Compositional Evaluation of Stochastic Workflows for Response Time Analysis of Composite Web Services

Laura Carnevali, Riccardo Reali, Enrico Vicario
{laura.carnevali,riccardo.reali,enrico.vicario}@unifi.it
Department of Information Engineering, University of Florence
via di Santa Marta 3, 50139 Florence, Italy

ABSTRACT

Workflows are patterns of orchestrated activities designed to deliver some specific output, with application in various relevant contexts including software services, business processes, supply chain management. In most of these scenarios, durational properties of individual activities can be identified from logged data and cast in stochastic models, enabling quantitative evaluation of time behavior for diagnostic and predictive analytics. However, effective fitting of observed durations commonly requires that distributions break the limits of memoryless behavior and unbounded support of Exponential distributions, casting the problem in the class of non-Markovian models. This results in a major hurdle for numerical solution, largely exacerbated by the concurrency structure of workflows, which natively subtend concurrent activities with overlapping execution intervals and a limited number of regeneration points, i.e., time points at which the Markov property is satisfied and analysis can be decomposed according to a renewal argument.

We propose a compositional method for quantitative evaluation of end-to-end response time of complex workflows. The workflow is modeled through Stochastic Time Petri Nets (STPNs), associating activity durations with Exponential distributions truncated over bilateral firmly bounded supports that fit mean and coefficient of variation of real logged histograms. Based on the model structure, the workflow is decomposed into a hierarchy of subworkflows, each amenable to efficient numerical solution through Markov regenerative transient analysis. In this step, the grain of decomposition is driven by non-deterministic analysis of the space of feasible behaviors in the underlying Time Petri Net (TPN) model, which permits efficient characterization of the factors that affect behavior complexity between regeneration points. Duration distributions of the subworkflows obtained through separate analyses are then repeatedly recomposed in numerical form to compute the response time distribution of the overall workflow.

Applicability is demonstrated on a case from the literature of composite web services, here extended in complexity to demonstrate scalability of the approach towards finer grain composition schemes, and associated with a variety of durations randomly selected from a data set in the literature of service oriented computing, so as to assess variability of accuracy and complexity of the overall approach with respect to specific timings.

ICPE '21, April 19–23, 2021, Virtual Event, France

ACM ISBN 978-1-4503-8194-9/21/04…$15.00
https://doi.org/10.1145/3427921.3450250

CCS CONCEPTS

• **General and reference** → *Performance*; • **Theory of computation** → **Stochastic approximation**; • **Information systems** → **Web services**.

KEYWORDS

Stochastic workflows, Markov regenerative processes, regenerative transient analysis, performance evaluation, composite web services.

ACM Reference Format:
Laura Carnevali, Riccardo Reali, Enrico Vicario. 2021. Compositional Evaluation of Stochastic Workflows for Response Time Analysis of Composite Web Services. In *Proceedings of the 2021 ACM/SPEC International Conference on Performance Engineering (ICPE '21), April 19–23, 2021, Virtual Event, France.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3427921.3450250

1 INTRODUCTION

Workflows of orchestrated activities arise in the practice of a wide variety of application contexts, with a *structure* reflecting designed or discovered artifacts, such as a Bill of Materials (BOM) [17], a Business Process Model (BMP) [1], or the Business Process Execution Language (BPEL) specification of a composed web service [15]. Individual activities in the workflow can often be enriched with *durational properties* derived from operation data [11, 37], acceptable assumptions [26], or Quality of Service (QoS) contracts [46], so as to obtain a stochastic model that opens the way to transient analysis for quantitative evaluation of time behavior, which in turn enables diagnostic, predictive, and prescriptive analytics.

As a common trait of all these scenarios, the validity of stochastic characterization normally requires that observed durations be described by general distributions (GEN) beyond the limit of memoryless EXP variables, and the representation of firm synchronization constraining concurrency often requires that durations be associated with a bounded support. Besides, the structure of concurrency of the workflow normally results in activities with overlapping execution intervals, limiting the number of regeneration points where the Markov condition is satisfied. The combination of these aspects casts the underlying stochastic process of the model in the class of Generalized Semi Markov Processes (GSMPs), which impairs efficient numerical solution techniques.

If the workflow model never reaches a state where two GEN activities overlap their durations, i.e., if it satisfies the so-called *enabling restriction*, then transient analysis can still resort to numerical solution techniques [14, 21, 40], also with the support of various tools [2, 41, 52]. Numerical methods have been formulated

to overcome the enabling restriction for models with deterministic (DET) and EXP timers [28, 30], though, in the practice, complexity again limits applicability to models with at most one DET timer enabled in each state. Yet, the structure of concurrency of a workflow normally includes different branches each with GEN durations starting at different time points. Multiple concurrent GEN durations can be managed by approximation through Continuous PHase type (CPH) distributions [8, 24, 35, 36], which brings back the model in the class of Markovian behavior at the cost of a state space expansion that trades accuracy for complexity and prevents representation of firmly bounded duration constraints. Following a different approach, transient analysis of workflows with concurrent overlapping activities with GEN durations can be performed through the method of stochastic state classes [23, 34], which achieves efficiency through symbolic manipulation of the analytical form of (multivariate) Probability Density Functions (PDFs), but requires that the degree of parallelism among concurrent GEN transitions remains limited, which is often not the case in tree-like or graph-like structured workflows.

Various compositional approaches have been proposed to face the problem, mainly with reference to workflows structured as trees, attaining different trade-offs between approximation and complexity. In [13], a stochastic fault tree where times to failure of leaf nodes are expolynomially distributed over a possibly bounded support is analyzed in bottom-up order by repeated derivation of the analytical form of the times to failure of intermediate nodes, fighting complexity through the pruning of expolynomial terms that are negligible within a given time window of prediction. In [4], attack trees with Erlang (and EXP) distributed activity durations combined according to Boolean and sequence gates are analyzed by approximating times with Acyclic Phase-Type (APH) distributions, while keeping a bounded complexity through a compression algorithm that controls the number of phases in the distribution. In [19], a stochastic tree with Erlang distributed times to failure of leaves is analyzed by approximating times to failure of intermediate gates with complexity-scaled distributions, while maintaining a stochastic order that guarantees safe approximation. In [50], semi-Markov models of composite web services with failures and restarts are analyzed to derive the closed-form of mean and variance of the end-to-end response time, also in the specific case that durations of atomic services are fitted through APH distributions [31].

In this paper, we propose a compositional approach for quantitative evaluation of the time behavior of complex workflows combining concurrent and sequential activities with generally distributed durations over bounded supports. The workflow is modeled through Stochastic Time Petri Nets (STPNs) where the duration of each activity is characterized by a shifted truncated EXP distribution.

The model structure drives the decomposition of the workflow into a hierarchy of subworkflows amenable to efficient solution by Markov regenerative transient analysis based on the method of stochastic state classes. To this end, nondeterministic analysis of the Time Petri Nets (TPNs) underlying the subworflows is performed to enumerate the space of possible behaviors, characterizing the degree of concurrency among activities and the number of events occurring while an activity is being executed. In turn, these features open the way to the definition of heuristics that are able to estimate the complexity of deriving a measure of probability associated with possible behaviors, determining whether regenerative analysis is affordable or the subworkflows need to be decomposed further. Then, the duration distributions of the decoupled subworflows obtained through separate regenerative analyses are repeatedly recomposed in numerical form, computing the end-to-end response time distribution of the overall workflow.

Application of the approach is motivated and demonstrated addressing the evaluation of the end-to-end response time distribution of composite web services, assuming that concurrency effects due to multiple users are negligible given that individual services can be horizontally scaled. Specifically, a case from the literature [11] is considered, characterizing the execution times of activities through shifted truncated EXP distributions that fit mean and variance of real logged histograms obtained from the WS-DREAM data set [51], widely used in the literature of web services. Notably, the model is extended in complexity to illustrate scalability of the approach, and experimented with different classes of timings.

The rest of the paper is organised in five sections. In Section 2, we provide an overview of the overall approach. In Section 3, we recall preliminary concepts on STPNs, Markov regenerative transient analysis, and nondeterministic analysis. In Section 4, we specify the addressed class of workflows. In Section 5, we develop the solution approach, describing decompostion of the model structure and recomposition of results, characterizing the main factors of complexity. In Section 6, we illustrate application with reference to the context of composite web services. In Section 7, we draw conclusions and discuss next directions. For the sake of readability, formal syntax and semantics of STPNs are recalled in the Appendix.

2 APPROACH OVERVIEW

Fig. 1 shows the Data Flow Diagram (DFD) of the overall approach which consists of three steps implemented in a toolchain available at https://doi.org/10.5281/zenodo.4519118:

- **Distribution derivation (processes 1– 3, Section 4.2).** The parameters of the shifted truncated EXP distributions of individual activities are selected so as to fit mean and variance of logged histograms (process 3) which, in turn, are derived from the WS-DREAM data set by parsing data (process 1) and removing outliers through the Inter-Quartile Range (IQR) rule, also known as Tukey's rule of thumb [42] (process 2).
- **Model generation (processes 4– 7, Section 4.1).** The STPN model of the workflow is generated by specifying the workflow structure as a Petri Net (PN) made of places, untimed transitions, and directed arcs (processes 4 and 5), and associating each transition modeling a workflow activity with an execution time distribution, randomly selected among those obtained at the previous step (processes 6 and 7).
- **Model evaluation (processes 8– 12, Section 5).** The end-to-end response time CDF of the overall workfow is obtained by decomposing it into subworkflows (process 8), performing regenerative transient analysis of the corresponding STPNs (process 9), and recomposing in numerical form the obtained response time CDFs of subworkflows (process 10). Accuracy measures are computed by comparing the obtained results

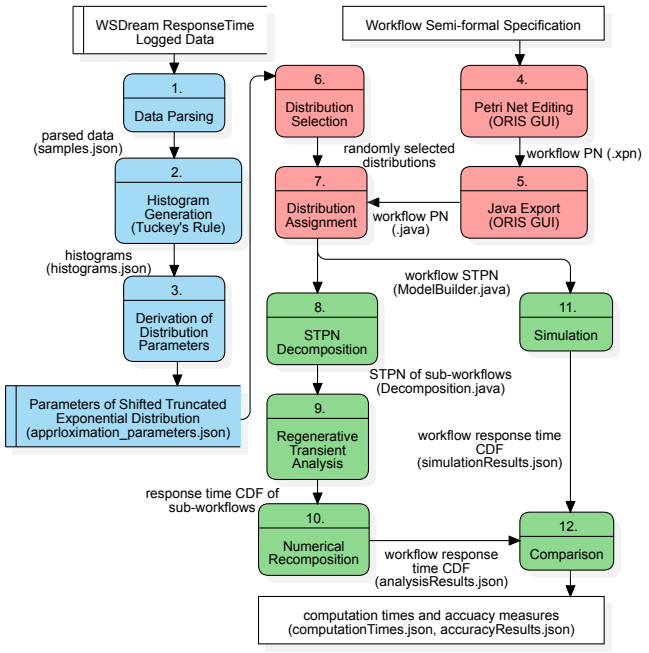

Figure 1: DFD of the approach. Processes in blue derive duration distributions; processes in red address model generation; processes in green perform model decomposition and evaluation, and numerical recomposition of results.

with a ground truth (process 12) derived through stochastic simulation of the workfloW STPN (process 11).

The approach is implemented in a toolchain that relies on Java 9, exploiting the SIRIO library [39] of the ORIS tool [34] to perform analysis and simulation of STPN models, and makes a limited use of Wolfram Mathematica (to derive the parameters of the execution time distributions) and Python (to parse data and plot results).

3 PRELIMINARIES

In this section, we provide background concepts on: STPNs (Section 3.1), used to specify workflows made of concurrent activities with bounded GEN duration; Markov regenerative transient analysis based on the method of stochastic state classes (Section 3.2), used to evaluate the response time distribution of the sub-workflows; and, nondeterministic analysis of TPNs (Section 3.3), used to enumerate the space of possible behaviors of the sub-workflows, which supports the characterization of the main factors that determine the complexity of regenerative transient analysis.

3.1 Stochastic Time Petri Nets

STPNs are a variant of non-Markovian stochastic Petri nets modeling stochastic timed systems where multiple concurrent events can be constrained to occur within bounded time intervals. Specifically, in an STPN, *transitions* (depicted as vertical bars, as in Fig. 2) represent events, *places* (depicted as circles) containing *tokens* (depicted as dots) model logical conditions that enable events, and *directed arcs* from input places to transitions and from transitions to output places determine token moves performed at the occurrence of events. A transition becomes enabled when all its input places contain at least a token, sampling a time to fire from a probability distribution possibly with bounded support. The transition with minimum time to fire is selected, removing a token from each of its input places and adding a token to each of its output places. The choice among transitions with equal time to fire is solved by a random switch determined by probabilistic weights. In so doing, an STPN decorates its underlying TPN through probability distributions and weights of transitions, thus associating the set of timed behaviors of the TPN with a measure of probability.

3.2 Regenerative transient analysis

STPNs with multiple concurrent GEN transitions can be analyzed with relative efficiency if the model satisfies the Markov property always with probability 1 at specific time instants, termed *regeneration points*. In this case, the model subtends a Markov Regenerative Process (MRP) [25], which can be solved numerically provided that the process is characterized in terms of a *local kernel*, describing the behavior until the first regeneration, and a *global kernel*, describing sequencing and timing of visits to subsequent regenerations. Solutions for evaluation of kernels have been consolidated under various restrictions, the most notable being the *enabling restriction*, requiring that GEN durations never overlap their activity cycles [9], and the *bounded regeneration restriction*, requiring a bounded number of firings between consecutive regeneration points [7].

Transient analysis of STPNs that subtend an MRP satisfying the bounded regeneration restriction can be performed through *regenerative analysis* based on the method of *stochastic state classes* [12, 23]. Given a sequence of firings, a stochastic state class encodes the marking plus the joint domain and (symbolically) the joint PDF of the absolute elapsed time and the times to fire of the enabled transitions. The joint domain can be efficiently encoded as a Difference Bounds Matrix (DBM) [18, 45], i.e., the solution space of a set of linear inequalities constraining the difference between pairs of timers. The joint PDF takes a continuous *piece-wise* representation over a partition of the domain in DBM zones and can be derived in closed-form if all the transitions of the STPN have a distribution in the class of expolynomial functions [41]. Enumeration of stochastic state classes between any two regenerations enables the computation of the local and global kernels. Then, numerical solution of Markov renewal equations formulated in terms of the kernels [25] provides transient marking probabilities, i.e., $p_m(t) := P\{M(t) = m\}$ $\forall\, t \in [0, t_{\max}]$, $\forall\, m \in \mathcal{M}$, where $M(t)$ is the process describing the marking of the STPN at time $t \in [0, t_{\max}]$, $t_{\max}$ is the analysis time limit, and $\mathcal{M}$ is the set of reachable markings.

The complexity of regenerative transient analysis significantly depends on the occurrence of regeneration points in the MRP underlying the STPN, which, in turn, depends on the degree of concurrency of the model and on the length of behaviors (in terms of number of firings) during which GEN transitions remain persistent. On the one hand, the occurrence of regeneration points cannot be easily controlled during the activity of model construction, so that minor modeling choices may result in major variations in the structure and complexity of the underlying MRP. On the other

hand, the MRP structure and complexity may be reduced only at the cost of significantly limiting the level of detail of the model, which may yield too coarse-grained results for complex systems, such as execution paths of web service architectures, industrial production processes, or large-scale fault-tolerant architectures. Therefore, efficient approaches are needed to decompose the model into analyzable sub-models and to derive estimates of the measures of interest from the results of separate analyses.

3.3 Nondeterministic analysis

Identification of the space of timed behaviors of the TPN underlying an STPN can be performed through *nondeterministic analysis* based on the method of *state classes* [45]. Given a sequence of transitions which can be fired with different timings, a state class consists of the marking reached through that firing sequence plus the joint domain of the remaining times of the transitions enabled after that firing sequence. The joint domain can be efficiently encoded and manipulated as a DBM [18], with polynomial complexity with respect to the number of the enabled transitions. Enumeration of state classes yields a graph termed *state class graph*, largely exploited in tools for verification of qualitative properties of concurrent timed systems [5, 6, 16, 20, 34]. In particular, the state class graph is sufficient to identify state classes that correspond to regeneration points (which we term *regenerative state classes*) as well as to characterize the concurrency degree of the GEN transitions and the number of firings to which a GEN transition is persistent. In particular, regenerative state classes are state classes where all the GEN transitions are newly enabled or DET or have a DET delay with respect to a newly-enabled transition. Notably, the state class graph can be proved to be finite under fairly general conditions, requiring in particular that the number of reachable markings be finite and the earliest and latest firing times of transitions be rational values [23].

4 STOCHASTIC MODEL

In this section, we illustrate the structure of concurrency of the STPN model of a workflow (Section 4.1) and the derivation of its stochastic parameters from available statistics of the execution times of its activities (Section 4.2), and we define the response time distribution of the workflow (Section 4.3).

4.1 Structure of concurrency

We consider workflows consisting of concurrent activities with GEN execution time, composed according to the *sequence*, *AND-split* (i.e., parallel split), *AND-join* (i.e., synchronization), *XOR-split* (i.e., exclusive choice), and *XOR-join* (i.e., simple merge) patterns [44]. Workflows can be modeled by STPNs, which support the representation of sequential, concurrent, and alternative behaviors with GEN duration, according to the considered control flow patterns. To illustrate the elements of structural complexity, Fig. 2 shows the STPN of a workflow with 5 sequences (modeled by transitions `t4` and `t7`, `t5` and `t8`, `t6` and `t9` , `t21` and `t23`, `t22` and `t24`); 3 AND-splits (modeled by transitions `t1`, `t13`, `t15`); 6 AND-joins (modeled by transitions `t11`, `t19`, `t20`, `t25`, `t26`); 1 XOR-split (modeled by transitions `t2`, `t3`); and, 1 XOR-join (modeled by transition `t10`).

Split-join patterns can be nested in well structured constructs [43]. In particular, an AND-join (or XOR-join) collects all the paths originated from the last unjoined AND-split (or XOR-split). For instance, the IMM transitions `t10` and `t11` account for an XOR-join and an AND-join, respectively, collecting all the paths originated from the XOR-split represented by transitions `t2` and `t3` and the AND-split represented by transition `t1`, respectively. Conversely, the IMM transition `t25` accounts for an AND-join collecting independent paths, i.e., not originated from the same unjoined AND-split.

Note that a token is contained in the input place of each GEN transition representing an initial activity of the workflow and in the input place of each IMM transition representing an XOR-split among initial activities of the workflow, which we term *initial places* of the STPN of the workflow, e.g., the workflow represented in Fig. 2 has 3 initial activities represented by the GEN transitions `t0`, `t21`, and `t22`, whose input places `p0`, `p24`, and `p27`, respectively, contain one token each. Moreover, the GEN transitions that represent a possible final activity of the workflow and the IMM transitions that represent a join of possible final activities of the workflow have the same output place, which we term *final place* of the STPN of

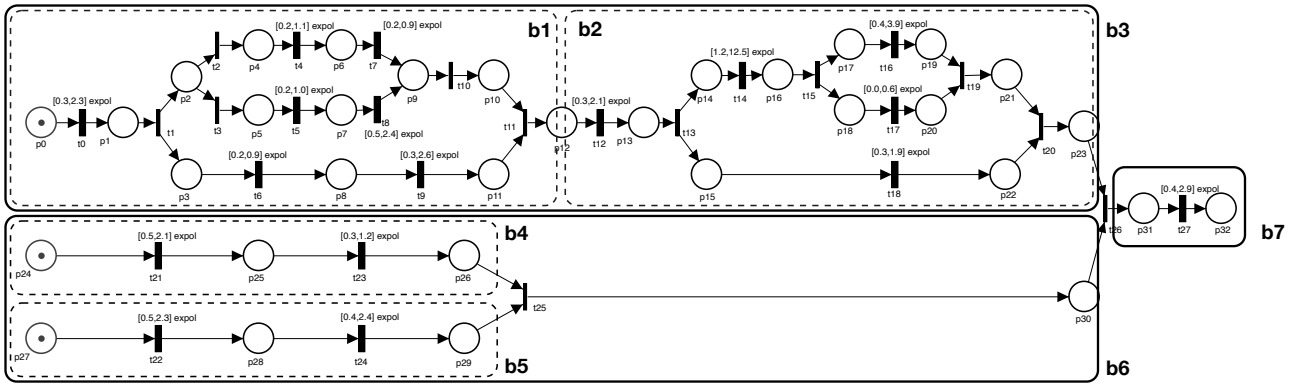

Figure 2: STPN model of a workflow consisting of 5 sequences, 3 AND-splits, 5 AND-joins, 1 XOR-split, and 1 XOR-join. The model is decomposed into the blocks b_3, b_6, and b_7 (tick-border boxes). In turn, b_3 and b_6 are derived as the composition of the blocks b_1, b_2 and b_4, b_5, respectively (dashed-border blocks). Times are expressed in s.

the workflow, e.g., the workflow of Fig. 2 has a single final activity modeled by transition `t27` with output place `p32`.

4.2 Stochastic parameters

We fit the duration distribution of each activity from a histogram of logged samples. As often done in statistical analysis of time series data, we first eliminate outliers, i.e., values that fall outside the overall pattern of the histogram and would distort the fitted distribution. To this end, we apply the IQR rule [42]. Specifically, given a sample vector $\mathbf{x}$ of durations, we discard values that are lower than $Q_1 - 1.5\,IQR$ or larger than $Q_3 + 1.5\,IQR$, where Q_1 is the first quartile, Q_3 is the third quartile, and $IQR := Q_3 - Q_1$ is the inter-quartile range. Let $\mathbf{y}$ be the obtained sample vector, where $\min\{\mathbf{y}\}$ is the minimum sample, $\max\{\mathbf{y}\}$ is the maximum sample, and $|\mathbf{y}|$ is the number of samples. Then, we derive the histogram $h : \mathcal{W} \to \mathbb{N}$ with support $[a, b]$ and N equal-width bins, where $a = \min\{\mathbf{y}\}$, $b = \max\{\mathbf{y}\}$, $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$ is the set of bins, and $h(w_n)$ is the number of elements of $\mathbf{x}$ falling in bin w_n $\forall\, n \in \{1, \ldots, N\}$. And, we compute the histogram $\bar{h} : \mathcal{W} \to \mathbb{R}$ representing relative frequencies of bins, i.e., $\bar{h}(w_n) = h(w_n)/(\omega \cdot \sum_{n=1}^{N} h(w_n))$.

To fit the obtained histogram $\bar{h}$ with a duration distribution having firmly bounded support, we extend the approximants of [47], which fit the sample mean and the sample variance with a shifted EXP distribution, a hypo-EXP distribution, an EXP distribution, or a hyper-EXP distribution depending on whether the sample coefficient of variation is lower than $1/\sqrt{2}$, between $1/\sqrt{2}$ and 1, nearly 1, or larger than 1, respectively. Given that the duration histograms of the WS-DREAM data set [51] considered in the experiments exhibit coefficient of variation lower than $1/\sqrt{2}$ (before and after outliers elimination), we extend the shifted EXP distribution of [47] into a shifted truncated EXP distribution. Specifically, we consider a PDF $f(x) = e^{-\lambda x}\, e^{\delta\lambda}/(1 - e^{-b+\delta})$ with support $[\delta, b]$, where δ and λ are selected by numerically solving a system of equations imposing that mean and variance of $f(x)$ are equal to the mean $\bar{\mu}$ and the variance $\bar{\sigma}$ of histogram $\bar{h}$, respectively:

$$\begin{cases} \dfrac{-1 - \delta\lambda + (1 + b\lambda)\, e^{(-b+\delta)\lambda}}{\lambda\,(-1 + e^{(-b+\delta)\lambda})} = \bar{\mu} \\ \dfrac{e^{2b\lambda} + e^{2\delta\lambda} - (2 + b^2\lambda^2 - 2b\delta\lambda^2 + \delta^2\lambda^2)\, e^{(b+\delta)ll}}{\lambda^2\,(e^{b\lambda} - e^{\delta\lambda})^2} = \bar{\sigma} \end{cases} \tag{1}$$

In a similar manner, shifted truncated EXP distributions could be used also to extend the approximants of [47] in the cases that the coefficient of variation of the observed data is larger than $1/\sqrt{2}$.

Note that associating each transition with a shifted truncated EXP distribution casts the resulting STPN model in the class of Deterministic and Stochastic Petri Nets (DSPNs) [29]. Notably, while the complexity of solution techniques developed for DSPNs limits applicability to models with at most one DET transition enabled in each state [28, 30], STPN models where multiple transitions with shifted truncated EXP distribution are concurrently enabled can be efficiently solved through regenerative transient analysis based on the method of stochastic state classes [23].

As an example of application, Fig. 3 plots the shifted truncated EXP approximant of a histogram of web service durations obtained from the WS-DREAM data set [51]. Specifically, the initial vector $\mathbf{x}$ contains 7467 samples between 0.2 s and 20 s, with $Q_1 = 0.5$ s, $Q_3 = 1.2$ s, and $IQR = 0.7$ s. The elimination of outliers yields a vector $\mathbf{y}$ consisting of 6427 samples between 0.2 s and 2.3 s, which is used to derive histogram $\bar{h}$. In turn, $\bar{h}$ has support $[a, b] = [0.2, 2.3]$ s, mean $\bar{\mu} = 0.7$ s, standard deviation $\bar{\sigma} = 0.4$ s, and coefficient of variation cv= 0.5. Finally, mean and standard deviation of $\bar{\sigma}$ are fitted with a shifted truncated EXP distribution with support $[\delta, b] = [0.3, 2.3]$ s and rate $\lambda = 2.4$.

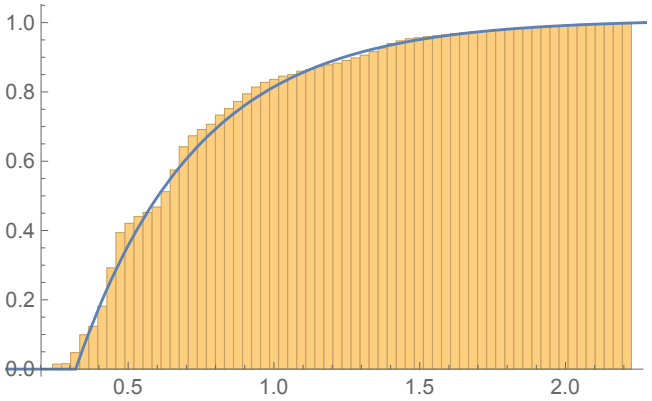

Figure 3: A shifted truncated EXP CDF that fits mean and variance of a histogram of web service durations obtained from the WS-DREAM data set [51]. The CDF is associated with transition `t0` in Fig. 2. Times are expressed in s.

4.3 Measure of interest

We evaluate the response time distribution $\Gamma(t)$ of the workflow, i.e., $\Gamma(t) := P\{\gamma \leq t\}$, where γ is the time needed to perform the overall workflow. If regenerative transient analysis of the STPN of the workflow can be afforded, $\Gamma(t)$ can be computed as the transient probability of the final place of the STPN:

$$\Gamma(t) = p_{m_{\text{fin}}}(t)\ \forall\, t \in [0, t_{\max}] \tag{2}$$

where m_{fin} is the marking that assigns one token to the final place and no token to any other place, and $p_{m_{\text{fin}}}(t)$ is the probability of marking m_{fin} at time t, computed as discussed in Section 3.2. For instance, in the example of Fig. 2, $m_{\text{fin}}(\texttt{p32}) = 1$ and $m_{\text{fin}}(\texttt{p}) = 0$ $\forall\, p \in P \setminus \{\texttt{p32}\}$, where P is the set of places of the STPN.

Conversely, if regenerative transient analysis is not viable, which occurs for complex workflows such as those considered in this paper, then a compositional approach becomes necessary to evaluate $\Gamma(t)$.

5 SOLUTION TECHNIQUE

In this section, we present a compositional approach to evaluate the response time distribution of complex workflows (Section 5.1), exploiting nondeterministic analysis to characterize the main factor of complexity of regenerative transient analysis (Section 5.2) and drive decomposition into a hierarchy of sub-workflows (Section 5.3). Then, we discuss the complexity of the approach (Section 5.4).

5.1 Overview

A workflow can be decomposed into a hierarchy of subworflows, exploiting the structure of its STPN model to identify submodels termed *blocks*, and using nondeterministic analysis to determine whether the obtained blocks are amenable to efficient regenerative transient analysis or need to be decomposed further. As recalled in Section 3.1, an STPN is a directed bipartite graph, where places and transitions represent nodes, pre-conditions and post-conditions account for directed arcs, and directed paths consist of alternating sequences of places and transitions, linked by pre-condition and post-condition relations. Based on the structure of concurrency described in Section 4.1, the STPN model of a workflow (or subworkflow) is a Directed Acyclic Graph (DAG), which can be explored starting from the places that contain a token, i.e., the input places of the GEN transitions modeling the initial activities of the workflow and the input places of the IMM transitions accounting for an XOR-split among alternative initial activities of the workflow.

Definition 1. A *block* of an STPN is a set of directed paths that have a common final place termed final place of the block. Similarly, the initial places of the paths are termed initial places of the block.

Definition 2. A *single-entry block* of an STPN is a block whose paths have a common initial place and a common final place, which are termed initial and final place of the block, respectively.

Definition 3. A *composite block* of an STPN is the composition of two single-entry blocks, either in *sequence* or in *parallel*.

- The sequence of two single-entry blocks b_1 and b_2 is a block such that the final place of b_1 is the initial place of b_2, so that the initial place of b_1 and the final place of b_2 are the initial place and the final place of the composite block, respectively.
- The parallel composition of two single-entry blocks b_1 and b_2 is a block such that the final place of both b_1 and b_2 is an input place of an IMM transition t having a single output place p, so that the initial places of b_1 and b_2 are the initial places of the composite block, while p is its final place.

For instance, in Fig. 2, b_1 is a single-entry block collecting paths ⟨p0, t0, p1, t1, p2, t2, p4, t4, p6, t7, p9, t10, p10, t11, p12⟩, ⟨p0, t0, p1, t1, p2, t3, p5, t5, p7, t8, p9, t10, p10, t11, p12⟩, and ⟨p0, t0, p1, t1, p3, t6, p8, t9, p11, t11, p12⟩; b_2 is also a single-entry block; and, b_3 is a composite block obtained as the series composition of b_1 and b_2. Conversely, b_6 is a composite block obtained as the parallel composition of the single-entry blocks b_4 and b_5.

The approach repeatedly performs a *top-down decomposition* of the STPN model of the workflow into a hierarchy of blocks that can be efficiently evaluated through regenerative transient analysis (see Sections 5.2 and Section 5.3 for details). The analysis of each block b yields the *numerical form* of the response time CDF $\Gamma_b(t)$ of the block, computed according to Eq. (2). Then, the approach repeatedly performs a *bottom-up recomposition* of the results of these separate analyses, combining the response time CDFs of pairs of blocks (either composed in sequence or in parallel) in order to evaluate the response time of the overall workflow.

- Given two blocks b_1 and b_2 with response time CDF $\Gamma_1(t)$ and $\Gamma_2(t)$, respectively, and response time PDF $\gamma_1(t)$ and $\gamma_2(t)$, respectively, the response time CDF $\Gamma_s(t)$ of the sequence of b_1 and b_2 is derived from the convolution of $\gamma_1(t)$ and $\gamma_2(t)$:

$$\Gamma_s(t) = \int_0^t \int_0^\tau \gamma_1(x)\,\gamma_2(\tau - x)\,dx\,dt \;\forall\, t \in [0, t_{\max}] \tag{3}$$

- Given two blocks b_1 and b_2 with response time CDF $\Gamma_1(t)$ and $\Gamma_2(t)$, respectively, the CDF $\Gamma_p(t)$ of the response time of the parallel composition of b_1 and b_2 through an AND-split&join pattern is the CDF of the maximum between the response times of b_1 and b_2, which is derived as the product of $\Gamma_1(t)$ and $\Gamma_2(t)$ given that the response times of b_1 and b_2 are independent random variables:

$$\Gamma_p(t) = \Gamma_1(t)\,\Gamma_2(t) \;\forall\, t \in [0, t_{\max}] \tag{4}$$

- Given two blocks b_1 and b_2 with response time CDF $\Gamma_1(t)$ and $\Gamma_2(t)$, respectively, the CDF $\Gamma_x(t)$ of the response time of the parallel composition of b_1 and b_2 through an XOR-split&join pattern with probabilities p_1 and $1 - p_1$, respectively, is derived as the weighted sum of $\Gamma_1(t)$ and $\Gamma_2(t)$:

$$\Gamma_x(t) = p_1\,\Gamma_1(t) + (1 - p_1)\,\Gamma_2(t) \;\forall\, t \in [0, t_{\max}] \tag{5}$$

For instance, the STPN model of Fig. 2 is decomposed into blocks b_3, b_6, and b_7, where b_3 is identified as the series of blocks b_1 and b_2, and b_6 as the parallel composition of blocks b_4 and b_5. Regenerative transient analysis of b_3 and b_6 yields the numerical form of their response time CDF $\Gamma_3(t)$ and $\Gamma_6(t)$, respectively. Then, the response time CDF $\Gamma_{3-6}(t)$ of their parallel composition is derived as the product of $\Gamma_3(t)$ and $\Gamma_6(t)$ according to Eq. (4). Given that b_7 contains the GEN transition t27 only, the numerical form of its response time CDF $\Gamma_7(t)$ is obtained by sampling the CDF $F_{\texttt{t27}}(t)$ of t27. Finally, the response time CDF $\Gamma(t)$ of the workflow is obtained through Eq. (3) from the convolution of the PDFs $\gamma_{3-6}(t)$ and $\gamma_7(t)$ corresponding to the CDFs $\Gamma_{3-6}(t)$ and $\Gamma_7(t)$, respectively.

5.2 Estimating the computational complexity of regenerative transient analysis of a block

As recalled in Section 3.2, regenerative transient analysis [23] of an STPN model consists of three steps: *i*) the enumeration of stochastic state classes; *ii*) the derivation of the local and the global kernels from the enumerated classes; and, *iii*) the solution of the

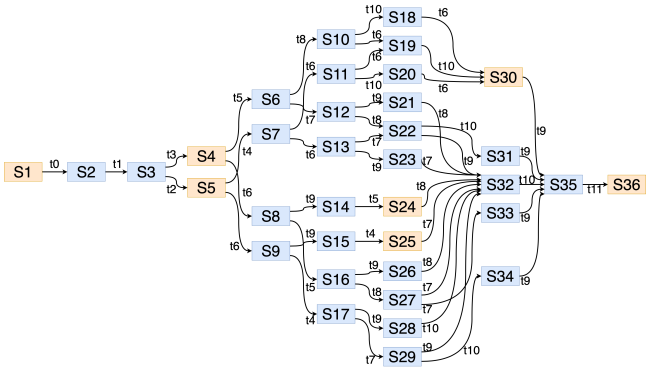

Figure 4: State class graph of block b_1 in Fig. 2. Regenerative state classes are highlighted in orange.

Markov renewal equations formulated in terms of the kernels. The complexity of each step mainly depends on the following factors.

(1) As illustrated by the experimental results reported in [38], the number of the enumerated stochastic state classes and the complexity of each of them (in terms of the number of DBM zones and expmonomial terms of the piecewise joint PDF) depend on the concurrency degree of the GEN transitions (i.e., the number of concurrently enabled GEN transitions), the number of firings to which a GEN transition remains persistent (i.e., the number of firings after which a transition remains enabled), and the number of expmonomial terms of the PDFs associated with the GEN transitions. In particular, the number of DBM zones is polynomial in the number of persistent transitions. Moreover, at each firing, the number of expmonomial terms of the joint PDF increases linearly with the polynomial degree. And, for the fired transition and for each disabled transition, the polynomial degree increases by one if the joint PDF contains no exponential factor.
(2) The kernels derivation and the solution of the Markov renewal equations have linear complexity in the number of stochastic state classes as well as in the number of DBM zones and expmonomial terms of the joint PDF of each class.
(3) The kernels derivation and the solution of the Markov renewal equations have linear and quadratic complexity, respectively, in the number of time points (i.e., the number of times that the time step is contained in the time limit).

Notably, the factors of complexity of regenerative transient analysis of an STPN can be characterized by inspecting the state class graph of the underlying TPN. In particular, based on the concurrency structure of a block, the number of stochastic state classes can be derived from the number of state classes according to a polynomial relation. Moreover, the concurrency degree of the GEN transitions can be derived from the number of transitions that are enabled in the state classes; the number of firings to which a GEN transition remains persistent can be computed as the length of paths such that the transition is persistent in each state class of the path; and, the number of DBM zones and the polynomial degree of the joint PDF of a stochastic state class can be derived from the sequence of firings that leads to the underlying state class. Overall, this characterization opens the way to the definition of *heuristics* that efficiently estimate the complexity of regenerative transient analysis of a block by exploiting nondeterministic analysis of the underlying TPN. Though multiple of the mentioned factors of complexity could be considered, including the level of DBM partitioning, the number of expmonomial terms, and polynomial degree of the joint PDFs, this strategy would yield conservative and thus unpractical criteria. Therefore, we rather resort to a heuristics that considers regenerative transient analysis of a block affordable if the maximum concurrency degree of the GEN transitions is not larger than a threshold D and the maximum number of consecutive firings from the initial state class is not larger than a threshold E.

For instance, Fig. 4 shows the state class graph of block b_1 depicted in Fig. 2. The state class graph consists of 36 state classes, including 7 regenerative state classes (highlighted in orange). The GEN transitions have maximum concurrency degree equal to 3 in state class S3, which has marking p2 p3 and 3 newly-enabled transitions (i.e.,t2, t3, and t6). The maximum number of consecutive firings from the initial state class is equal to 9. Note that, due to the structure of concurrency of the STPN model of a workflow, the state class graph of a block has a single terminal state class (i.e., S36).

5.3 Decomposition into blocks

The STPN of a workflow is repeatedly decomposed into blocks until the response time CDF of each block can be efficiently evaluated through regenerative transient analysis. Decomposition is performed based on the STPN structure, which is explored as a DAG to identify split and join patterns. Specifically, a block b to be decomposed is explored starting from each place of a set $P^\star$:

- If b is the overall worflow, then $P^\star$ initially collects the places of b that contain a token, i.e., the input places of the GEN transitions representing initial activities of the workflow and the input places of the IMM transitions modeling an XOR-split among initial activities of the workflow. For instance, the visit of the STPN of Fig. 2 starts with $P^\star = \{\texttt{p0}, \texttt{p24}, \texttt{p27}\}$.
- Otherwise, $P^\star$ initially consists of the initial place of b. For instance, the visit of block b_2 in Fig. 2 starts with $P^\star = \{\texttt{p12}\}$.

A visit of block b from a place $p \in P^\star$ can be performed according to the *join strategy* or the *split strategy*, which exploit an AND-join pattern or an AND-split pattern, respectively, to identify a new block of b, respectively. The two strategies operate as follows.

- **Join strategy.** In a visit of block b from place $p \in P^\star$ through the join strategy, a new block is identified as soon as one of the following two conditions becomes true:
  - **Condition 1**: a number of AND-join IMM transitions (i.e., IMM transitions with multiple input places) is visited after the same number of AND-split IMM transitions (i.e., IMM transitions with multiple output places).
    In this case, the last visited IMM transition t synchronizes the sub-workflows originated from the first visited AND-split construct, and thus the set of paths that start with place $p \in P^\star$ and end with the output place p' of t is a new single-entry block. Moreover, to let block b be explored beyond the newly identified block, p' is added to $P^\star$.
    For instance, in a visit of the STPN of Fig. 2 from place p0, the AND-join IMM transition t11 is visited after the AND-split IMM transition t1. Therefore, block b_1 is identified and the output place p12 of t11 is added to $P^\star$.
  - **Condition 2**: an AND-join IMM transition t is visited without having visited any AND-split IMM transition.
    In this case, t synchronizes independent sub-workflows, and thus the set of paths starting with place p and ending with the input place p' of t is a new single-entry block. If any other block synchronized by t has been identified, the effort needed to perform regenerative transient analysis of the parallel composition of all the blocks is estimated: if the analysis is affordable, then the set collecting the synchronized blocks, transition t, and its output place p'' is a new (analyzable) composite block, otherwise the synchronized blocks remain distinct. In both cases, p'' is added to $P^\star$ to explore b beyond the identified blocks.

For instance, in a visit of the STPN of Fig. 2 from place p24, the AND-join IMM transition t25 is visited without having visited any AND-split transition, and thus block b_4 is identified. When also block b_5 is identified (through a visit of the STPN from place p27), the parallel composition of b_4 and b_5 is identified as an analyzable composite block b_6 (since it has maximum degree of parallelism equal to 2). Moreover, the output place p30 of t25 is added to $P^\star$.

- **Split strategy.** In a visit of block b from place $p \in P^\star$ through the split strategy, multiple new blocks are identified as soon as an AND-split transition t is visited. By construction, t has a single input place p_i and multiple output places $p_{o_1}, \ldots, p_{o_n}$, used to identify the blocks connected by t.
  - First, the single-entry block that collects the paths starting with $p \in P^\star$ and ending with p_i is identified as a new block, unless the block contains an IMM transition only (which occurs if p is the input place of t, i.e., $p = p_i$).
  - Then, for each place $p_o \in \{p_{o_1}, \ldots, p_{o_n}\}$, a visit of block b is performed from p_o until the AND-join IMM transition t' that synchronizes the sub-workflows originated from t, identifying the single-entry block that starts with p_o and ends with the input place of t' that has just been visited.
  - Finally, the output place of t' is added to $P^\star$ to explore block b beyond the identified blocks.

  For instance, if the analysis of block b_1 in Fig. 2 were not viable, then a visit of b_1 from its initial place p0 according to the split strategy would stop on the AND-split IMM transition t1, identifying the single entry-block with input place p0 and output place p1. Then, a visit of b_1 from places p2 and p3 would identify the single entry-block collecting the paths starting with p2 and ending with p10 and the single entry-block collecting the paths starting with p3 and ending with p11, respectively. Finally, p12 would be added to $P^\star$.

In both strategies, when the considered block b has been visited starting from each place of $P^\star$, then the effort needed to perform regenerative transient analysis of sequences of two identified blocks is evaluated in order to identify possible analyzable composite blocks. For instance, the concurrency degree among the GEN transitions is equal to 2 and 3 for blocks b_1 and b_2 in Fig. 2, respectively, so that their sequential composition has concurrency degree 3. Moreover, the number of consecutive firings starting from the initial state class is equal to 9, 7, and 16 for b_1, b_2, and their sequential composition, respectively. Assuming thresholds $D = 3$ and $E = 20$, the sequence of b_1 and b_2 is considered as an analyzable block.

We consider a *heuristics* that alternatively applies the join strategy and the split strategy during the decomposition into blocks: Specifically, the STPN of the workflow is decomposed according to the join strategy. Then, if any of the identified blocks is too complex to be analyzed, then, by construction, the block includes (at least) an AND-split transition that boosts parallelism, which can be exploited to reduce complexity by decomposing the block according to the split strategy. If any of the identified blocks is not analyzable, it is decomposed according to the join strategy, and so on. In so doing, the join strategy identifies behaviors that aggregate multiple regeneration epochs, while the split strategy separates behaviors of the same regeneration epoch that have different firing sequence.

In both strategies, if the path from the initial place of the block to the first AND-join or AND-split transition contains unmerged XOR-split transitions, then the portions of the model identified by the XOR-split and the XOR-join transitions are considered as separate blocks. For instance, in Fig. 2, if b_1 could not efficiently be analyzed, its decomposition would yield 4 blocks containing transition t0 and the series of transitions t4,t7 and t5,t8 and t6,t9.

5.4 Computational complexity

The solution technique performs tree steps: *i*) top-down decomposition of the STPN model of the whole workflow into blocks; *ii*) regenerative transient analysis of each block; and, *iii*) bottom-up recomposition of the results of these separate analyses. Specifically:

(1) The decomposition of the STPN model of a workflow into blocks requires to explore the model structure and to perform nondeterministic analysis of all the identified blocks, as discussed in Sections 5.2 and 5.3. On the one hand, the complexity of a visit of the model through the join and the split strategies requires linear complexity in the number of places and transitions, which is relatively small even for complex workflows. On the other hand, nondeterministic analysis of all the identified blocks can be performed very efficiently. Therefore, the complexity of the decomposition of the STPN of a workflow into blocks turns out to be very limited.

(2) As discussed in Section 5.2, the complexity of regenerative transient analysis of a block mainly depends on the maximum concurrency degree of the GEN transitions, the maximum number of firings to which a GEN transition remains persistent, and the number of time points where the response time CDF of the block is computed. While the impact of the first two factors is limited by the heuristics that estimates whether the analysis of a block is affordable, the number of time points remains the main (quadratic) factor of complexity of regenerative transient analysis.

(3) The recomposition of the analysis results consists in performing sums/products of the response time CDFs and convolutions of the response time PDFs of pairs of blocks, yielding linear and quadratic complexity, respectively, in the number of time points used to represent CDFs in numerical form.

According to this, the number of time points comprises the main (quadratic) factor of complexity of the overall approach.

6 CASE STUDY

In this section, we demonstrate the approach with reference to the context of Service-Oriented Architectures (SOA), where the workflow abstraction naturally captures the concept of an application built by composition of independent, self-contained, loosely coupled services [48]. In this context, time behavior is a key figure of the QoS, driving various stages of development and operation, including early evaluation of design or deployment choices [3, 32], dynamic selection and compositions adapted to runtime conditions [11] or to a specific user profile [49], or optimized within a multi-objective QoS model [3], or cast within a problem of quantitative verification supporting service selection and resource provisioning [10].

In particular, we experiment the approach on the *TravelPlan* composite service example [11], which we extend in complexity by considering a finer granularity of composed services (Section 6.1) with the twofold aim of demonstrating scalability of the proposed approach and advocating its suitability in the ongoing evolution towards finer-grained composition schemes promoted by the emergence of microservice and RESTful architectures [22] (Section 6.2). To this end, we consider a variety of durations selected from the WS-DREAM data set [51], widely used in the literature on service oriented computing, and we test different combinations of durations of individual services to assess accuracy and complexity of results with respect to different timings. Moreover, we rely on the assumption that individual services are horizontally scaled, which allow to evaluate the response time of the composite web service, without concerning about multiple requests and queue effects.

Experiments have been performed on a single core of an Intel(R) Xeon(R) Gold 5120 CPU 2.20 GHz equipped with 32.0 GB RAM.

6.1 Model of a composite web service

6.1.1 Structure of concurrency. The TravelPlan process presented in [11] is a composite web service aimed to plan a travel, providing a solution that includes choices for flights to a given city, hotels near a given attraction, and transports from the airport to the hotel (either cab or shuttle depending on the arrival time of the flight and the latest possible hotel check-in) and from the hotel to the attraction (either car rental or metro depending on the distance of the hotel from the attraction). Fig. 5 shows the STPN model of an extension of the TravelPlan process, where the granularity of the composed services is increased including different entity searching services that operate with different filtering options (e.g., searching both the cheapest and the best ranked hotels, modeled by transitions `getCheapestHotels` and `getBestRankedHotels`, respectively), and geographical information retrieval services (e.g., getting the position of metro stops, modeled by transition `getMetroStops`) used by other services (e.g., getting hotels near a metro stop close to an attraction, modeled by transitions `getHotelsNearStop` and `getAttractionsNearMetroStops`, respectively).

6.1.2 Stochastic parameters. The execution times of the activities of the workflow of Fig. 5 are obtained from the WS-DREAM data set, which collects the response times of 4500 web services, invoked by 142 users in 64 different time slices, for a total amount of 40 896 000 available durations. We consider data related to 100 services, independently of the user and the time slice, and we derive a duration histogram for each service. To this end, as illustrated in Section 4.2, outliers are discarded according to the IQR rule [42], rejecting values lower than $Q_1 - 1.5\,IQR$ or larger than $Q_3 + 1.5\,IQR$, where Q_1, Q_3, and IQR are the first quartile, the third quartile, and their inter-quartile range, respectively. For each service, the remaining samples are collected in a 64-bin histogram.

Table 1 shows the statistics of the obtained histograms. As typical of web service response times, histograms have tight support, with width ranging between 0.37 s and 10.61 s, and equal to 1.75 s on average. The lower bound a of the support is comprised between

Table 1: Average value avg, standard deviation SD, coefficient of variation CV, minimum value min, and maximum value max of the expected value μ, the standard deviation σ, the coefficient of variation σ/μ, the support lower bound a, the support upper bound b, and the support width $b-a$ of 100 histograms of web service durations obtained from the WS-DREAM data set [51] through the approach of Section 4.2.

	avg	SD	CV	min	max
μ	0.525 s	0.842 s	1.605	0.054 s	7.799 s
σ	0.101 s	0.303 s	2.988	0.004 s	2.441 s
σ/μ	0.179	0.137	0.766	0.016	0.574
a	0.157 s	0.239 s	1.522	0 s	1.943 s
b	1.901 s	1.722 s	0.904	0.372 s	12.555 s
$b-a$	1.749 s	1.561 s	0.892	0.372 s	10.612 s

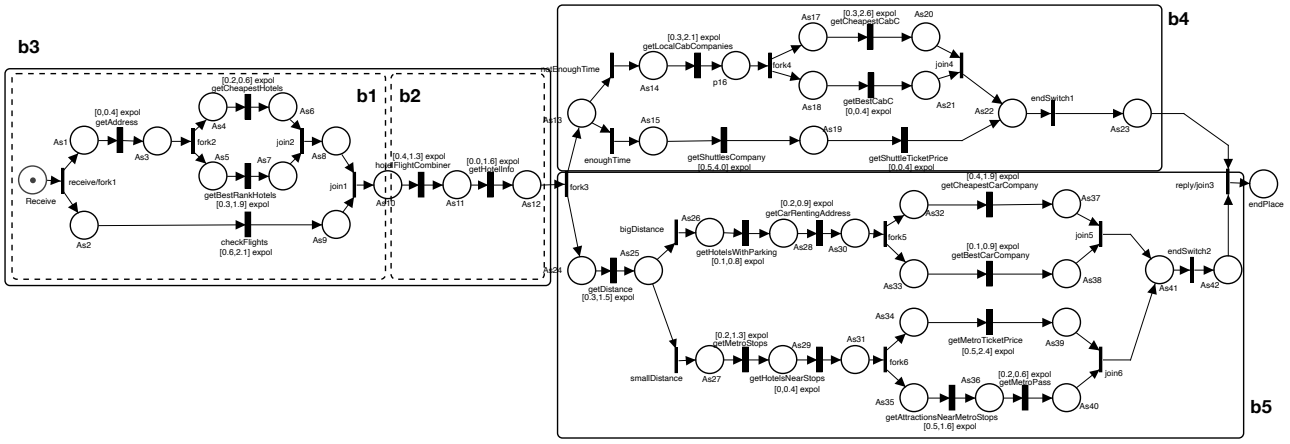

Figure 5: STPN model of an extended version of the TravelPlan composite web service [11]. The model is decomposed into the analyzable blocks b_3, b_4, and b_5 (thick border boxes), where b_3 is derived as the composition of the smaller blocks b_1 and b_2 (dashed border boxes). Timings, expressed in s, are those of the experiment with best accuracy.

0 s and 1.94 s, with expected value equal to 0.16 s and low standard deviation equal to 0.24 s. The upper bound b of the support takes values in a larger interval between 0.37 s and 12.56 s, with expected value equal to 1.90 s and low standard deviation equal to 1.72 s. On average, sample durations have expected value $\mu = 0.53$ s, standard deviation $\sigma = 0.10$ s, and coefficient ot variation equal to 0.18. Note that the maximum value of the coefficient of variation is equal to $0.57 < 1/\sqrt{2}$, enabling fitting of each histogram through a shifted truncated EXP distribution, as discussed in Section 4.2.

6.2 Experimental results

Experimentation aims at evaluating the accuracy and the computational complexity of the proposed approach with respect to specific duration distributions associated with the workflow activities. To this end, the approach is repeatedly applied to evaluate the response time distribution of 1000 workflows whose STPN model has the concurrency structure of Fig. 5 and is made of GEN transitions whose duration distributions are randomly selected among 100 shifted truncated EXP distributions fitting mean and variance of web service response time histograms obtained from the WS-DREAM data set [51]. IMM transitions of the STPN have weight 1, so that alternative behaviors of XOR-split patterns are equiprobable. In the step of model decomposition, the complexity of regenerative transient analysis of a block is estimated assuming a threshold $D = 3$ on the maximum concurrency degree of the GEN transitions and a threshold $E = 10$ on the maximum number of consecutive firings from the initial state class, decomposing the model into blocks b_3, b_4, and b_5 illustrated in Fig. 5. The overall analysis process is repeated for different values of the time tick equal to 0.1 s, 0.05 s, 0.025 s, and 0.0125 s. At each repetition, the time limit used by regenerative transient analysis of each block and by numerical computations performed to recompose the results of these separate analyses is computed based on the supports of the workflow activities.

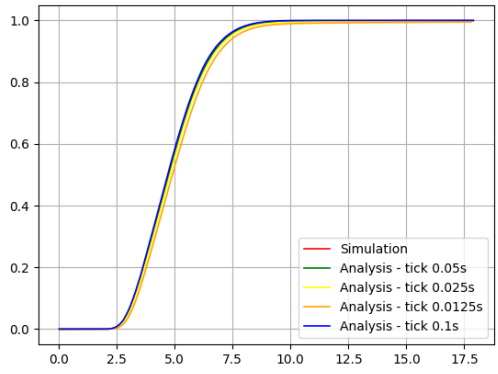

Figure 6: Response time CDFs of the workflow of Fig. 5 derived by the proposed analysis process with time tick 0.1 s, 0.05 s, 0.025 s, **and** 0.0125 s **and by simulation with tick** 0.0125 s.

Table 2: Average value avg, standard deviation SD, coefficient of variation CV, minimum value min, and maximum value max of the JS divergence of the obtained results from the simulation results, for different time tick values.

tick	avg	SD	CV	min	max
0.1 s	0.0169	0.0070	0.4151	0.0065	0.0533
0.05 s	0.0060	0.0013	0.2215	0.0037	0.0140
0.025 s	0.0034	0.0007	0.1954	0.0021	0.0061
0.0125 s	0.0028	0.0008	0.3000	0.0014	0.0056

The response time CDF of each workflow evaluated by the approach is compared with a ground truth obtained by performing stochastic simulation of the workflow STPN, using the Jensen-Shannon (JS) divergence [27, 33] to determine both the number of simulation runs and the accuracy of the analysis results. Specifically, the JS is a symmetric measure evaluating the discrepancy between two random variables defined over the same probability space. Specifically, given the PDFs f_a and f_s of the workflow response time computed by the proposed approach and by simulation, respectively, obtained by derivation of the corresponding CDFs, the JS divergence $D_{JS}(f_a \,||\, f_s)$ of f_a from f_s is defined as:

$$D_{JS}(f_a \,||\, f_s) = \frac{1}{2} D_{KL}(f_a \,||\, Z) + \frac{1}{2} D_{KL}(f_s \,||\, Z), \tag{6}$$

where $Z(t) = \frac{1}{2}(f_a(t) + f_s(t)) \; \forall\, t \in \Omega$ is the random variable that averages the input variables, Ω is a set of equidistant time points covering the support of f_a and f_s, and $D_{KL}(\cdot \,||\, \cdot)$ is the Kullback-Leibler divergence(KL) [27, 33] defined as

$$D_{KL}(f_a \,||\, f_s) = \sum_{t \in \Omega} f_a(t) \cdot \log\left(\frac{f_s(t)}{f_a(t)}\right). \tag{7}$$

Note that the symmetry property holds for the JS divergence, i.e., $D_{JS}(f_a \,||\, f_s) = D_{JS}(f_s \,||\, f_a)$, but not for the KL divergence, i.e., $D_{KL}(f_a \,||\, f_s) \neq D_{KL}(f_s \,||\, f_a)$. According to this, $D_{JS}(\cdot \,||\, \cdot)$ turns out to be a symmetrized and smoothed version of $D_{KL}(\cdot \,||\, \cdot)$.

To derive the ground truth, 5000, 10 000, 15 000, ..., 50 000 simulation runs are performed with time tick equal to 0.0125 s, computing the JS divergence of the workflow response time CDF evaluated by the 5000-run, ..., 45 000-run simulation with respect to the 50 000-run simulation. On average, the JS divergence is nearly equal to 0.0157 for the 5000-run simulation and converges to 0.0043 for the

Table 3: Average value avg, standard deviation SD, coefficient of variation CV, minimum value min, and maximum value max of the computation times of the proposed approach and of simulation, for different time tick values.

Analysis times					
tick	**avg**	**SD**	**CV**	**min**	**max**
0.1 s	0.1795 s	0.0670 s	0.3730	0.0320 s	0.5210 s
0.05 s	0.3883 s	0.1533 s	0.3949	0.0780 s	1.1260 s
0.025 s	0.9393 s	0.4375 s	0.4358	0.1820 s	2.8170 s
0.0125 s	3.1536 s	2.4648 s	0.7816	0.5450 s	17.0310 s
Simulation times					
tick	**avg**	**SD**	**CV**	**min**	**max**
0.0125 s	48.9412 s	2.4869 s	0.0508	44.1120 s	58.1200 s

45 000-run simulation, showing that the results of the 50 000-run simulation can be considered as the ground truth.

Table 2 shows, for each time tick value, the statistic of the JS divergence of the analysis results from the ground truth, computed over the mentioned 1000 experiments. As expected, the minimum, maximum, average value, and standard deviation decrease with the time tick, which is the only parameter that introduces approximation error in results, both in sub-workflow analysis and in numerical recomposition. Results are accurate already with the most coarse-grained time tick 0.1 s, with divergence value not larger than 0.0533 and equal to 0.0169 on average. As the time tick decreases nearly by one order of magnitude, also the divergence value decreases by one order of magnitude, being not larger than 0.0056 and equal to 0.0028 on average for the most fine-grained time tick 0.0125 s. This trend is also evident from the plots of the response time CDFs shown in Fig. 6, where the curves computed by the proposed approach rapidly converge to the simulation curve as the time step decreases.

Table 3 shows the statistics of the observed computations times. Specifically, as the time tick halves, the average computation time of the proposed approach increases at least by a factor of 2, being 0.1795 s for the most coarse-grained time tick 0.1 s and 3.1536 s for the most fine-grained time tick 0.0125 s. The same trend is observed for the standard deviation, minimum, and maximum, showing that the approach is able to efficiently evaluate the response time CDF of complex workflows. In particular, for the most fine-grained time tick 0.0125 s, the computation time is lower than the simulation time by more than an order of magnitude on average, and by more than a factor of 3 in the worst case. Note that simulation is facilitated by the assumption of shifted truncated EXP distributions, which can be efficiently sampled by the inverse transformation method.

To fairly compare analysis with simulation, we consider the analysis with time tick 0.0125 s, requiring 3.1536 s on average, and the 5000-run simulation with the same time tick, requiring 4.499 s. Results show that, with comparable computation time, the analysis accuracy is on average one order of magnitude better than that of simulation: in fact, the JS divergence from the ground truth is equal to 0.0157 for the 5000-run simulation, while, for the analysis, it is equal to 0.0028 on average and to 0.0056 in the worst case.

7 CONCLUSIONS

We have presented an end-to-end compositional approach for the evaluation of the response time CDF of complex workflows starting from samples of the execution times of individual activities. To this end, the workflow is represented through STPNs, associating activity execution times with shifted truncated EXP distributions that fit mean and standard deviation of real logged histograms. Then, the workflow is decomposed into a hierarchy of subworkflows that can be efficiently analyzed through regenerative transient analysis based on the method of stochastic state classes, using the state class graph of the underlying TPNs to characterize the main factors of complexity of regenerative transient analysis and thus to drive the level of decomposition. Finally, the execution time CDFs of the identified subworkflows, computed through separate analyses, are repeatedly recomposed in numerical form to derive the response time CDF of the overall workflow.

Experiments address the quantitative evaluation of the response time of a composite web service of the literature, extended in complexity, to illustrate the scalability of the approach with respect to finer-grained composition schemes, and associated with a variety of durations randomly selected from a data set in the literature of service oriented computing, to assess variability of results with respect to specific timings. The obtained results show that the approach achieves high accuracy and good performance with respect to a ground truth estimated through stochastic simulation, as well as with respect to a simulation with comparable computation time.

The proposed approach is open to various extensions. On the one hand, the model expressivity could be extended to include non-free choice constructs, considering loops and other control-flow patterns that break the well-formed structure considered in this paper. On the other hand, any distribution in the class of expolynomial functions could be used to fit the observed duration histograms, with bounded or unbouded support, with a unique analytical form over the entire domain or piecewise defined. In particular, shifted truncated EXP distributions could be used to extend the approximants of [47] in the cases that the coefficient of variation of the observed data is larger than $1/\sqrt{2}$. Moreover, the approach could be applied in a variety of relevant contexts that go beyond the specific domain of software services considered in this experimentation, e.g., supply chain management, business processes, physical processes.

REFERENCES

[1] Ruth Sara Aguilar-Saven. 2004. Business process modelling: Review and framework. *Int. Journal of production economics* 90, 2 (2004), 129–149.

[2] Elvio Gilberto Amparore, Gianfranco Balbo, Marco Beccuti, Susanna Donatelli, and Giuliana Franceschinis. 2016. 30 years of GreatSPN. In *Principles of Performance and Reliability Modeling and Evaluation*. Springer, 227–254.

[3] Danilo Ardagna and Barbara Pernici. 2007. Adaptive service composition in flexible processes. *IEEE Trans. on Software Engineering* 33, 6 (2007), 369–384.

[4] Florian Arnold, Holger Hermanns, Reza Pulungan, and Mariëlle Stoelinga. 2014. Time-dependent analysis of attacks. In *Proc. Int. Conf. on Principles of Security and Trust*. Springer, 285–305.

[5] Gerd Behrmann, Alexandre David, Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. 2011. Developing Uppaal over 15 years. *Softw. Pract. Exper.* 41, 2 (Feb. 2011), 133–142.

[6] Bernard Berthomieu, P.-O. Ribet, and François Vernadat. 2004. The tool TINA – construction of abstract state spaces for Petri Nets and Time Petri Nets. *International Journal of Production Research* 42, 14 (2004).

[7] Marco Biagi, Laura Carnevali, Marco Paolieri, Tommaso Papini, and Enrico Vicario. 2017. Exploiting Non-deterministic Analysis in the Integration of Transient Solution Techniques for Markov Regenerative Processes. In *Proc. Int. Conf. on Quantitative Evaluation of Systems*. Springer, 20–35.

[8] Andrea Bobbio, András Horváth, and Miklós Telek. 2005. Matching three moments with minimal acyclic phase type distributions. *Stochastic models* 21, 2-3 (2005), 303–326.

[9] Andrea Bobbio and Miklos Telek. 1995. Markov regenerative SPN with non-overlapping activity cycles. In *Proc. Int. Comput. Perf. and Depend. Symp.* 124–133.

[10] Radu Calinescu, Lars Grunske, Marta Kwiatkowska, Raffaela Mirandola, and Giordano Tamburrelli. 2010. Dynamic QoS management and optimization in service-based systems. *IEEE Trans. on Software Engineering* 37, 3 (2010), 387–409.

[11] Gerardo Canfora, Massimiliano Di Penta, Raffaele Esposito, and Maria Luisa Villani. 2005. QoS-aware replanning of composite web services. In *Proc. IEEE Int. Conf. on Web Services*. IEEE, 121–129.

[12] Laura Carnevali, Leonardo Grassi, and Enrico Vicario. 2009. State-density functions over DBM domains in the analysis of non-Markovian models. *IEEE Transactions on Software Engineering* 35, 2 (2009), 178–194.

[13] Laura Carnevali, Lorenzo Ridi, and Enrico Vicario. 2009. Stochastic Fault Trees for cross-layer power management of WSN monitoring systems. In *Proc. Int. Conf. on Emerging Technologies & Factory Automation*. IEEE, 1–8.

[14] Hoon Choi, Vidyadhar G Kulkarni, and Kishor S Trivedi. 1994. Markov regenerative stochastic Petri nets. *Performance evaluation* 20, 1-3 (1994), 337–357.

[15] Francisco Curbera, Yaron Goland, Johannes Klein, Frank Leymann, Dieter Roller, Satish Thatte, and Sanjiva Weerawarana. 2002. Business process execution language for web services.

[16] Conrado Daws, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. 1995. The Tool KRONOS. In *Hybrid systems III*. 1066, Springer.

[17] Ton G de Kok and Jan C Fransoo. 2003. Planning supply chain operations: definition and comparison of planning concepts. *Handbooks in operations research and management science* 11 (2003), 597–675.

[18] David L Dill. 1989. Timing assumptions and verification of finite-state concurrent systems. In *Proc. Int. Conf. on Computer Aided Verification*. Springer, 197–212.

[19] Jean-Michel Fourneau and Nihal Pekergin. 2015. A numerical analysis of dynamic fault trees based on stochastic bounds. In *Proc. Int. Conf. on Quantitative Evaluation of Systems*. Springer, 176–191.

[20] G. Gardey, D. Lime, M. Magnin, and O.(H.) Roux. 2005. Roméo: a tool for analyzing Time Petri Nets. *CAV'05* (2005).

[21] Reinhard German and Christoph Lindemann. 1994. Analysis of stochastic Petri nets by the method of supplementary variables. *Perf. Eval.* 20, 1-3 (1994), 317–335.

[22] Robert Heinrich, André van Hoorn, Holger Knoche, Fei Li, Lucy Ellen Lwakatare, Claus Pahl, Stefan Schulte, and Johannes Wettinger. 2017. Performance engineering for microservices: research challenges and directions. In *Proc. of the 8th ACM/SPEC on Int. Conf. on Performance Engineering Companion*. 223–226.

[23] András Horváth, Marco Paolieri, Lorenzo Ridi, and Enrico Vicario. 2012. Transient analysis of non-Markovian models using stochastic state classes. *Performance Evaluation* 69, 7-8 (2012), 315–335.

[24] András Horváth and Miklós Telek. 2002. PhFit: A General Phase-Type Fitting Tool. In *Proc. Int. Conf. on Comput. Perf. Eval., Modelling Tech. and Tools*. 82–91.

[25] V. Kulkarni. 1995. *Modeling and analysis of stochastic systems*. Chapman & Hall. http://www.crcpress.com/product/isbn/9781439808757

[26] H Dharma Kwon, Steven A Lippman, and Christopher S Tang. 2010. Optimal time-based and cost-based coordinated project contracts with unobservable work rates. *Int. Journal of Production Economics* 126, 2 (2010), 247–254.

[27] Jianhua Lin. 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information theory* 37, 1 (1991), 145–151.

[28] Christoph Lindemann. 1995. DSPNexpress: a software package for the efficient solution of deterministic and stochastic Petri nets. *Perf. Eval.* 22, 1 (1995), 3–21.

[29] Christoph Lindemann. 1998. Performance modelling with deterministic and stochastic Petri nets. *ACM SIGMETRICS Perf. Eval. Review* 26, 2 (1998), 3.

[30] Christoph Lindemann and Axel Thümmler. 1999. Transient analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. *Perf. Eval.* 36 (1999), 35–54.

[31] Yanjie Liu, Zheng Zheng, and Jiantao Zhang. 2019. Markov Model of Web Services for Their Performance Based on Phase-Type Expansion. In *Proc. DASC-PICOM-CBDCOM-CYBERSCITECH*. IEEE, 699–704.

[32] Daniel A Menascé. 2004. Response-time analysis of composite Web services. *IEEE Internet computing* 8, 1 (2004), 90–92.

[33] Frank Nielsen. 2019. On a generalization of the Jensen-Shannon divergence and the JS-symmetrization of distances relying on abstract means. *arXiv preprint arXiv:1904.04017* (2019).

[34] Marco Paolieri, Marco Biagi, Laura Carnevali, and Enrico Vicario. to appear. The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems. *IEEE Transactions on Software Engineering* (to appear).

[35] Philipp Reinecke, Tilman Krauß, and Katinka Wolter. 2012. Cluster-based fitting of phase-type distributions to empirical data. *Computers & Mathematics with Applications* 64, 12 (2012), 3840–3851.

[36] Philipp Reinecke, Tilman Krauß, and Katinka Wolter. 2013. Phase-Type Fitting Using HyperStar. In *Proc. Europ. Perf. Eng. Workshop*. 164–175.

[37] Andreas Rogge-Solti, Wil MP van der Aalst, and Mathias Weske. 2013. Discovering stochastic petri nets with arbitrary delay distributions from event logs. In *Proc. Int. Conf. on Business Process Management*. Springer, 15–27.

[38] Luigi Sassoli and Enrico Vicario. 2007. Close form derivation of state-density functions over DBM domains in the analysis of non-Markovian models. In *Proc. Int. Conf. on Quantitative Evaluation of Systems*. IEEE, 59–68.

[39] SIRIO Library. 2020. https://github.com/oris-tool/sirio.

[40] Miklós Telek and András Horváth. 2001. Transient analysis of Age-MRSPNs by the method of supplementary variables. *Perf. Eval.* 45, 4 (2001), 205–221.

[41] Kisho S Trivedi and Robin Sahner. 2009. SHARPE at the age of twenty two. *ACM SIGMETRICS Performance Evaluation Review* 36, 4 (2009), 52–57.

[42] John W Tukey. 1977. *Exploratory data analysis*. Vol. 2. Reading, MA.

[43] Wil Van Der Aalst, Kees Max Van Hee, and Kees van Hee. 2004. *Workflow management: models, methods, and systems*. MIT press.

[44] Wil MP van Der Aalst, Arthur HM Ter Hofstede, Bartek Kiepuszewski, and Alistair P Barros. 2003. Workflow patterns. *Dist.&paral. datab.* 14, 1 (2003), 5–51.

[45] Enrico Vicario. 2001. Static analysis and dynamic steering of time-dependent systems. *IEEE transactions on software engineering* 27, 8 (2001), 728–748.

[46] Changzhou Wang, Guijun Wang, Haiqin Wang, Alice Chen, and Rodolfo Santiago. 2006. Quality of service (QoS) contract specification, establishment, and monitoring for service level management. In *Proc. IEEE Int. Enterprise Distributed Object Computing Conference Workshops*. IEEE, 49–49.

[47] Ward Whitt. 1982. Approximating a point process by a renewal process, I: Two basic methods. *Operations Research* 30, 1 (1982), 125–147.

[48] Elyas Ben Hadj Yahia, Laurent Réveillere, Yérom-David Bromberg, Raphaël Chevalier, and Alain Cadot. 2016. Medley: An event-driven lightweight platform for service composition. In *Int. Conf. on Web Engineering*. Springer, 3–20.

[49] Yilei Zhang, Zibin Zheng, and Michael R Lyu. 2011. WSPred: A time-aware personalized QoS prediction framework for Web services. In *IEEE Int. Symp. on Software Reliability Engineering*. IEEE, 210–219.

[50] Zheng Zheng, Kishor S Trivedi, Kun Qiu, and Ruofan Xia. 2015. Semi-markov models of composite web services for their performance, reliability and bottlenecks. *IEEE Transactions on services computing* 10, 3 (2015), 448–460.

[51] Zibin Zheng and M. R. Lyu. 2008. WS-DREAM: A distributed reliability assessment Mechanism for Web Services. In *Proc. IEEE Int. Conf. on Dependable Systems and Networks With FTCS and DCC*. 392–397.

[52] Armin Zimmermann. 2017. Modelling and performance evaluation with TimeNET 4.4. In *Int. Conf. on Quantitative Eval. of Systems*. Springer, 300–303.

APPENDIX: STOCHASTIC TIME PETRI NETS

An STPN is a tuple $\langle P, T, A^-, A^+, EFT, LFT, F, W\rangle$: P and T are disjoint sets of places and transitions, respectively; $A^- \subseteq P \times T$ and $A^+ \subseteq T \times P$ are sets of pre-condition and post-condition relations, respectively; EFT and LFT associate each transition $t \in T$ with an earliest firing time $EFT(t) \in \mathbb{Q}_{\geqslant 0}$ and a latest firing time $LFT(t) \in \mathbb{Q}_{\geqslant 0} \cup \{\infty\}$ such that $EFT(t) \leq LFT(t)$; F associates each transition $t \in T$ with a CDF F_t for its duration $\tau(t) \in [EFT(t), LFT(t)]$, i.e., $F_t(x) = P\{\tau(t) \leq x\}$, with $F_t(x) = 0$ for $x < EFT(t)$ and $F_t(x) = 1$ for $x > LFT(t)$; W associates each transition $t \in T$ with a weight $W(t) \in \mathbb{R}_{\geqslant 0}$. If omitted, we assume $W(t) = 1 \; \forall \, t \in T$.

A place p is termed *input* or *output* place for a transition t if $(p, t) \in A^-$ or $(t, p) \in A^+$, respectively. A transition t is termed *immediate* (IMM) if $EFT(t) = LFT(t) = 0$ and *timed* otherwise. A timed transition is termed *exponential* (EXP) if $F_t(x) = 1 - e^{-\lambda x}$ for some rate $\lambda \in \mathbb{R}_{>0}$, or *general* (GEN) if F_t is a non-EXP distribution. A GEN transition t is termed *deterministic* (DET) if $EFT(t) = LFT(t) > 0$. For each transition t with $EFT(t) < LFT(t)$, we assume that F_t can be expressed as the integral function of a PDF f_t, i.e., $F_t(x) = \int_0^x f_t(y)\, dy$. Similarly, an IMM or DET transition $t \in T$ is associated with the generalized distribution of a Dirac delta function $f_t(y) = \delta(y - \overline{y})$ with $\overline{y} = EFT(t) = LFT(t)$.

A transition t is *enabled* by a marking $m \in \mathcal{M}$ if m assigns at least one token to each of its input places, i.e., $m(p) > 0 \; \forall \, p \mid \langle p, t\rangle \in A^-$. The state of an STPN is a pair $\langle m, \vec{\tau}\rangle$ where $m \in \mathcal{M}$ is a marking, $E(m)$ is the set of transitions enabled by m, and $\vec{\tau}$ is a vector assigning a *time to fire* $\vec{\tau}(t) \in \mathbb{R}_{\geqslant 0}$ to each enabled transition $t \in E(m)$. A transition t is *firable* in a state $s = \langle m, \vec{\tau}\rangle$ if it is enabled by m and has minimum time to fire. A transition t that is firable in s is selected to fire with probability $p_t = W(t)/(\sum_{u \in E_{\min}} W(u))$, where $E_{\min}$ is the set of transitions that are firable in s.

The firing of transition t in state $s_1 = \langle m_1, \vec{\tau}_1\rangle$ yields a new state $s_2 = \langle m_2, \vec{\tau}_2\rangle$, where: *i)* m_2 is derived from m_1 by (1) removing a token from each input place of t and (2) adding a token to each output place of t; *ii)* $\vec{\tau}_2$ is obtained from $\vec{\tau}_1$ by sampling the time to fire of each *new-enabled* transition t' according to distribution $F_{t'}$, i.e., $\vec{\tau}_2(t') \sim F_{t'}$, and reducing the time to fire of each *persistent* transition t' by the sojourn time in m, i.e., $\vec{\tau}_2(t') = \vec{\tau}(t') - \vec{\tau}(t)$, where a transition t' enabled by m_2 is termed persistent if it is distinct from t and enabled by m_1 and by the intermediate markings after steps (1) and (2), and it is termed newly-enabled otherwise.