Enschede, Netherlands, 2012.
- [6] A. Rozinat, R. S. Mans, M. Song, and W. van der Aalst, "Discovering simulation models," *Information Systems*, vol. 34, no. 3, pp. 305–327, 2009.
- [7] A. Adriansyah, B. F. van Dongen, and W. M. P. van der Aalst, "Conformance checking using cost-based fitness analysis," in *Proceedings of the 2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, pp. 55–64, Helsinki, Finland, August 2011.
- [8] A. Rogge-Solti, W. van der Aalst, and W. Mathias, *Discovery Stochastic Petri Nets with Arbitrary Delay Distribution from Event Log*, Springer, Berlin, Germany, 2014.