A Brief Study on Stochastic Petri Net

(CS 590 final project – small WCP test report)

Yue Ben (Cindy)

2013 spring

Stochastic Petri nets (SPNs) are extended Petri nets where the transitions fire after a probabilistic delay determined by a random variable. There have been a lot of members added to the SPN family, featuring different time specifications. The most fundamental Markov SPN model and Generalized SPN will be studied in this report. Some relatively mature analysis methods of Petri Nets or Markov processes (if applicable) can be applied directly to SPNs. However, there is still the complexity and state explosion problems for SPNs restricting its application. Nowadays, I see SPNs not so broadly used as described in papers from last century, but it's still in use in some simple systems such as Discrete Events Systems. A recent application on this will be briefly studied.

1 Background

Stochastic Petri Nets were proposed in the Performance Evaluation (PE) field. [1] The PE area can be initially subdivided into two subareas:

- One contains measurements, benchmarks, and prototypes. All they three are based on observing the behavior of the systems or its approximations in operation; also prototypes require more details than possible.
- The other is therefore introduced for PE in design phase – modeling, which contains simulation models and analytical models. By modeling one can focus on only important characteristics of the system. In the case of simulation models, the model is given by means of a computer program, whereas in the case of analytical models the model is given in mathematical terms.

Models can be either deterministic or probabilistic. Although it is clear that most systems of interest exhibit a deterministic behavior, it may be simpler to describe complex, detailed deterministic phenomena by means of macroscopic probabilistic assumptions. This is often done because details are not known, and even when they are, their inclusion may lead to very complex models. Furthermore, the probabilistic approach may be advantageous because it may provide sufficient accuracy, while yielding more general results, and it may permit the study of sensitivity to parameter variations.

It is important a model can describe the system behavior along the time scale. Although most modern systems are digital in nature and use a discrete time scale, models often use a continuous time scale for greater simplicity. Indeed, if the time axis is discrete, multiple events may occur between two consecutive time marks, thus making it hard to explore all possible combinations among these events. In the continuous time scale, instead, using appropriate probabilistic assumptions, it is possible to order events and to consider only one event at a time.

To satisfy all those requirements above, the model of Stochastic Petri Net (SPN) was introduced in the 1980s as an extension of the basic and untimed graphical Petri Net (PN) model, to model a system for performance evaluation.

Definition: Petri Net (PN) is a triple $N = \langle P, T, F \rangle$. P and T are sets of places and transitions respectively - they are both called elements, and they are disjoint. Flow relations, or arcs, are defined as $F \subset (P \times T) \cup (T \times P)$.

There are tokens in each place to decide the transitions firing following the place. A marking is a mapping which assigns tokens to each place. An execution of a Petri Net is therefore the transformation of the marking, which can be presented by a Reachability Graph (RG) – with each unique marking as a state, and the transformations between markings as arcs.

To add time phase to PNs, there are several alternative concerns characterizing the different proposals:

- The PN elements (either places or transitions) with which timing is associated,
- The semantics of the firing in the case of timed transitions (atomic firing or firing in three phases),
- The nature of the temporal specification (either deterministic or probabilistic).

Following these concerns, timed model of PNs are briefly divided into two main groups: Timed PN (TPN) [1]with deterministic or interval duration of events/transitions; Stochastic PN (SPN) with random/stochastic durations. The most commonly seen SPN models are PN models that are augmented with a temporal specification by associating a firing delay with transitions. The specification of the firing delay is of probabilistic nature, so that either the probability density function (pdf) or the probability distribution function (PDF) of the delay associated with a transition is needed.

The mathematical framework underlying SPN is the theory of stochastic processes. A stochastic process is a mathematical model useful for the description of phenomena of a probabilistic nature as a function of a parameter that usually has the meaning of time.

*Definition: A **stochastic process** $\{X(t), t \in T\}$ is a family of random variables defined over the same probability space, taking values in the state space S, and indexed by the parameter t, which assumes values in the set T; normally T = (0,∞).*

For certain types of SPNs, such as Markov SPNS in Section 2.2, their underlying stochastic processes are Markov processes.

*Definition: A **Markov process** is a stochastic process that satisfies the **Markovian property**, which says the behavior in the future (at some time t) depends only on the present situation, and not on the history. Markov processes with a discrete state space are called **Markov chains (MC)**. If the parameter t is discrete, the process is a **discrete-time Markov chain (DTMC)**. If the parameter t is continuous, the process is a **continuous-time Markov chain (CTMC)**.*

CTMC was extremely popular in the applied stochastic modeling field earlier than the 1980s, because it ensures the steady-state[2] solution can be factored in the product of the steady-state solutions of the individual queues, reducing complexity. However, it lacks descriptive power for phenomena such as synchronization and blocking. Therefore, Stochastic Petri Nets were introduced by applied stochastic

---

[1] In some articles SPN is defined as part of TPN.

[2] A system in a **steady state** has numerous properties that are unchanging in time. This implies that for any property *p* of the system, the partial derivative with respect to time is zero:$\partial p/\partial t = 0$. If a system is in steady state, then the recently observed behavior of the system will continue into the future. In stochastic systems, the probabilities that various states will be repeated will remain constant.

researchers, in a way that they are equivalent to CTMCs from the point of view of the solution, but offer a much greater descriptive power. This is also why the first SPNs were assigned the negative exponential PDF. Later they are extended in many other ways (Section **Error! Reference source not found.**).

2 STOCHASTIC PETRI NET (SPN)

Stochastic Petri nets are a form of Petri net where the transitions fire after a probabilistic delay determined by a random variable.

Formally, a *stochastic Petri net* is a five-tuple *SPN* = (*P*, *T*, *F*, *M0*, *Λ*) where:

- *P* is a set of states, called *places*.
- *T* is a set of *transitions*.
- *F* where $F \subset (P \times T) \cup (T \times P)$ is a set of flow relations called "arcs" between places and transitions (and between transitions and places).
- M0 is the initial marking.
- *Λ* is the array of *firing rates λ* associated with the transitions. The firing rate, a random variable, can also be a function λ(M) of the current marking, i.e. marking-dependent.

The marking-dependent timing rates can be helpful in solving problems like "m/m/n/n queue" problem, [13] which is more on the theoretical statistics side, and I will not study it in more detail. Most commonly seen SPNs are using marking-independent firing rates.

2.1 AN OVERVIEW OF DIFFERENT SPNS

Over the long time since the initial SPN was introduced, there have been a wide range of branches in the SPN family. Different SPNs, deterministic or probabilistic, Markovian or not, were targeting different stochastic processes. Paper [1], books like [4], and especially paper [2] by Gianfranco etc. in 1994, devoted great effort characterizing the underlying stochastic processes of different SPNs.

Initially SPN was proposed associating exponentially distributed firing delays to the transitions of a Petri net – it was Markov SPN actually, or Markovian SPN. Then Generalized SPNs were introduced to allow exponential transitions as well as immediate transitions. Underlying both of them are Markovian Chains (MCs), which can be automatically transformed into from the original Markov SPNs or GSPNs.

The need for non-exponentially distributed transition firing times in SPN's had been observed by many authors. Extended stochastic Petri nets (ESPNs) was defined to allow arbitrary distributed firing, assuming the underlying stochastic behavior is a semi-Markov process. Ciardo proposed several extensions to ESPNs and called this modeling formalism semi-Markov Process SPNs (SMP-SPNs). Deterministic and stochastic Petri nets (DSPNs), introduced by Ajmone Marsan and Chiola as an extension to GSPNs, include exponentially distributed and constant timing. If at most one deterministic transition is enabled in a marking, the steady-state solution can be computed using an embedded MC.

There are still a lot of other extensions on all those SPNs above. However, due to the time and energy limitation, in this report I'll only focus on the general definition of SPN, and the very fundamental Markov SPNs, as well as briefly GSPN and one other interesting branches, which is History-dependent SPN, a non-Markovian model.

2.2 MARKOV SPN

Markov SPNs are SPNs with negative exponential pdf. SPN models were proposed by researchers active in the applied stochastic modeling field, with the goal of developing a tool which allowed the integration of formal description, proof of correctness, and performance evaluation. For what concerns the last aspect, the proposals aimed at an equivalence between SPN and CTMC models, thus it is necessary to introduce temporal specifications such that the future evolution of the model, given the present marking, is independent of the marking history. To this purpose, sojourn times in markings must be random variables with negative exponential pdf, as explained below.

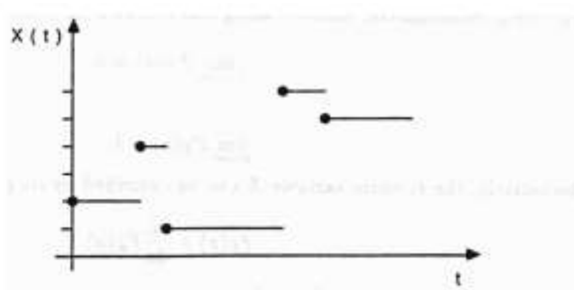

Figure 1 Time phase of CTMC

The sample path of a CTMC has the appearance depicted in Figure 1. Each horizontal segment represents the sojourn time in a state (black dots denote right-continuity). The Markovian property requires that sojourn times in states be exponentially distributed random variables. Indeed, the negative exponential pdf $f_X(x) = \mu e^{-\mu x} u(x)$ where u(x) is the unit step function, and p is the parameter (or rate) of the pdf, is the only continuous pdf for which the memoryless property holds.

In Markov SPNs, each transition firing is an atomic operation, i.e. tokens are removed from input places and put into output places with a single and indivisible operation. This is an important mechanism of the transitions firing, only with which can we have its reachability better studied.

The good point of Markov SPN exists when obtaining steady-state solutions inherited from underlying MC, so that certain performance problems can be solved easily. But there are also a list of problems:

- Solution complexity if the system is complex. It’s still being studied how to decompose the model into sub-models that can be studied separately and later re-composed, to reduce complexity.
- Structural characteristics of underlying PNs are changed dramatically, since the transition firing is changed. Thus it is not so convenient to apply study methods of PN to such SPNs.
- To compute other performance measures (such as delay distribution) for special SPN classes are still nontrivial, where model parameters can hardly be computed.

2.2.1 Relationship between SPN and MC

As introduced in previous sections, the SPNs were initially introduced to be equal to CTMC in some aspect. Which is to say, the underlying stochastic processes of most SPNs are assumed to be Markovian, i.e. following the Markovian Property.

The reachability graph of SPNs can be mapped directly to a Markov process. It satisfies the Markovian property, since its states depend only on the current marking. Each state in the reachability graph is mapped to a state in the Markov process, and the firing of a transition with firing rate λ corresponds to a Markov state transition with probability λ. A simple example is shown below in Figure 2

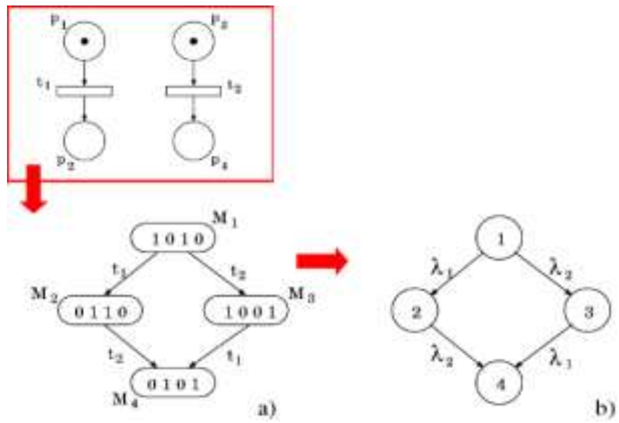

Figure 2 From SPN to MC

2.3 GENERALIZED SPN (GSPN)

The first SPNs were proposed associating exponentially distributed firing delays to the transitions of a Petri net, but this strict condition lead to the great difference between such SPNs and their underlying PNs. Therefore, Generalized stochastic Petri nets (GSPNs), were introduced by Ajmone Marsan, Balbo, and Conte later, by relaxing this condition by allowing “immediate“ transitions, with a constant zero firing time. [2] GSPNs were applied to the performance evaluation of multiprocessor systems. 11[6] Sometimes when some events take extremely small time to occur, it is useful to model them as instantaneous activities, and GSPN was designed for such modeling purpose.

Other than the immediate transitions, the remaining transitions have exponentially distributed firing times, thus extended from Markov SPNs actually, and they are called “timed transitions” in GSPNs.

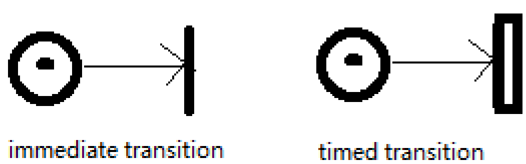

Figure 3 GSPN notations for transitions

In GSPN’s, firings of immediate transitions have priority over firings of timed transitions. Each immediate transition has associated a weight which determines the firing probability in case of conflicting immediate transitions.

2.3.1 Extended RG

For GSPNs, they have so called “Extended Reachability Graph” (ERG). Markings (states in ERG) enabling immediate transitions are passed through in zero time and are called *vanishing*; while markings enabling timed transitions only are called *tangible*. The ERG containing vanishing marking is no longer a CTMC. Since the process spends zero time in vanishing markings, they do not contribute to the time behavior of the system, and therefore must be eliminated to obtain the underlying CTMC.

To eliminate vanishing markings, one solution is to enable only timed transitions while executing a GSPN; a second solution is to enable one immediate and one timed transition respectively and alternately; or a third solution is to enable multiple immediate transitions at the same time.

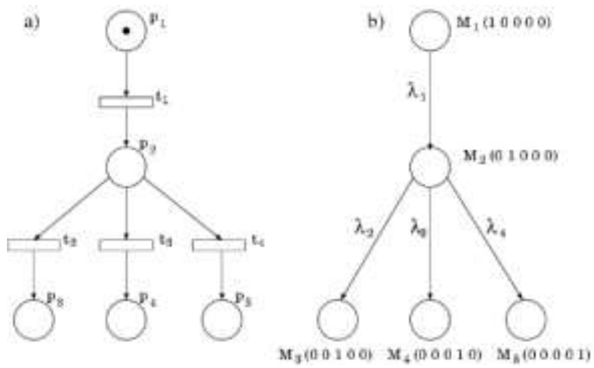

Figure 4 Eliminate vanishing markings in ERG for GSPN -1

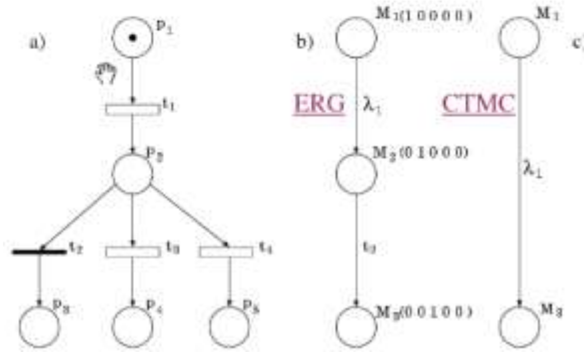

Figure 5 Eliminate vanishing markings in ERG for GSPN -2

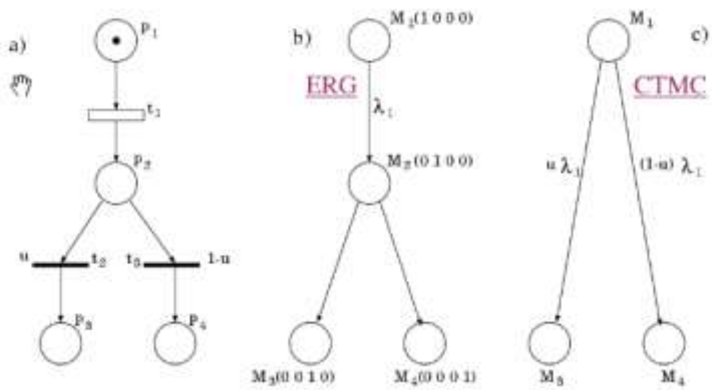

Figure 6 Eliminate vanishing markings in ERG for GSPN -3

All these solutions are feasible and are suitable for different situations, which the researchers have to decide by themselves.

2.3.2 Steady-state solution

It has been proved that GSPNs are equivalent to CTMCs, and solution methods for the derivation of the steady state probability distribution are also developed. [6]-[9]

The basic idea is to build a system of linear, first-order, ordinary differential equations to obtain transient solution, or to build a system of linear equations to get steady-state solution.

This methodology is also referred to as the “traditional methodology”.

2.3.3 Combined solution

To solve a SPN model means evaluating the (transient/steady-state) probability vector over the state space (markings). However, the modeler wants to interact only at the PN level since it is better descriptive and intuitional. As a result, the analytical procedure must be transparent to the analyst, which stimulating a need to define the output measures at the PN level, in term of the PN primitives.

The output measures defined at the PN level contain: probability of a given condition on the PN; time spent in a marking; mean or first passage time; distribution of tokens in a place; expected number of firing of a PN transition. All these measures can be formulated in terms of reward functions (MRM).

To sum up, solving models with SPN requires only the topology of the PN, the firing rates of the transitions and the specification of the output measures. The subsequent steps consist in generation of the RG and associated MC; transient and steady-state solution of the MC; and evaluation of the relevant process measures. All these steps must be completely automatized by a computer program, thus making transparent to the user the associated mathematics.

2.4 History- dependent SPN

Around 2009, History-Dependent Stochastic Petri Nets (HD-SPN), which is a non-Markovian SPN, was introduced by Wil van der Aalst etc. [5] For many real-life processes, choices made in the past can influence choices made later in the process. For example, taking one more iteration in a loop might increase the probability to leave the loop, etc. HD-SPNs were defined in a way that the transition firing probability distributions depend not only on the marking of the net, but also on the history of the net.

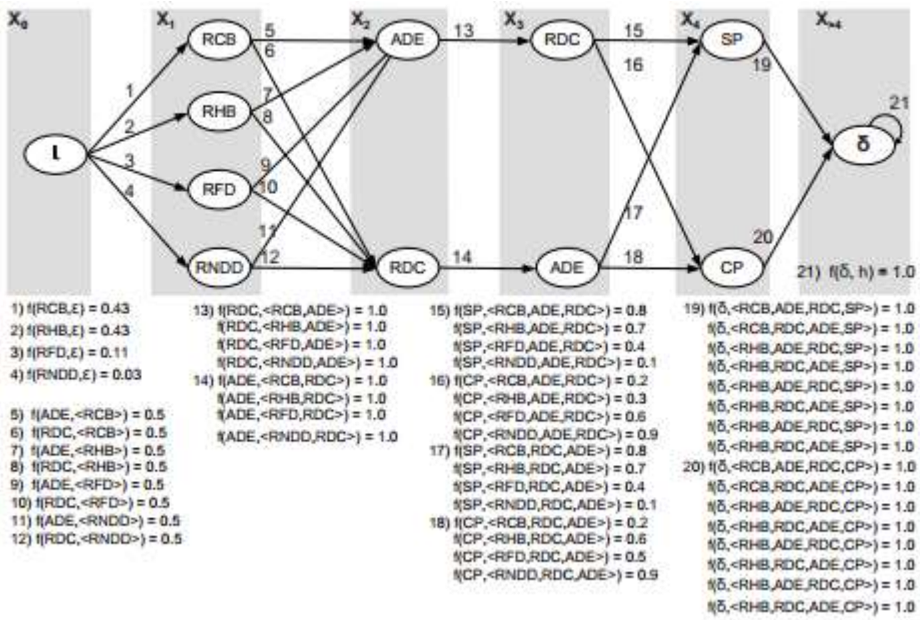

Figure 7 A discrete stochastic process modeled with HD-SPN

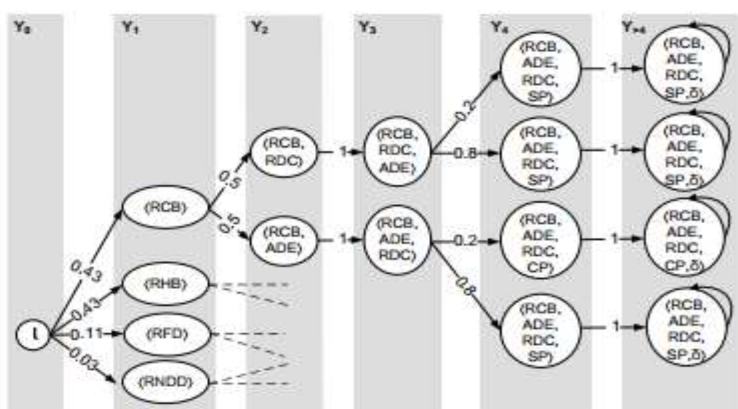

Figure 8 The underlying MC (partly) of the example in figure 7

They used the HD-SPN to discover the probabilistic mechanism from event logs of workflows, i.e. real-life observations were used to learn relevant correlations, and presented the usefulness of such model.

This is not an isolated case to design model making decisions based on history. As in lots of state estimation research or system monitoring studies, they define mathematically reward (or cost) functions all based on the history.

I also used the CPN tools used in this paper before for modeling colored Petri Net, it is a very powerful framework to model and simulate PN-modelled systems.

3 ANALYZE SPN

Since most SPNs are constructed from basic PN and MC, it has been long pursued that we use existing analysis methods of PN and MC to study the SPN model.

Specific methods have also been investigated recently in order to design graphs or structural models according to the sequences of events (i.e., inputs) and/or measurable states (i.e., outputs) that are measured. When sequences of states and events are both collected, the problem may be solved according to integer linear programming or binary linear programming.[3] The idea is to minimize a cost function that depends on the number of arcs in the graph and on the initial marking.

3.1 ANALYSIS FOR UNDERLYING PN

There are some basic Petri net properties [10], and SPNs mostly follow these properties as well.

- Reachability: The reachability problem for Petri nets is to decide, given a Petri net *N* and a marking *M*, whether the marking is in some fire sequence of N's RG, i.e. $M \in RG(N)$.
- Liveness property applied to the transitions firing in Petri nets. There are different degrees of liveness. A PN is called L_k-live if and only if all of its transitions are L_k-live, where a transition is
  - *dead*, if and only if it can never fire;
  - L_1-live (*potentially fireable*), if and only if it may fire, i.e. it is in some firing sequence;
  - L_2-live if and only if it can fire arbitrarily often;
  - L_3-live if and only if it can fire infinitely often, i.e. if for every positive integer k, it occurs at least k times in *V*, for some prefix-closed set of firing sequences;
  - L_4-live if and only if it may always fire, i.e., it is L_1-live in every reachable marking in the RG.
  - Note that L_{j+1}-liveness implies L_j-liveness, for j ∈ $\{1,2,3\}$.
- Boundness: A place in Petri net is called *k-bounded* if it does not contain more than *k* tokens in all reachable markings, including the initial marking; it is said to be *safe* if it is 1-bounded; it is *bounded* if it is *k-bounded* for some *k*.

One can surely construct the reachability graph of Petri Nets to prove its properties, but sometimes, the reachability graph may become huge: exponential in the number of places, or infinite at all – which is called the "largeness problem". Structural analysis enables us to prove some properties without constructing the reachability graph. The main techniques are place invariants, traps, and so on.[11]

3.2 ANALYSIS OF UNDERLYING MC (IF APPLICABLE)

If the regenerative transitions in SPN have exponential firing distributions and the subordinated processes are CTMCs, it is appropriate to attempt an analytical solution.

The brief idea has been discussed in the section 2.3.2 for GSPNs.

4 APPLICATIONS OF SPN

Nowadays, SPN is not as popular as before, yet it can still be seen modelling Discrete-Event Systems (DES) for static study. Discrete Event Systems (DESs) are characterized by signals that switch from one value to another rather than changing their value continuously. DESs occur naturally in engineering practice. An example is paper [3] written by Leclercq etc. in 2011, where the authors use SPN for detecting, isolating and identifying faults in DES – roughly in the static modelling checking scope.

Their methodologies are basically following the following line:

- First, learn reference model (timed PN) by collecting and statistically analyzing output alarm sequences, during which they estimate the parameters of the pdfs for transition firing periods.
- Next, use the reference models for fault detection and isolation issues.
- Finally, consider stochastic PNs with normal and exponential pdfs to include a representation of the faulty behaviors.

They used the model of the normal-stochastic PNs (NSPN). An NSPN is a PN with some normal transitions and some exponential ones. Such models are useful in representing dynamical systems with normal operations whose periods have variability (according to normal pdfs) and fault and recovery processes whose periods have exponential pdfs. In this paper normal events are assumed to occur according to normal pdfs, and fault events occur according to exponential pdfs. For any event in the considered sequences, the type of pdf (normal or exponential) is assumed to be known.

It is very common researchers extend some formal model in their own way to make it better suit the system to be modelled. In this paper they also did this. But after all the complexity of applying SPNs are not well solved yet, and now it's mostly used for relatively simple systems.

5 CONCLUSION

In this report I briefly studied the SPN family and several different types of SPNs. SPNs were first developed in the 1980s for system performance evaluation research, by several statistical researchers. The theoretical statistical model MC was very popular at that time, thus most SPNs have the tendency of assuming a Markov process is underlying the modeled system, to take advantage of the research results of applying MC to PE study.

The PN itself is already a very descriptive formal model, also a great advantage is its graphical presentation and automatic simulation on computers – a lot of tools [12] are ready-to-use.

After combining MC and PN to get the initial SPNs (CTMC-SPN actually), the SPNs have been extended in a variety of aspects, such as GSPN, DSPN and ESPN, which update firing time distributions; and recent HD-SPN, which models non-Markov processes.

Many researches have been done to apply to SPNs the techniques for analyzing traditional Petri Nets (e.g. reachability graph and structural analysis) and studying Markov processes (mostly mathematical calculation). But almost every time researchers want to reuse an existing technique to SPN, they have to extend or modify the SPN in a degree. I think the naive reason is the complexity of SPNs, since they are designed to have more powers, they are more difficult to handle and get perfectly solved. Nowadays I see only in Discrete Event Systems are SPNs still broadly used, maybe this is just a temporal trend, but still showing the imperfection of the SPN series.

All these papers I studied are restricted to either time or purpose. The papers [1] and [2] introducing basic concepts of SPNs are from very old times, thus restricting the views contained inside. Over the yeas there have been more extensions as well as solution algorithms added to the SPN family. In paper [3] which introduces a modern application of SPN, SPN is still used for relatively simple systems, showing the disadvantages of the SPN models are still not eliminated yet.

Further work will be to look into details of all the solution algorithms and most recent techniques, especially on the statistical side.

REFERENCE

[1]. Marsan, M Ajmone. "Stochastic Petri Nets: An Elementary Introduction." In *Advances in Petri Nets 1989*, 1-29: Springer, 1990.

[2]. Ciardo, Gianfranco, Reinhard German, and Christoph Lindemann. "A Characterization of the Stochastic Process Underlying a Stochastic Petri Net." *Software Engineering, IEEE Transactions on* 20, no. 7 (1994): 506-15.

[3]. Lefebvre, D., and E. Leclercq. "Stochastic Petri Net Identification for the Fault Detection and Isolation of Discrete Event Systems." *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 41, no. 2 (2011): 213-25.

[4]. Bérard, Béatrice, MariaPaola Cabasino, Angela Febbraro, Alessandro Giua, and Carla Seatzu. "Petri Nets with Time." Chap. 16 In *Control of Discrete-Event Systems*, edited by Carla Seatzu, Manuel Silva and Jan H. van Schuppen. Lecture Notes in Control and Information Sciences, 319-41: Springer London, 2013.

[5]. Schonenberg, Helen, Natalia Sidorova, Wil van der Aalst, and Kees Hee. "History-Dependent Stochastic Petri Nets." Chap. 31 In *Perspectives of Systems Informatics*, edited by Amir Pnueli, Irina Virbitskaite and Andrei Voronkov. Lecture Notes in Computer Science, 366-79: Springer Berlin Heidelberg, 2010.

[6]. Ajmone Marsan, Marco, Gianni Conte, and Gianfranco Balbo. "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems." *ACM Transactions on Computer Systems (TOCS)* 2, no. 2 (1984): 93-122.

[7]. Marsan, M Ajmone, and Giovanni Chiola. "On Petri Nets with Deterministic and Exponentially Distributed Firing Times." In *Advances in Petri Nets 1987*, 132-45: Springer, 1987.

[8]. Chiola, Giovanni, Marco Ajmone Marsan, Gianfranco Balbo, and Gianni Conte. "Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications." *Software Engineering, IEEE Transactions on* 19, no. 2 (1993): 89-107.

[9]. Buchholz, Peter, and Miklós Telek. "Stochastic Petri Nets with Matrix Exponentially Distributed Firing Times." *Performance Evaluation* 67, no. 12 (12// 2010): 1373-85.

[10]. Murata, Tadao. "Petri nets: Properties, analysis and applications." Proceedings of the IEEE 77, no. 4 (1989): 541-580.

[11]. “Petri nets: Structural analysis”, 2013/05, http://www.lsv.ens-cachan.fr/~schwoon/enseignement/verification/ws0910/nets2.pdf

[12]. Tools and Software for Petri Net, 2013/05, http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/

[13]. http://www.docstoc.com/docs/84743139/Probability-and-Statistics-with-Reliability-Queueing-and-Computer-Science-Applications,2013/05/08