We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

172,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Assessing Risks in Critical Systems using Petri Nets

Lucio Flavio Vismari and João Batista Camargo Junior
*Computer and Digital Systems Engineering Department
School of Engineering at University of São Paulo (Poli-USP)
São Paulo, Brazil*

1. Introduction

The advent of ubiquitous distributed computing, communication and sensing systems has created an environment in which one can access, process and communicate huge amounts of data. This environment could be mentioned as the major enabler for new applications of control for large-scale, complex systems (Murray et al., 2003). Currently, energy and water distribution, transportation and manufacturing processes are some application areas that use the integration of distributed computing and communication in supervision and control systems to optimize their processes performance and to increase their service capacity. In order to reduce time and cost of development, increase flexibility for systems expansions or modifications, promote greater reuse of proven components and, consequently, reduce the system lifecycle costs and enable a higher number of applications (Hammett, 2003), some systems are being implemented with distributed computing architectures controlled over open (standard-oriented) communication networks (mostly commercial communication networks) and with “Commercial Off The Shelf” (COTS) hardware and software components (Ahlström & Torim, 2002; Knight, 2002b).

If, on the one hand, the characteristics of distributed computing systems with supervision and control loop closed over networks make some applications feasible, on the other hand they pose new challenges and problems to be solved. One of the main challenges for applying this new system paradigm over real applications is its complexity. The huge number of functional elements distributed and interacting over open communication networks makes the system behavior unpredictable, both in operational conditions and in presence of failures. Another challenge comes from the interdisciplinary characteristic of the new paradigm. Particular concepts of control, computer sciences and communication disciplines must be considered together. As a consequence, new points-of-view to be considered are introduced into the systems lifecycle. As an example, concepts of time delays, data transfer rates and packet length (from communication discipline); stability (from control discipline), security and real-time computing (from computing discipline) must be systemically considered in all system lifecycle stages. A last challenge mentioned here is the lack of well-established methods to assess systems based on this new paradigm, which is a need considering all the new paradigm characteristics relationship between

supervision and control system and the application process. In this case, one of the current problems here is modeling and analyzing hybrid systems, in which continuous and discrete variables must be modeled and analyzed together.

Challenges increase when this new supervision and control paradigm is applied to dependable critical applications, mainly regarding safety, where failures could result in loss of lives, significant damage to property or to the environment (Knight, 2002a). Due to the unacceptable consequences that could impact the environment when critical systems are working out of their specifications, any (dependable) critical system must be submitted to a rigorous process of assessment to guarantee that all the risks contained in its lifecycle is acceptable to the society. Hence, due to the characteristics of complexity (defined by the strong interdependence between multidisciplinary variables) in the current critical systems, the challenge is to deal with all of these relevant characteristics during the system assessment process in a complete and efficient way. Therefore, the mission of this chapter is presenting an assessment framework based on the Fluid Stochastic Petri Net (FSPN), the Reward Structures and Variables and the Discrete Event Simulation (DES), which could deal with the characteristics and properties found in the current (dependable) critical systems.

The Fluid Stochastic Petri Net (FSPN) is a formalism commonly applied to model and analysis systems with the same characteristics identified in the current dependable critical systems (concerning safety domain, security domain, time-related domain and hybrid systems domain). In FSPN formalism, the primitive elements of a Stochastic Petri Net (SPN) – such as places (P), transitions (T) and arcs (A) – can represent continuous and discrete variables in hybrid systems, besides representing the (stochastic) dynamic behavior between those interdependent parts (a desirable property when complex systems are modelled). Using FSPN, it is possible to perform a stochastic analysis obtaining performance, reliability, availability and safety measures (dependability measures) based on the uncertainty parameter of the systems. Finally, the FSPN has the capacity to model continuous state spaces, interdependent concerning the discrete process, resulting in a useful modeling formalism to the real systems.

The FSPN, in its pure formalism, allows representing a limited number of system metrics. Improving the representativeness of metrics in the proposed framework, the concept of *Reward Structures and Variables* was incorporated. The existence of the reward structures in the FSPN formalism allows assessing complex and hybrid systems using a variety of measures, defined during the modeling phase. Obtaining the measures of interest in a quantitative way requires calculating the state probability of the modeled system. Thus, in this framework we propose the use of Discrete Event Simulation (DES) to obtain the expected values of reward functions. *Simulation* is a better choice to solve non-Markovian Petri nets, as well as complex and non-limited nets in which state spaces are analytically unsolvable. Since the Stochastic Petri Net is a formalism applied to describe the dynamics of discrete systems, where the state evolution (mark changing) occurs in discrete and well defined events, the solution of nets by *Discrete Event Simulation* (DES) is an extension of the formalism itself. However, the simulation technique only allows statistically estimating the values of interest. Therefore, the quantitative analysis using simulation techniques, mainly DES, must incorporate statistical inference concepts as a way to analyze values with acceptable confidence levels, and this approach was incorporated in the proposed framework.

In short, this chapter will address the key topics related to the proposed assessment framework. Both the formal concepts of FSPN and its ability to represent hybrid systems will be presented. Also, the formalisms of reward nets and their use in quantitative assessment process will be discussed. A topic about resolution methods in FSPN and the ways to apply the DES to estimate measures of interest in FSPN models will be included. Finally, the application of the proposed framework (presented in this chapter) to a real dependable critical system – the Air Traffic Control (ATC) system – will be shown.

2. A Framework to assess risks in dependable critical systems

2.1 The current system paradigm and the proposed assessment framework

An accident can be defined as a consequence of a hazardous (or unsafe) state that promotes unacceptable losses of lives, injuries and material and environmental losses (Kirner, 1997). Therefore, an accident is an event (or a sequence of events) that violates system safety requirements and, consequently, the main objective of a safety-critical system is to minimize, to acceptable levels, conditions that can contribute to the occurrence of an accident. In this way, a safety assessment process needs to identify what hazard (or unsafe) states can be reached in a system, assess their risks and, finally, to establish countermeasures to minimize and protect a system against the occurrence of hazard states that can lead it to an accident (Leveson, 1995). Hence, it is mandatory to have a complete and efficient way to consider all the current safety-critical system characteristics during the assessment process, allowing the complete identification and the assessment of emergent risks in systems.

As mentioned before, dependable critical systems are being implemented with distributed computing architectures over open-standard communication networks, and the introduction of this paradigm promotes a huge number of interdependent, multidisciplinary characteristics in systems. Two of the most important characteristics that need to be considered in the safety assessment process are regarded to time-related domain and hybrid domain:

Time-related domain: implementation of distributed architectures controlled over networks introduces time-related characteristics in systems, reflected by strong timing constraints. In the case of control being closed-loop over networks, stochastic variations of time delay in networks and in control processing must be considered. Delays in control system may promote effects such as performance degradation and destabilization (Tipsuwan & Chow, 2003) that are dangerous in safety critical applications. Another temporal constraint aspect is related with task sequences. In safety critical systems, tasks must be rigorously triggered. Currently, time-triggered protocols are recommended in safety-critical applications controlled over networks since they guarantee, with some level of confidence, a correct task sequence, despite its lower performance compared to event-triggered protocols (Philippi, 2002; Heck et al., 2003). Concluding, network characteristics such as delays and latency must be considered over safety critical systems requirements.

Hybrid domain: as reported by Heck et al. (2003), a hybrid system is an abstract system with both continuous and discrete components. Current safety critical systems have computer-based systems controlling safety critical applications. Computer-based systems have their behavior performed by discrete-state machines, modeled by discrete states, and safety-critical applications have their behavior modeled by continuous variables (e.g. aircraft velocity, wind speed, temperature, etc). A hybrid model is necessary to modeling and

analyzing the affectation of disturbances/failures in control system over safety related applications and vice-versa.

Based on the new paradigm characteristics listed before, it was verified in the literature (Tuffin et al., 2001; Wolter, 2000; Gribaudo, 2003) that Fluid Stochastic Petri Nets (FSPN) are a formalism applied to modeling and assessing systems with the same characteristics identified for the current safety critical system paradigm. In FSPN formalism, primitive elements of a Stochastic Petri net (SPN) such as places (P), transitions (T) and arcs (A) can represent both continuous (P_c, T_c, A_c) and discrete (P_d, T_d, A_d) variables in a hybrid system, besides the dynamic behavior between those parts. FSPNs are fully applied to modeling and verify functional and safety properties in hybrid systems (Gribaudo et al., 2002). A formal definition of FSPN can be made by (adapted from Computer Science Department (2000) and Ciardo et al. (1999)):

$$\text{FSPN} = (P, T, A, f, \Pi, g, \text{Dist}, r, \text{Afet}, w, b, \mu_0, \chi_0) \quad (1)$$

where some import arguments of (1) are defined as (the other will be defined further on):

$$P = P_d \cup P_c = (p_1, p_2, \dots, p_{\#P_d}) \cup (q_1, q_2, \dots, q_{\#P_c}), P_d \cap P_c = \emptyset.$$

$$\mu_{(t)}(p_i) \in \mathbb{N}, \forall p_i \in P_d \text{ is a marking (state) in discrete net at time } t, t \in \mathbb{R}^+,$$

$$\chi_{(t)}(q_i) \in \mathbb{R}^+, \forall q_i \in P_c \text{ is a marking (state) in continuous net at time } t \in \mathbb{R}^+,$$

$$T = T_T \cup T_I = (t_1, t_2, \dots, t_{\#T_I}) \cup (u_1, u_2, \dots, u_{\#T_I}), T_T \cap T_I = \emptyset.$$

$$A = A_d \cup A_c = ((P_d \times T) \cup (T \times P_d)) \times \hat{S} \rightarrow \mathbb{N} \cup ((P_c \times T_I) \cup (T_I \times P_c)) \times \hat{S} \rightarrow \mathbb{R}^+.$$

$$f = ((P_c \times T_T) \cup (T_T \times P_c)) \times \hat{S} \rightarrow \mathbb{R}^+ \text{ : marking-dependent fluid rate.}$$

Graphical representation of FSPN elements are illustrated in Fig. 1.

Fig. 1. Graphical representation of FSPN elements

According to Gribaudo et al. (2003), it is possible to unify stochastic and formal methods using FSPN to modeling and analyzing systems with discrete and continuous parts. Considering the evolution of FSPN from elementary Petri nets, FSPN can be used for modeling system functional properties related to their discrete states, such as conformity and reachability states. Considering the evolution of FSPN from stochastic Petri nets (SPN), it is possible to use FSPN to perform stochastic analysis obtaining performance, reliability, availability and safety measures (dependability measures) based on the systems uncertainty parameter. Finally, FSPN is capable of modeling continuous state spaces interdependent of discrete process, resulting in a useful modeling formalism for real systems.

A systematic way to define measures of interest in a net, including models in FSPN, is using *reward functions*, where *reward structures* can be represented, at a high abstraction level, with conditional expressions (German, 2000). Union of Stochastic Petri Nets (SPN) and Reward Structures formalism are named Stochastic Reward Nets (SRN) (Muppala et al., 1994). In SRN, measures are obtained by reward variables, as instantaneous reward (Equation 2), accumulated reward (equation 3) and average reward (equation 4) variables.

$$R_{inst}(t) = \sum_{n \in S} rr_n \cdot 1_{\{N(t)=n\}} + \sum_{g \in T} \sum_{n \in S} ri_n^g \cdot 1_{\{g \text{ fires at } t \text{ in state } n\}} \cdot \delta, \quad (2)$$

$$R_{acum}(t) = \int_0^t R_{inst}(\varphi) d\varphi \quad (3)$$

$$R_{aver}(t) = \frac{1}{t} \cdot R_{acum}(t) \quad (4)$$

$R_{inst}(t)$, $R_{acum}(t)$ and $R_{aver}(t)$ present stochastic behavior if their impulsive reward vector (ri) is not considered. They can thus be characterized by their probability distributions. According to Muppala et al. (1994), output measures from a Stochastic Reward Net (SRN) can be represented by expected values of reward variables. Therefore, considering $E[R_{acum}(t)]$ the expected value for reward variable $R_{acum}(t)$ (German, 2000):

$$E[R_{acum}(t)] = \sum_{n \in S} rr_n \cdot \int_0^t \pi_n(x) dx, \text{ where } \pi_n(t) = \Pr(N(t) = n), n \in S. \quad (5)$$

$E[R_{inst}(t)]$ is calculated as a consequence of (5), where:

$$E[R_{inst}(t)] = \frac{d}{dt} E[R_{acum}(t)] = \sum_{n \in S} rr_n \cdot \pi_n(t) \quad (6)$$

Getting quantitative measures from Stochastic Petri nets (and their evolutions) using reward structures demands the calculation of state probabilities $\pi_n(t) = \Pr(N(t) = n)$. Vector $\pi(t) = (\pi_1(t), \pi_2(t), \dots, \pi_m(t))$, $S = \{1, 2, \dots, m\}$ can be calculated using 2 possible ways: *Analytical Techniques* and *Simulation* (Computer Science Department, 2000). Applying *analytical techniques* to solve stochastic Petri nets (and their evolutions) requires nets with Markovian (memoryless) characteristics. This fact narrows modeling capabilities, restricting net transitions to exponential and geometric fire time probability distribution only. Besides constraints related to transition types, complex and non-limited nets can be unsolvable by analytical techniques because of the large number of states they produce.

Simulation is a better choice for solving non-Markovian Petri nets; solving complex and non-limited nets that state space size is analytically unsolvable. Once stochastic Petri nets are a formalism applied to describe dynamics of discrete systems, where state evolution (mark changing) occurs in discrete and well defined events, nets solution by *Discrete Event Simulation* (DES) is an extension of the Petri nets formalism. However, the simulation technique allows only doing an inference of values of interest. In this way, quantitative analysis using simulation techniques, mainly DES, must incorporate statistical inference concepts as a way of evaluating values with acceptable confidence levels. Therefore, it is

necessary to have good statistical estimators to calculate values (5) and (6) to use the simulation technique.

Fig. 2 summarizes the process of obtaining measures of interest in the Assessment Framework, where FSPN, DES, SRN and Statistical Inference are put together as the way to assess the characteristics and properties found in the current dependable critical systems. The main concepts of FSPN, SRN and DES used in the framework are detailed in the following items.

Fig. 2. Obtaining measures of interest using the Assessment Framework

2.2 The Fluid and Stochastic Petri nets and its ability to represent Hybrid Systems

2.2.1 Petri nets and their evolution in the assessment of systems

Petri nets (PN) are a graphical and mathematical formalism used in the specification, modeling, verification and analysis of systems properties. Particularly, PN are used to model complex systems that have concurrent, parallel, asynchronous, distributed and/or non-deterministic processes (Murata, 1989), and to assess availability, performance and other systems characteristics (Computer Science Department, 2000; Bobbio, 1988). For example, the Petri nets have a wide application in modeling computer systems (Peterson, 1989 apud Labeau et al., 2000), considering the relationship between Hardware, Software and Human-Machine Interface in a systemic context (Leveson & Stolzy, 1987).

However, the elementary structure of a Petri Net is restricted to deal with real systems. For example, an elementary PN can not assess whether a place has no tokens (test of zero). Nor is it possible to relate the cardinality of the arcs with the markings of the net (marking-dependent arcs). Another elementary PN limitation is not solving the firing of two or more transitions in conflict. Thus, as a way of increasing the power of modeling, some structural

extensions have been created in the basic concept of PN, such as guards, priority, arc multiplicity and marking-dependent arc cardinality.

An important extension to Petri nets is considering, beyond immediate firing transitions, firing time to the transitions, allowing the use of PN in the assessment of important aspects of systems, such as availability, performance, reliability, safety and other measures based on time. At first, 2 extensions were adopted in the timed Petri nets concept (Murata, 1989): the deterministic-timed Petri nets and the Stochastic Petri nets (SPN). In both cases, time delays (τ) were associated with transitions (T) so that, if $t \in T$ is timed with $\tau_t \in \mathbb{R}^+$, and $\delta(\mu_k, t_k) = \mu_{(k+1)}$ (where $\delta : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$ is the *next state function* and $\mu: P \rightarrow \mathbb{N}$ is the *marking function* that maps the set of places P in a Petri net to non-negative integers), the transition from state k to state $k+1$ ($\mu_k \rightarrow \mu_{(k+1)}$) will consume a time $\tau_{t,k}$ between enabling the transition t ($t \in H(\mu_k)$), where $H(\mu) \in T$ is the set of enabled transitions in the marking μ of the Petri net), and its effective firing event ($\uparrow t$). The difference between these extensions is the category of delay (variable) used. In the deterministic-timed Petri nets, τ_t is deterministic (not a random variable). In the case of Stochastic Petri nets (SPN), the set of time delays is created by random variables $\tau_t : \Omega \rightarrow \mathbb{R}$ defined in a space of probability $(\Omega, \mathcal{P})$ with probability density function $F\tau_t(x) = P(\tau_t < x)$.

When time delays are introduced in Petri net transitions, marking (states) are represented by $s_k = (\mu_k, \tau_{t,k}) \in \mathbb{N}^{#P} \times (\mathbb{R}^+)^{\#T}$. Thus, a timed Petri net has a firing sequence $T\uparrow = (t_0, t_1, \dots, t_n)$ that produces a marking sequence $(s_0, s_1, \dots, s_n)$ associated to the events of time $(e_0, e_1, \dots, e_n)$, where $(e_{k+1} - e_k) = \tau_{t,k}$. Hence, the reachability set S to a timed Petri net is the set of markings defined by $s_k \in S$.

Due to the dynamic execution of SPN having stochastic characteristic related to the time (stochastic transitions), the firing dynamics can be represented by a stochastic process (German, 2000). A stochastic process is formally defined as a family of random variables $\{N(t), t \in Y\}$, where $N(t)$ is defined on a space of probability $S = (\Omega, \mathcal{P})$ and indexed by the parameter t , t varying on a set *index* Y . For an SPN, we have $N(t) \in S$, where S is the set of tangible states of an SPN and t is the parameter of time, which can be both discrete ($t = e_k =$

$e_0 + \sum_{t \in \sigma, n=[0,k]} \tau_{t,n}$) and continuous ($t \in \mathbb{R}^+$), describing the relationship between the states of a

stochastic process at different time instants.

Stochastic Petri nets with all their transitions associated with exponential distribution functions are called Generalized Stochastic Petri nets (GSPN), and due to their memoryless property they can be represented by a Markovian process. In this case, the dynamic behavior of markings are mapped into a *time-continuous homogeneous Markov chain* with an isomorphic state space to the reachability tree of the GSPN (Bobbio, 1988). Thus, when a net is memoryless (a Markovian net), one can ignore the time delay (τ_t) of enabled transitions at μ_k that were not fired in this marking (state) and, after any transition firing, became disabled at marking $\mu_{(k+1)}$. Moreover, if a net has transitions with distribution functions different from an exponential, the SPN can not be represented by a Markovian process. Besides, it is not possible to use the properties of a Markov chain; the execution dynamics of a SPN must establish what policy of time resampling to t_q must be used when $\{t_j, t_q\} \in H(\mu_k)$, $\delta(\mu_k, t_{jk}) = \delta(\mu_k, t_{qk}) = \mu_{(k+1)}$ and $t_q \notin H(\mu_{(k+1)})$, but $t_q \in H(\mu_{(m > k+1)})$. 3 resampling policies are defined: PRD (preemptive repeat different), PRI (preemptive repeat identical) and PRR (preemptive repeat resume). PRD is the policy used in a Markovian model (memoryless),

where a new time delay to t_q is resampled when it is enabled again. PRI uses the same sampled time delay at the last t_q enabling event. PRR uses the remaining time delay at the last t_q disabling event with no t_q firing.

2.2.2 The FSPN and the Hybrid Systems (HS)

The Stochastic Petri nets (SPN) can model and assess dynamic systems with both discrete state spaces and stochastic properties by incorporating characteristics of elementary Petri nets and stochastic temporal characteristics. However, a large segment of real world systems is hybrid, where continuous processes (which contain real values) are controlled by discrete logic processes (such as computers). The control system of an aircraft, the manufacturing processes, the communication networks and intelligent systems are examples of real-world hybrid systems (Tuffin et al., 2001).

According to Varaiya (1999) apud Tuffin et al., (2001), a Hybrid System (HS) consists of 2 interdependent components: a continuous component, $q \in \mathbb{R}^n$, and an event-driven finite state space $p \in \mathbb{Q}$. Given a state (q_k, p_k) , only the continuous state evolves according to the set of differential equations $dq(t)/dt \in F(q(t), p_0) \subset \mathbb{R}^n$, during the time interval $[t_k, t_{k+1})$. At time t_{k+1} , there is $q(t_{k+1}) \in G(p_k, p_{k+1})$, where G is an "enabling zone" that checks the permission of migration for the discrete state p_k to p_{k+1} . If the migration is allowed, there is an instantaneous state migration in the discrete model (p_k para p_{k+1}) and the continuous state is brought to q_{k+1} according to the relation $(q(t_{k+1}), q_{k+1}) \in R(p_k, p_{k+1}) \subset \mathbb{R}^n \times \mathbb{R}^n$, where $R()$ is defined for each pair (p_k, p_{k+1}) in the discrete model. The dynamics of the model evolves in the same way for all the horizon time.

Thus, the continuous part of a hybrid system is modeled by differential equations, while the discrete events evolves in temporal events (Computer Science Department, 2000), and both are related through the mechanism explained above. The formalism of the Fluid and Stochastic Petri nets (FSPN) embodies the representation of continuous (fluid) places and arcs to the characteristics of Stochastic Petri Nets (SPN), where the evolution of continuous values is represented by fluid rate functions (related to fluid arcs), and the variation of discrete states follows the same rules of Stochastic Petri Nets. Primarily, the concept of FSPN introduced by Trivedi & Kulkarni (1993) intended to solve problems of state explosion in the reachability trees of nets, avoiding the accumulation of marks (tokens) in discrete places. Therefore, a FSPN was not defined for modeling hybrid systems.

Still relating the differences between formalisms (FSPN and HS), and due to the legacy of SPN, the FSPNs are focused on the evaluation of measures in stochastic processes involved in the modelled system. The formalism of hybrid systems (HS) focuses on system reachability and decision analyses (model of control). The influence of uncertainties in the HS modeling is represented by the boundaries of the states in model. Even with these differences in formalism and application, Tuffin et al., (2001) proved that hybrid systems can be represented by the Fluid Stochastic Petri nets.

As shown by equation (1), a FSPN can be formally defined by a 13-upla as $FSPN = (P, T, A, f, \Pi, g, Dist, r, Afet, w, b, \mu_0, \chi_0)$, where:

$$\begin{aligned} P &= P_d \cup P_c = (p_1, p_2, \dots, p_{\#P_d}) \cup (q_1, q_2, \dots, q_{\#P_c}), P_d \cap P_c = \emptyset. \\ \mu_{(t)}(p_i) &\in \mathbb{N}, \forall p_i \in P_d \text{ is a marking (state) in discrete net at time } t, t \in \mathbb{R}^+, \\ \chi_{(t)}(q_i) &\in \mathbb{R}^+, \forall q_i \in P_c \text{ is a marking (state) in continuous net at time } t \in \mathbb{R}^+, \\ T &= T_T \cup T_I = (t_1, t_2, \dots, t_{\#T_t}) \cup (u_1, u_2, \dots, u_{\#T_i}), T_T \cap T_I = \emptyset. \end{aligned}$$

$A = A_d \cup A_c = ((P_d \times T) \cup (T \times P_d)) \times \hat{S} \rightarrow N \cup ((P_c \times T_I) \cup (T_I \times P_c)) \times \hat{S} \rightarrow R^+$, where:
 $A_d = I_d \cup O_d \in (N^{\#P_d \rightarrow N})^{\#P_d \times \#T} \cup (N^{\#P_d \rightarrow N})^{\#T \times \#P_d}$: marking-dependent cardinality of input and output arcs (I e O) between discrete places P_d and transitions T .
 $A_c = I_c \cup O_c \in (N^{\#P_c \rightarrow N})^{\#P_c \times \#T_I} \cup (N^{\#P_c \rightarrow N})^{\#T_I \times \#P_c}$: marking-dependent *fluid impulse* of input and output arcs (I e O) between fluid (continuous) places P_c and the immediate transitions T_I .
 $f = ((P_c \times T_I) \cup (T_I \times P_c)) \times \hat{S} \rightarrow R^+$: marking-dependent *fluid rate*.
 $\Pi \in N^{\#T_I}$ is the *priority vector* of immediate transitions T_I .
 $\text{Dist} \in T_T \times \hat{S} \rightarrow F$ are the marking-dependent firing time distributions of transitions T_T (where F is the set of positive distribution functions $[0, \infty]$).
 $r \in (\{PRD, PRS, PRR\})^{\#T_t}$ are the resampling policies to the transitions T_T .
 $A_{fet} : T \rightarrow T_T \in (\{PRD, PRS, PRR\} \times \{PRD, PRS, PRR\})^m$ is the resampling policy that a transition $t_i \in T_T$ must adopt on marking $\mu_{(k+1)}$ where $\{t_i, t_j \cup u_j\} \in H(\mu_k)$, $\delta(\mu_k, t_{ik}) = \delta(\mu_k, t_j \cup u_j) = \mu_{(k+1)}$ e $t_i \in H(\mu_{(k+1)})$. This property is named *afectation*.
 $w \in N^{\#T} \rightarrow R^+$ are the marking dependent weights of transitions T .
 $b : P_c \times N^{\#P_d} \rightarrow R_0$ are the fluid limits to each fluid (continuous) place P_c .
 $g : (\hat{S} \rightarrow \{True = 1, False = 0\})^{\#T}$ is a guard vector of T .
 $\mu_0 = \mu_{(t=0)}(p_i)$, $\forall p_i \in P_d$: marking from discrete net at $t = 0$ (*discrete initial marking*).
 $\chi_0 = \chi_{(t=0)}(q_i)$, $\forall q_i \in P_c$ e $\chi(q_i) \leq b_q$: marking from discrete net at $t = 0$ (*continuous initial marking*).
 $S = (\mu, \chi)_{(t)} \subseteq \hat{S} = N^{\#P_d} \times (R^+)^{\#P_c} \times R^+$, where $\hat{S}$ is the potential state space and S is the effective state space of FSPN. In the same way as SPN, FSPN can be represented by a stochastic process $\{N(t) \in \hat{S}, t \in R^+\}$.

Regarding the FSPN execution dynamics, a transition $t \in T$ is enabled in a marking $(\mu, \chi)_{(t)} \in \hat{S}$ at time $t \in R^+$, $t \in H((\mu, \chi)_{(t)})$, for the same conditions of a Stochastic Petri net, i.e.:

$$A_{p_i, t}((\mu, \chi)_{(t)}) \leq \mu_{(t)}(p_i) \vee f_{q_i, t}((\mu, \chi)_{(t)}) \leq \chi_{(t)}(q_i), \forall p_i \in P_d, \forall q_i \in P_c. \quad (7)$$

and

$$g_t((\mu, \chi)_{(t)}) = 1. \quad (8)$$

Similarly, for $t \in H((\mu, \chi)_{(t)})$, a marking $(\mu, \chi)_{(t)}$ is considered *immediate* when $t \in T_I$, and a marking $(\mu, \chi)_{(t)}$ is considered *tangible* when $t \in T_T$.

Given $t \in H((\mu, \chi)_{(t=k)})$, $t \in T$, marking $(\mu, \chi)_{(t=k)}$ is changed to the state $(\mu, \chi)_{(t=k+\tau)}$ in 2 ways (Computer Science Department, 2000) (where τ is the time between the event $t \in H((\mu, \chi)_{(t=k)})$ and $\uparrow t$, i.e. the timing of t at $t = k \in R^+$):

- **1st way:** with respect to *discrete arcs* (A_d) and *fluid impulses* (A_c), $(\mu, \chi)_{(t=k)}$ changes in form:

$$\mu_{(t=k+\tau)}(p) = \mu_{(t=k)}(p) + A_{t,p}((\mu, \chi)_{(t=k)}) - A_{p,t}((\mu, \chi)_{(t=k)}), \forall p \in P_d \quad (9)$$

$$\chi_{(t=k+\tau)}(q_i) = \min(b_q(\mu_{(t=k+\tau)}) ; \max(0 ; \chi_{(t=k)}(q_i) + A_{t,q}((\mu, \chi)_{(t=k)}) - A_{q,t}((\mu, \chi)_{(t=k)}))), \forall q \in P_c \quad (10)$$

- **2nd way:** with respect to *fluid arcs* f , if the marking $(\mu, \chi)_{(t=k)}$ is *tangible*, the fluid level flow continuously through arcs f enabled by transitions $t_i \in T_T$ until $t_i \notin H((\mu, \chi)_{(t=k)})$.

Considering $\delta_q^{\text{pot}}((\mu, \chi)_{(t=k)}) = \sum_{t \in \Lambda} (f_{t,q}(\mu, \chi)_{t=k} - f_{q,t}(\mu, \chi)_{t=k})$, where $\Lambda = H((\mu, \chi)_{(t=k)})$, as the *potential rate of change of fluid level* on state $(\mu, \chi)_{(t=k)}$. The fluid level of $q \in P_c$, during the state $(\mu, \chi)_{(t=k)}$, changes continuously as:

$$\frac{d\chi_q}{dt} = 0, \text{ if } (b_q((\mu, \chi)_{(t=k)}) = 0) \vee ((\chi_{(t=k)}(q) = b_q((\mu, \chi)_{(t=k)})) \wedge (\delta_q^{\text{pot}}((\mu, \chi)_{(t=k)}) > 0)) \vee (\chi_{(t=k)}(q) = 0 \wedge \delta_q^{\text{pot}}((\mu, \chi)_{(t=k)}) < 0). \quad (11)$$

$$\frac{d\chi_q}{dt} = \delta_q^{\text{pot}}((\mu, \chi)_{(t=k)}), \text{ to the other cases.} \quad (12)$$

The stochastic evolution of the FSPN markings follows the same dynamic execution rules of a SPN, unless a transition $t_i \in H((\mu, \chi)_{(t=k)})$ was disabled because $\chi_{(t=k)}(q_i)$ achieved a value before t_i fires. Moreover, the evolution of the discrete markings of the net occurs in an atomic way, in the same way as the SPN.

2.3 Improving the quantitative assessment with Reward nets

The Stochastic Petri Nets (SPN) and their extensions, such as FSPN, allow obtaining relevant system measures, such as availability, performance, reliability, safety and other measures based on time. The stochastic measures of a SPN is obtained by calculating the probability of occurrence of the system states (Bobbio, 1988), where $\pi(t) = (\pi_1(t), \dots, \pi_m(t))$, $S = \{1, \dots, m\}$, and $\pi_m(t) = P(N(t) = n)$, $n \in S$ (probability of transient state, which represents the probability of the net being in the marking n at time t , $N(t) \in \hat{S}$, $t \in R^+$). Other measures are obtained by the stochastic analysis of the model elements, such as the frequency of transition firing or the number of tokens in one place at time t . However, only these sets of measures are available to achieve with Stochastic Petri Nets (SPN) and their extensions.

A systematic way to define other measures of interest in a net other than those listed above is through the *reward function*, where the reward structures can be expressed, at a high abstraction level, by conditional marking-dependent expressions (German, 2000). The net formalism which allows specifying output measures with reward-based functions are defined as "Reward Nets". The fusion between the Stochastic Petri Nets and the reward formalism is defined as Stochastic Reward Nets (SRN) (Muppala et al., 1994). In the stochastic process level, a reward structure is defined by a *reward rate vector* "rr" (where $rr_n \subset rr$, state $n \in S$ and S is the net state space), and an *impulsive reward vector* "ri" (where $ri_n \subset ri$, transition $g \in T$ (transition set of net) in state $n \in S$) (German, 2000). As was presented before, 3 reward variables could be defined in a given reward structure: instantaneous reward (equation 2), accumulated reward (equation 3) and average reward (equation 4) variables. Considering Equations (2)(3)(4):

$$R_{\text{inst}}(t) = \sum_{n \in S} rr_n \cdot 1_{\{N(t)=n\}} + \sum_{g \in T} \sum_{n \in S} ri_n^g \cdot 1_{\{g \text{ fires at } t \text{ in state } n\}}, \delta \text{ is an instantaneous reward}$$

variable at time t , where $1_{\{\text{predicate}\}}$ returns 1 if " $\{\text{predicate}\} = \text{TRUE}$ ", and returns 0 in the other cases; $N(t) \in S$ is the stochastic process defined to the net in assessment; δ is a Dirac

impulsive function, where $\delta(t - \tau) = 0$ to $\forall t \neq \tau$ e $\delta(t - \tau) = \infty$ to $t = \tau$, where $\int_{-\infty}^{+\infty} \delta(x) dx = 1$, the area of which is given by $\mathbf{ris}_n$ at firing of $\mathbf{g}$.

$R_{acum}(t) = \int_0^t R_{inst}(\varphi) d\varphi$ is an *accumulated reward variable* at t .

$R_{aver}(t) = \frac{1}{t} \cdot R_{acum}(t)$ is an *average reward variable* at t .

The variable $R_{inst}(t)$ has no stochastic behavior due to the presence of impulses (δ) (German, 2000). However, variables $R_{acum}(t)$ and $R_{aver}(t)$ have stochastic behavior and can be characterized by their probability distributions. As mentioned by Muppala et al., (1994), the SRN output measures are expressed in terms of expectation ($E[\cdot]$) of the reward variables, defined as the amount of interest in the analysis.

Therefore, considering $E[R_{acum}(t)]$ as the expectation of *accumulated reward variable* $R_{acum}(t)$ (German, 2000):

$$E[R_{acum}(t)] = \sum_{n \in S} rr_n \cdot \int_0^t \pi_n(x) dx + \sum_{g \in T} \sum_{n \in S} ri_n^g \cdot \int_0^t \varphi_n^g(x) dx \quad (13)$$

Where $\pi_n(t) = P(N(t)=n)$, $n \in S$, is the probability of transient state, which represents the probability of the net being in the marking n at time t , given that $E[1_{\{N(t)=n\}}] = \pi_n(t)$; and

$\int_0^t \varphi_n^g(x) dx = E$ [number of fires of $\mathbf{g} \in T$ at marking $\mathbf{n} \in S$ and during the period $[0, t]$].

Considering $E[R_{aver}(t)]$ as the expectation of the *average reward variable* $R_{aver}(t)$:

$$E[R_{med}(t)] = E\left[\frac{1}{t} \cdot R_{acum}(t)\right] = \frac{1}{t} \cdot E[R_{acum}(t)] \quad (14)$$

For the *instantaneous reward variable* $R_{inst}(t)$, it is possible to calculate its expectation if $R_{inst}(t)$ does not contain impulses (δ). Therefore, obtaining $E[R_{inst}(t)]$ through $R_{acum}(t)$, we have:

$$E[R_{inst}(t)] = \frac{d}{dt} E[R_{acum}(t)] = \sum_{n \in S} rr_n \cdot \pi_n(t) + \sum_{g \in T} \sum_{n \in S} ri_n^g \cdot \varphi_n^g(t) \quad (15)$$

So as to simplify the equating, the definition of the $\mathbf{ris}_n$ (impulsive reward vector) can be suppressed from a net under consideration. Thus, making $\mathbf{ris}_n = 0$, $\forall \mathbf{g} \in T$, $\forall \mathbf{n} \in S$, we have:

$$R_{inst}(t) = \sum_{n \in S} rr_n \cdot 1_{\{N(t)=n\}} \Rightarrow E[R_{acum}(t)] = \sum_{n \in S} rr_n \cdot \int_0^t \pi_n(x) dx \quad (16)$$

And the other variables are calculated as a consequence of the definition above.

The analysis of a net using reward equations requires the calculation of its probability of state $\pi_n(t) = P(N(t) = n)$, $n \in S$. Redefining it as the vector $\pi(t) = (\pi_1(t), \pi_2(t), \dots, \pi_m(t))$, $S = \{1, 2, \dots, m\}$, if $\pi = \lim_{t \rightarrow \infty} \pi(t)$ exists, where π is considered the *limit state probability*. If this probability

does not exist, the definition $\pi_n = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \pi_n(x) dx$ can be used, where π_n is the time-

averaged limit probability to a state $n \in S$. However, if both π and π_n exist, both probabilities have the same value, and both can be interpreted as the fraction of time in which a process remains in a state when it is observed for a long period of time (German, 2000).

Based on the presented reward structure and on the possible calculations over the state probabilities, both the transient analysis as the stationary analysis can be performed on nets. In the transient analysis, the state of the system can be observed at specific instants of time. In stationary analysis, the behavior of the system can be obtained when it is in stable operation (steady-state behavior).

2.4 Resolving FSPN models with Discrete Event Simulation

2.4.1 Analytical Techniques versus Discrete Event Simulation

As could be noticed before, obtaining quantitative measurements for Stochastic Petri Nets (and their evolutions) involves calculating the probability of the state vector $\pi_n(t) = P(N(t)=n)$. There are 2 different techniques for solving these nets: Analytical Techniques and Simulation (Computer Science Department, 2000).

Using analytical techniques for the solution of Stochastic Petri Nets (SPN) assumes that the nets have Markovian characteristics (memoryless systems), which restricts the use of transitions with exponential and geometric distribution functions to the nets. To these memoryless nets, it is possible to transform them into a Continuous Time Markov Chain (CTMC) or into a Discrete Time Markov Chain (DTMC), and solving them using the properties of the Markov theory, e.g., calculating the probability of the state equations using its transition probability matrix P , where $\pi(k) = \pi(0) \cdot P^k$ (considering a DTMC) or $\pi(t) = \pi(0) \cdot e^{Pt}$ (considering a CTMC) (German, 2000).

In addition to the restrictions of the types of transition used in SPN, the modeling of complex and non limited nets may result in an excessive number of states, making the computational resolution of their state equations unfeasible and forcing the application of numerical approaches and simplifications to the Analytical Technique. Related to this states space problem is the resolution of the Fluid and Stochastic Petri nets (FSPN) by analytical techniques. The proposed techniques in Trivedi & Kulkarni (1993) and Horton et al (1998) are feasible only to certain strict conditions of FSPN. Because their reachability and characteristics of their decision processes, FSPN of up to 2 dimensions can be calculated (Asarin, Maler & Pnueli (1995) apud Ciardo, Nicol & Trivedi (1999)).

The *simulation* technique is a feasible solution to the resolution of non-Markovian Petri nets, as well the solution of nets the complexity of which makes its state space unsolvable by analytical methods. Since Stochastic Petri Nets (SPN) is a method used to describe the dynamics of discrete event systems, the evolution of states (markings) of which occurs in well defined and discrete events, its resolution by simulation, mainly *discrete event simulation*, becomes an extension of this method.

According to Gordon (1978), the analysis of a system by Discrete Event Simulation (DES) is modeled by a set of elements called State Descriptors (D_e) - which represents the system states ($d_i \in N, \forall i \in D_e$) - and by an element that represents the *simulation time* (t), so that $t \in T_s$ and $T_s \in [0, T_{MAX}]$, the goal of which is representing the time evolution of the simulation and identifying the moment in which changes in the system state occur.

The state descriptors values ($d_i, \forall i \in D_e$) are changed in discrete events (e_n), so that $e_n \in E$ and $n = \{1, 2, \dots\}$, where E is the set of conditions that promote immediate change in one or more values of the state descriptors (d_i). The simulation dynamics occurs by changes in the values of state descriptors (d_i) for each event (e_n) runned until the time specified for simulation (T_{MAX}) is fully covered or until there are no more conditions for the occurrence of events.

The system events occur in chronological order, generating a sequence of executions $(e_0, e_1, \dots, e_n)$ at instants $(t = t_0, t_1, \dots, t_n)$, where $(t_0 \leq t_1, t_1 \leq t_2, \dots, t_{n-1} \leq t_n \leq T_{MAX})$.

Table I shows the equivalence between the Discrete Event Simulation (DES) and Stochastic Petri nets (SPN) concepts, considering the characteristics of time and state for a modeled system.

	Discrete Event Simulation (DES)	Stochastic Petri nets (SPN)
<i>State</i>	<i>State Descriptor values</i> $(d_i \in N, \forall i \in D_e)$	<i>Marking on nets</i> $(\mu(k) \in N, \forall k \in P)$
<i>Time</i>	<i>Simulation Time ($t \in T_s$)</i> <i>(instants $t_0, t_1, \dots, t_n$)</i>	<i>Firing time ($e_0, e_1, \dots, e_m$) to the transitions</i> <i>($t_0, t_1, \dots, t_n$)</i>

Table I – Equivalence between DES and SPN models

As an example, given a SPN with initial marking μ_0 , which occurs in the firing sequence $(t_0, t_1, \dots, t_n) \in \sigma$, promoting the sequence of markings $(\mu_0, \mu_1, \dots, \mu_n) \in \mu$ and the spent time in each marking is $(\tau_{t_0,0}, \tau_{t_1,1}, \dots, \tau_{t_n,n})$, corresponding to the notation $(s_0, s_1, \dots, s_n) = ((\mu_0, \tau_{t_0,0}), (\mu_1, \tau_{t_1,1}), \dots, (\mu_n, \tau_{t_n,n}))$, the SPN for a DES could be written as:

<u>SPN dynamics</u>	<u>Discrete Event Simulation</u>
$(s_0, s_1, s_2, \dots, s_n) =$	.
$= ((\mu_0, \tau_{t_0,0}), (\mu_1, \tau_{t_1,1}), \dots, (\mu_n, \tau_{t_n,n})) =$	.
$= ((\mu_0, e_0), (\mu_1, e_1), \dots, (\mu_n, e_n)) =$	.
$= ((\mu_0, 0), (\mu_1, \tau_{t_1,0}), \dots, (\mu_n, \sum_{i=0}^{n-1} \tau_{t_i,i})). \rightarrow$	$((d, t_0 = 0), (d, t_1), \dots, (d, t_n)).$

Given that the system state at instant t , modeled for Discrete Event Simulation (DES) assessment, is represented by the set of values of state descriptors $(d = (d_0, d_1, \dots, d_n))$ in an instant $t \in T_s$, resulting in the 2-upla (d, t) , the sequence of markings μ_k obtained by the SPN running in the instants of time e_k is equivalent to the evolution of the values of d occurred in the instants t_k , where $((\mu_0, 0), (\mu_1, e_1 = 0 + \tau_{t_0,0}), \dots, (\mu_n, e_n = e_{n-1} + \tau_{t_{n-1},n-1})) = ((d, t_0), (d, t_1), \dots, (d, t_n))$.

For a Petri net, the set of conditions (E) to the Discrete Event Simulation model is obtained by means of the execution rules related to that Petri net. As an example, the discrete event simulation process to a DTMC (Discrete Time Markov chain) $\{N(k) = s \mid s \in S = \{s_1, s_2, \dots, s_n\}, k \in N\}$ with a transition matrix P can be executed, according to Ripley (1987):

If $N(n) = s$, choose $N(n+1) = i$ using the discrete distribution $\{p_{s,i} \in P \mid i \in S\}$.

In the case of a Stochastic Petri net modeled by transitions with any probability distribution, it can be simulated by means of the following algorithm:

1. Given (μ_k, e_k) , get $t_i \in H(\mu_k), \forall t_i \in T$.
2. Sample τ_{t_i} to $\forall t_i \in H(\mu_k)$ by means of their probability distribution functions.

3. Respecting the net execution rules, make $\delta(\mu_k, t_{ik}) = \mu_{k+1}$ and $e_{k+1} = e_k + \tau_{ti}$.
4. Return to step "1" and repeat the procedure to the next reached state (μ_{k+1}) and while $\exists t_i \in H(\mu_{k+1}), \forall t_i \in T \vee e_{k+1} \leq T_{MAX}$.

The temporal representation of a Discrete Event Simulation (DES) process (the value of the *simulation time* element) may change according to 2 different methods (Gordon, 1978):

Event-oriented: the value of the *simulation time* advances to the instant in which the next detected event happens. This process is known as the *next event simulation*, due to both the next state and its time of occurrence are simulated from the current state of the model.

Interval-oriented: the value of *simulation time* advance at time intervals (usually uniform and short intervals), and where the existence of a condition is verified for a migration of state.

According to Gordon (1978), each method of temporal evolution in DES is appropriate for a particular type of system to be simulated. The event-oriented method is recommended for the simulation of purely discrete systems, such as the systems modeled by Stochastic Petri nets. On the other hand, the event-oriented method is recommended for the simulation of purely continuous systems, due to the evolution of states in the time being modeled by differential equations.

2.4.2. Simulation of Continuous-Discrete (hybrid) models

In addition to the purely discrete systems and the purely continuous systems, there are systems in which the mechanism of change of states regarding time occurs in a discrete-continuous way. The Fluid and Stochastic Petri Nets (FSPN), as defined above, are a method for modeling and evaluating continuous-discrete systems. In these systems, the states can change either in a continuous way, in a discrete way or in a continuous way added by discrete jumps.

The process of simulating continuous-discrete systems depends on the modes of interaction between the continuous and the discrete model variables. According to Computer Science Department (2000), there are 3 possible modes of interaction between $\mu(p)$ and $\chi(q)$, where $p \in P_d$ and $q \in P_c$ in a FSPN:

1. Discrete variations in $\chi(q)$ dependent on $\mu(p)$: continuous variables ($\chi(q)$) can receive discrete changes in their values. In the FSPN, this is equivalent to modelling fluid impulses, represented by arcs A_c .
 2. Variations in $\chi(q)$ and/or $\mu(p)$ dependents of $\chi(q)$: an event that changes both discrete variables ($\mu(p)$) and continuous ($\chi(q)$) may be executed if the value of continuous variables reach pre-defined levels. In FSPN, this is equivalent to modelling (fluid) marking-dependent *guards* for immediate transitions ($g:((q \in P_c) \rightarrow \{1, 0\})^{\#Ti}$).
 3. Continuous variation rate in $\chi(q)$ dependent on $\mu(p)$: a discrete event in time can promote the change in the functional description of continuous variables. In FSPN, this is equivalent to modelling marking-dependent *fluid rates* ($f=((P_c \times T_T) \cup (T_T \times P_c)) \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$).
- 2 types of events can promote those possible interactions between continuous and discrete variables in a continuous-discrete system:

- a. Time events: events that occurs in specific instants of time.
- b. State events: events that occur when a determined state condition is satisfied.

As can be noticed, the Discrete Event Simulation process for Fluid and Stochastic Petri nets requires an approach that considers both their time events and their state events. The time

events are related to the event-oriented method of temporal representation, and its execution dynamics makes use of simulation queues, where events are scheduled for future executions. On the other hand, the state events are related to the interval-oriented method of temporal representation, in which the time events are monitored during the simulation, without being queued (Computer Science Department, 2000).

A FSPN can be solved using DES since its execution rules be respected. For example, given a marking $(\mu, \chi) \in \hat{S}$, the content of fluid places $\chi_{(t)}(q_i)$, $q_i \in P_c$, can change while $t \in H((\mu, \chi)_{(t)})$, $t \in T$. If $t \in T_T$ and $(t, q_i) \subseteq f$, the value of $\chi_{(t)}(q_i)$ changes during the interval $[t, t+\tau)$ in which $t \in H((\mu, \chi)_{(t+\tau)})$ through the system of differential equations:

$$\forall q \in P_c, \frac{d\chi_q(\tau)}{dt} = \sum_{t \in \Lambda} (f_{t,q}(\mu, \chi)_{t=k} - f_{q,t}(\mu, \chi)_{t=k}) \quad , \text{ where } \Lambda = H((\mu, \chi)_{(t)}) \quad (17)$$

The initial condition $(\mu, \chi)_{(t)}$ is unchanged during the interval $[t, t+\tau)$, and the system perceives that the continuous part of the state has changed only at the end of this time range, where $t \notin H((\mu, \chi))$. Due to this fact, the events of state, especially those dependent on $\chi(q)$ are related to the interval-oriented method of temporal representation. By moving up the time in small steps, it is possible to maintain a significantly high update rate to $\chi(q)$ values (inversely proportional to the time step), reducing the latency between the occurrence of a “real condition” of state change and its detection by the process of state monitoring of the simulation, or even the loss of the desired conditions.

Therefore, during the simulation execution, the resolution process of the systems differential equations requires to know the evolution of $H((\mu, \chi)_{(t)})$ (transitions enabled at time t) in function of time. Thus, $H((\mu, \chi)_{(t)})$ can have the following interdependencies to the model states (Ciardo, Nicol & Trivedi, 1999): $H((\mu, \chi)_{(t)})$ can be *independent* of $\chi(q)$ and $H((\mu, \chi)_{(t)})$ can be *dependent* on $\chi(q)$ values, $q \in P_c$.

The simplest case to handle in a DES of FSPN is when $H((\mu, \chi)_{(t)})$ is independent of $\chi(q)$, where the cardinality of the arcs and conditions of guards do not depend on $\chi(q)$. Thus, the discrete part of the model evolves through the execution of discrete events, and the continuous part can evolve independently, updating the values of $\chi(q)$ only in the discrete instants e_n and based on the time difference between the current event and the previous event (τ). This is equivalent to using only the event-oriented method of temporal representation on the system simulation. The simulation process becomes complex in case $H((\mu, \chi)_{(t)})$ depends on $\chi(q)$, the arcs cardinality and the conditions of guards of which are in function of $\chi(q)$. In this case, it is mandatory to know the future instant of time τe at which a change in $H((\mu, \chi)_{(t+\tau e)})$ occurs. At first, obtaining τe demands to know $\chi(q)$ during $[t, t + \Delta)$ by solving the differential equations of the model, in which Δ is a future time step used to obtain the behavior of $H((\mu, \chi))$ in that period. If there is no change in $H((\mu, \chi)_{(t+\Delta)})$ during $[t, t + \Delta)$, the values of $\chi(q)$ are updated to $t + \Delta$, and the simulation continues with the search. This is equivalent to using the interval-oriented method of temporal representation in the system simulation.

2.5. Estimation of the Assessment Measures by Simulation

As mentioned earlier, it is possible to obtain multiple measures of interest in nets using reward functions; their formalization is done by means of the Reward Stochastic nets (SRN).

A reward structure is characterized by a *reward rate vector* “rr” and an *impulsive reward vector* “ri”, which produces 3 different reward variables, $\mathbf{R}_{\text{inst}}(t)$, $\mathbf{R}_{\text{acum}}(t)$ e $\mathbf{R}_{\text{aver}}(t)$, where it is possible to obtain their expectations ($\mathbf{E}[\cdot]$) used for analysis.

Using analytical methods, it is possible to calculate the expectations $\mathbf{E}[\mathbf{R}_{\text{inst}}(t)]$, $\mathbf{E}[\mathbf{R}_{\text{acum}}(t)]$ and $\mathbf{E}[\mathbf{R}_{\text{aver}}(t)]$ by means of their literal expressions, but this could be not feasible in some cases and, therefore, the simulation technique is required. However, the use of the simulation technique can only infer (or estimate) the values of these statistical parameters. This limitation to obtain the desired measures of interest do not invalidate the assessment framework proposed here, but the framework must incorporate concepts of statistical inference to the assessment process as a means to calculate the desired measures with significant confidence, as illustrated in Fig. 2.

Therefore, it is necessary to have good statistical estimators to calculate values of Equations

(5) and (6) by Discrete Event Simulation (DES). $\bar{X} = 1/N \cdot \sum_{i=1}^N x_i$ and $s_x^2 = \frac{1}{N-1} \cdot \sum_{i=1}^N (x_i - \bar{X})^2$ are good estimators for expectation and variance of statistical variable “X” ($\mathbf{E}[X]$ and $\mathbf{VAR}[X]$, respectively), where x_i is the i-upla sample obtained from the Discrete Event Simulation process and “N” is the size of the sample. Confidence Interval (CI) for statistical variable “X” at confidence level “ γ ”, $\gamma \in [0,1]$, is given by $[\bar{X} - t_{N-1, \frac{\gamma}{2}} \cdot s_{\bar{X}} / \sqrt{N} ; \bar{X} + t_{N-1, \frac{\gamma}{2}} \cdot s_{\bar{X}} / \sqrt{N}]$, where

$t_{N-1, \frac{\gamma}{2}}$ is a t-Student distribution (to a large sample size “N”, respecting the well-known

Central Limit Theorem).

Therefore, the expectation of variables $\mathbf{R}_{\text{inst}}(t)$, $\mathbf{R}_{\text{acum}}(t)$ e $\mathbf{R}_{\text{aver}}(t)$ could be estimated by means of estimators $\bar{\mathbf{R}}_{\text{inst}}(t)$, $\bar{\mathbf{R}}_{\text{acum}}(t)$ e $\bar{\mathbf{R}}_{\text{aver}}(t)$, where:

$$\bar{\mathbf{R}}_{\text{inst}}(t) = 1/N \cdot \sum_{i=1}^N \mathbf{R}_{\text{inst}}(t)_i, \text{ estimator to } \mathbf{E}[\mathbf{R}_{\text{inst}}(t)]. \quad (18)$$

$$(s_{\bar{\mathbf{R}}_{\text{inst}}(t)})^2 = \frac{1}{N-1} \cdot \sum_{i=1}^N (\mathbf{R}_{\text{inst}}(t)_i - \bar{\mathbf{R}}_{\text{inst}}(t))^2, \text{ estimator to } \mathbf{VAR}[\mathbf{R}_{\text{inst}}(t)]. \quad (19)$$

$$\mathbf{CI}(\mathbf{R}_{\text{inst}}(t), \gamma) = [\bar{\mathbf{R}}_{\text{inst}}(t) - t_{N-1, \gamma/2} \cdot s_{\bar{\mathbf{R}}_{\text{inst}}(t)} / \sqrt{N} ; \bar{\mathbf{R}}_{\text{inst}}(t) + t_{N-1, \gamma/2} \cdot s_{\bar{\mathbf{R}}_{\text{inst}}(t)} / \sqrt{N}] \quad (20)$$

Estimators $\bar{\mathbf{R}}_{\text{acum}}(t)$ and $\bar{\mathbf{R}}_{\text{aver}}(t)$ follow the same structure of Equations (18), (19) and (20).

$\mathbf{R}_{\text{inst}}(t)_i$ is the i-th sample obtained from variable $\mathbf{R}_{\text{inst}}(t)$ to constitute the estimator of parameter $\mathbf{E}[\mathbf{R}_{\text{inst}}(t)]$. Please note that $\mathbf{R}_{\text{inst}}(t)$ is considered a random variable when its *impulsive reward vector* “ri” is null.

As previously mentioned, these properties are valid only if $\bar{\mathbf{R}}_{\text{inst}}(t)$ is a random variable with normal distribution, where:

$$\bar{\mathbf{R}}_{\text{inst}}(t) \sim N(\mathbf{E}[\bar{\mathbf{R}}_{\text{inst}}(t)], \mathbf{VAR}[\bar{\mathbf{R}}_{\text{inst}}(t)]) = N(\mathbf{E}[\mathbf{R}_{\text{inst}}(t)], \mathbf{VAR}[\mathbf{R}_{\text{inst}}(t)]/N) \quad (21)$$

If not, the Central Limit Theorem (CLT) can be applied, which involves using a sample N as large as enough:

$$\lim_{n \rightarrow \infty} \left(\frac{\bar{\mathbf{R}}_{\text{inst}}(t) - \mathbf{E}[\mathbf{R}_{\text{inst}}(t)]}{\sqrt{\mathbf{VAR}[\mathbf{R}_{\text{inst}}(t)]/n}} \right) = Z = N(0,1) \quad (22)$$

2.6. Runs versus Simulation length

A system assessment process by simulation is related to the trade-off between the *simulation length* (T_{length}) and the number of *runs* (*Runs*) to be used, because, besides the total processing time (time of computational resources required) is given by the product $T_{length} \times Runs$, the T_{length} and *Runs* characteristics influence the values of the estimates obtained. For example, each of the N samples $R_{inst}(t)_i$ used to estimate $E[R_{inst}(t)]$ by means of estimator $\bar{R}_{inst}(t)$ are obtained by means of N independent *Runs* of length $t = T_{length}$ each. The process where samples are obtained through the execution of N simulation runs of length T_{length} , is named Replication. In this case, the following considerations can be made regarding the possible configurations of the simulation used to obtain an estimate of $E[R_{inst}(t)]$:

- Using a large simulation length (T_{length}) and a small number of *Runs*, the influence of initial condition $((\mu, \chi)_{(0)})$ over the estimative is eliminated. This is useful in the stationary (steady-state) analysis of systems. However, the small number of samples ($Runs = N$) undertakes the calculation of the estimation confidence interval (CI), as shown before.
- The CI of the estimate is improved, without burdening the total simulation time, increasing the number of *Runs* and reducing the length of the simulation (T_{length}). However, this fact makes the initial condition $((\mu, \chi)_{(0)})$ have strong influence over the estimative, and it is useful in the transient analysis of the systems.

The process of estimation of parameters is only useful when it is possible to get an estimate with a satisfactory confidence interval, which requires a large number of samples (*Runs*). However, as it could be seen, a large number of *Runs* increases the processing time of the simulation. Since a stationary analysis requires a large simulation length (T_{length}), performing stationary (steady-state) analysis on systems can be computationally unfeasible. Hence, when replication is applied in stationary analysis, in which it is not possible to increase T_{length} or *Runs* indefinitely, it is recommended to initiate the model $((\mu, \chi)_{(0)})$ in a state closer to the stationary state so as to minimize the affectation of the initial condition over the estimatives obtained (Computer Science Department, 2000).

3. Case study: assessing risks in critical systems using FSPN

This chapter proposed an assessment framework that could deal with the characteristics found in the current paradigm of safety critical systems. After presenting the fundamental concepts of the framework proposed, this item presents, as a case study, its implementation in a real world safety critical system that has the characteristics of the paradigm presented: the Air Traffic Control system (ATC) under the CNS/ATM concept (Communication, Navigation, Surveillance / Air Traffic Management), especially using ADS-B (Automatic Dependent Surveillance by Broadcast) and GNSS (Global Navigation Satellite System) elements such as navigation and surveillance, respectively. This case study presents the main steps for the implementation of the framework, as the modeling of the system by FSPN and the definition of metrics of interest by reward nets, illustrating and validating the proposed framework. The details of this case study, as well as the definition of CNS/ATM and other system details were previously presented in detail in Vismari & Camargo Junior (2008).

As defined in ICAO (2007), CNS/ATM is "... employing digital technologies, including satellite systems together with various levels of automation, applied in support of a

seamless global air traffic management system". Applying this new paradigm will transform the current air traffic system, based on voice communications (Radio) and independent surveillance (Radar), into a large-scale distributed digital network. The CNS/ATM provides significant technological advances to communication, navigation and surveillance air traffic system elements to support a seamless global air traffic management system. Its key-elements are based on a global navigation satellite system (GNSS) – that will support navigation functionalities to aircrafts as well as surveillance information to air traffic control – and, mainly, based on an Aeronautical Telecommunication Network (ATN) – that will support efficient and safe digital communications among all the system users. One of the ATN air-ground applications is the Automatic Dependent Surveillance by Broadcast (ADS-B), which provides "ADS reports" (aircraft containing aircraft position and other information reports required for aircraft flight management) to networks end users (e.g. controllers, pilots) (ICAO, 1999). ADS-B is to be applied as a surveillance system to air traffic control (ATC) service in areas where installing Radar equipments is unfeasible (e.g. ocean), and, besides that, ADS-B will complement surveillance capabilities in areas where Radar service is available.

The ADS-B is a fusion of communication (ATN) and navigation (GNSS) system elements (and their functionalities), being responsible for the surveillance system communication process. It is composed of a transmission element (responsible for generating and sending ADS reports), a data link (communication path) and a reception element (responsible for receiving, processing and displaying surveillance information to the end application). The data contained in the ADS-B report come from the navigation system (that will be at most represented by the GNSS in the CNS/ATM paradigm) obtained from the onboard aircraft navigation equipment.

Since ADS-B has caused such radical changes in the surveillance concept, it is necessary to verify if ADS-B could be used in air traffic control systems (ATC) – the most critical level of the air traffic management system. The ATC main responsibility is to apply and maintain the safe separation between aircrafts within its delegated space sector. Therefore, follow the application of the proposed framework to assess the ADS-B based ATC system.

3.1 Modelling the ATC system with FSPN

In this case study, the functional elements considered (modelled) to assess risks in Air Traffic Control systems (ATC) were: *Airspace* (route configuration, flight plans and so on), *Aircrafts* (flight dynamic), *Navigation* (including *Pilot* characteristics), *Communication*, *Surveillance* and *Air Traffic Controller* (ATCo). Navigation was based on GNSS and surveillance was based on ADS-B. Fig. 3 shows, at a high abstraction level, the architecture of the ATC systems considered in this work (not including Airspace).

Fig. 3. The ATC system architecture

The parameters considered for each ATC system element are:

- Regarding *Airspace*, it is illustrated in Fig. 4. The route structure was considered in a Cartesian (non-spherical) plan $((X,Y) \in (R^+)^2)$ containing unidirectional routes, A and B, at the same flight level (altitude) and converging at a Θ angle ($\Theta=45^\circ$) to the notification point O_z (with well-known coordinates $O_z=(X_{NOT(z)}, Y_{NOT(z)})$). Pairs of aircrafts, A_1 and A_2 , were planned to conduct flights adhering to routes A and B, respectively, at the same flight level and at the same velocity (280kts). In those flights, the minima separation planned between aircrafts A_1 and A_2 (D_{min}) occurs when A_1 aircraft reaches the notification point O_z (A_2 following A_1).

Fig. 4. The Airspace element

- Regarding *Aircraft*, three main flight parameters were considered for each one ($i=1,2$): position ($P_i(t)$), Velocity ($V_{TAS,i}(t)$) and Heading ($\psi_i(t)$). Then, the state of each aircraft at time t is defined as $A_i(t)=\{P_i(t), V_{TAS,i}(t), \psi_i(t)\}$. The aircrafts were responsible for following their own flight plans, controlling their $\psi_i(t)$ using onboard navigation systems (*Autopilot* and *Flight Management Computer* (FMC)) and obeying the ATCo instructions (reconfiguring *autopilot* with instructed resolution heading ($\Psi_{ATCo,i}$), in case of resolution manoeuvres.

- Regarding *Air Traffic Controller (ATCo)*, its *availability to detect conflicts* (A_{ATCo}) and the time spent to make a *decision* ($T_{decision}$), as well as the *resolution communication* (T_{comm}) to aircrafts involved in the conflict, were taken into consideration. A conflict resolution process (*detection-resolution-communication*) begins after ATCo observes, at least, " $N_{ConfATCo}$ " successive aircraft target pairs on the surveillance screen with separation lower than D_{ATCo_min} (minima separation standard). A conflict resolution process is concluded when heading the resolution communication to each aircraft pilot (Ψ_{ATCo1} and Ψ_{ATCo2}). All times involved were represented by statistical variables with Normal distribution ($N(\mu, \sigma^2)$).
- Regarding *Pilots*, the time spent by pilots to receive and acknowledge the resolution heading ($\Psi_{ATCo i}$) communicated by the ATCo and to re-configure its value at the aircraft autopilot were considered. This time was represented by the statistical variable T_{comm} at the ATCo model.
- Regarding *Navigation*, the aircraft position was obtained by its own GNSS equipment onboard. The considered parameter was the *GNSS accuracy* ($\varepsilon_{GNSS0} \sim N(0.0; 44.1m)$ per dimension).
- Regarding *Surveillance*, the aircraft positions applied to ATCo was obtained by the onboard GNSS receiver (of navigation system) and transmitted/received by the ADS-B system. The considered statistical parameters were *scan rate* ($\tau_{scan, i} \sim 14s @ 95\%$) and *screen presentation latency* ($\tau_{lat, i} \sim 1.2s @ 95\%$), both with exponential distributions.

The model in FSPN of each ATC system element is:

Aircraft Model: Fig. 5 illustrates the FSPN model representing aircraft parameters $P_i(t)$ and $\psi_i(t)$ (one model to each aircraft). $V_{TAS, i}(t)$ is constant ($V_{TAS, i}$). Marking in place $K\psi$ ($\mu(K\psi)$) represents the heading in a $\psi_i = \alpha_i + (\mu(K\psi) - M) \cdot \Delta\psi$ form, where M is the number of heading steps ($\Delta\psi$). Firing time distribution of transitions τ_{ψ_I} and τ_{ψ_D} are given by $\tau_{\psi} = (\Delta\psi \cdot ||V_{TAS}||) / (g \cdot \tan(\phi))$. Their firing events are enabled by guards $g(\tau_{\psi_I})$ and $g(\tau_{\psi_D})$, which are controlled by the aircraft navigation system. The aircraft position is given by continuous variables $P_i(t) = (X_i(t), Y_i(t))$. The position dynamics were implemented by fluid arcs fxi , gxi , fyi and gyi , and their cardinalities were given by $B_{X_i}(\mu) = V_{TAS, i} \cdot \sin(\alpha_i + (\mu(K\psi_i) - M_i) \cdot \Delta\psi_i)$ and $B_{Y_i}(\mu) = V_{TAS, i} \cdot \cos(\alpha_i + (\mu(K\psi_i) - M_i) \cdot \Delta\psi_i)$.

Fig. 5. Aircraft Model in FSPN

ATCo Model: Fig. 6 illustrates the FSPN model representing the Air Traffic Controller (ATCo) parameters. Immediate transition " $t_Conflict_occured$ " is enabled to fire by guard $g(t_Conflict_occured)$ when $||P_{SUR,1}(t) - P_{SUR,2}(t)|| < D_{ATCo_min}$ (guard dependent of Surveillance model). The guard $g(t_NRes_conflict)$ enables " $t_NRes_conflict$ " (transition

with higher priority than t_{detected}) to fire when N_{ConfATCo} successive conflicts does not occur.

Fig. 6. ATCo Model in FSPN

Airspace Model: Airspace was modelled by variables used as input to other model elements. In this way, the airspace model was represented by $\{X_{\text{NOT}(z)}, Y_{\text{NOT}(z)}, D_{\text{min}}, \Theta, V_{\text{TAS},i}\}_{[\text{Airspace}]}$.

Navigation Model: the navigation system was modelled with 3 functionalities: i. *Obtaining aircraft position* (represented by GNSS navigation position ($P_{\text{NAV}}(t)$)); ii. *Managing flight plan* (represented by FMC, indicating to the autopilot the desired flight parameters ($(V_{\text{TAS}}, \psi)_{\text{desired}}$) based on the flight plan parameters and on the $P_i(t)$); iii. *Aircraft monitoring and control* (represented by the autopilot, which receives ψ_{desired} from the FMC and controls $\psi_i(t)$ according to the aircraft roll angle (ϕ)). Their models, presented in Fig. 7, represent 3 states in dynamic aircrafts heading: “Maintaining Heading”, “Decreasing Heading” and “Increasing Heading”. Heading correction algorithm applies over the aircraft a roll angle value (ϕ) based on $||\psi_i(t) - \psi_{\text{desired}}||$. Heading correction behaviour is represented by logical conditions at transition guards in navigation model, which defines its duration, when the number of marking in navigation places is not zero ($\mu(\text{navigation places}) \neq 0$) and, consequently, it defines the duration to be enabled by guards $g(\tau_{\psi_I})$ and $g(\tau_{\psi_D})$ for each aircraft heading models. Algorithms modelling this navigation functionality were implemented at $g(T_{\text{Inc_Head}_i})$ and $g(T_{\text{Dec_Head}_i})$, both calculated only when $\mu(\text{Mant_Head}_i) > 0$. GNSS algorithm obtains $P_{\text{NAV}}(t_n) = (X(t) + \varepsilon x_{\text{GNSS}}, Y(t) + \varepsilon y_{\text{GNSS}}) = (X_{\text{NAV}}(t), Y_{\text{NAV}}(t))$, where $\varepsilon_{\text{GNSS}}$ is the position accuracy of the GNSS system.

Fig. 7. Navigation in FSPN

Surveillance Model: the FSPN model, illustrated in Fig. 8, represents the surveillance parameters mentioned before. T_{SCAN} and $t_{sampling}$ firing events ($\uparrow T_{SCAN}, \uparrow t_{sampling}$) represent the process of sampling targets, where $P_{SUR,i}(\uparrow T_{SCAN}, \uparrow t_{sampling}) = (X_{SUR,i}, Y_{SUR,1})$, $i=1,2$. $P_{SUR,i}(\cdot)$ are presented to ATCo when $\uparrow t_{shows_ATCo}$, T_{lat} times occur after $\uparrow T_{SCAN}$ and $\uparrow t_{sampling}$ events. The firing time distributions are stochastic, and there is one model embedded for each aircraft. The source of information ($P_{SUR,i}(t_n) = P_{NAV,i}(t_n)$) provides the same position data obtained by the navigation system).

Fig. 8. Surveillance Model in FSPN

3.2 Defining metrics of interest by Reward Nets

In this risk assessment study, it was evaluated the ATC system ability to detect and solve the events of loss of minima separation between aircrafts that could affect air traffic safety (defined as "AIRPROX") based on its own elements. Toward it, two environment processes were observed: the real process (the real aircrafts states) and the monitored process (aircraft states observed by surveillance elements). The AIRPROX events were evaluated over the real process, verifying in which situations those events could occur due to the inefficiency of ATC elements and functionalities. In the original study (Vismari & Camargo Junior, 2008), a set of 6 metrics concerning AIRPROX were adopted. Some of them – related to the reward net concept – are presented next:

$T_{L_{MIN},i}$: exposure time to real separation losses lower than a value L_{MIN} .

$$T_{L_{min_i}} = 1_{\{\|P_1(t) - P_2(t)\| < (L_{min} + "i" \cdot 0,5NM)\}} \text{ , where } i = 0, 1, 2 \dots \text{ .} \quad (23)$$

T_{PS_INT} : exposure time to real separation losses lower than D_{ATCo_min} per ATCo intervention.

$$T_{PS_INT} = \frac{1_{\{\|P_1(t) - P_2(t)\| < D_{min_ATCo}\}}}{\#(\uparrow t_{\text{detected}})} \quad (24)$$

N_{PS_Interv} : number of conflicts presented in the surveillance screen per ATCo intervention.

$$N_{PS_Interv} = \frac{\#(\uparrow t_{\text{Conflict_occured}})}{\#(\uparrow t_{\text{detected}})} \quad (25)$$

An estimation to each adopted metrics was obtained by Discrete Event Simulation considering both the *functional conditions*, using sensitivity analysis at $D_{ATCo_min} \in [4.4; 5.8]$ NM and the *degraded conditions of GNSS position accuracy* (ε_{GNSS0}) as well as the *ADS-B scan rate* ($\tau_{scan,i}$). In the simulation process, and for each parameter configuration, 10,000 aircraft approach scenarios, with 3105 seconds length each, were run. The results obtained in this case study, as well as the risk analysis performed on them, are illustrated in Vismari & Camargo Junior (2008).

4. Conclusions

This chapter demonstrated that the Fluid Stochastic Petri Nets (FSPN) is a good graphic-mathematical formalism for modeling and assessing the current paradigm of safety critical systems. In addition to its ability to formally represent systems both by its mathematical expressions and by its graphical formalism, the FSPN allows the modeling of complex hybrid systems - the continuous and discrete elements of which have high interdependence - and with stochastic temporal characteristics. Furthermore, the existence of the Reward Structures in the FSPN formalism allows assessing complex hybrid systems based on a variety of metrics, defined during the modeling process, allowing its use in various fields of application. The use of Discrete Event Simulation enables the analyst modeling transitions on FSPN with the widest range of probability distributions, not limited to the Markovian paradigm.

The problem found in the assessment framework proposed in this chapter, in particular in the solving method using discrete event simulation, is related to the conflict between "cost in processing time" and "confidence intervals of results". For example, in the case study, we modeled the system using 40 discrete places, approximately 5 fluid places, 80 discrete arcs, 10 fluid arcs, 50 discrete transitions and a huge number of guard functions. To obtain each metric in a specific system configuration, approximately 2.8 hours of computer processing were necessary in a 2.4GHz Intel Dual Xeon with 2.5 GB RAM, using a time interval of 0.5 seconds for the discretization of the continuous net, and using the SPNP - Stochastic Petri Net Package (Trivedi, 1999) as the FSPN computational tool. Even with the high processing cost, the use of the FSPN formalism is highly recommended for the proposed Assessment

Framework, especially when used together with Reward Nets, Discrete Event Simulation and Statistical Inference. Moreover, this high processing cost can motivate new researches focused on improving the performance of the simulation methods.

At last, the Assessment Framework proposed in this chapter was intended to be applied in safety assessment works. However, it could be applied to other kinds of systems to evaluate characteristics such as cost-benefit analysis, availability, reliability and others, depending on the the metrics adopted. To validate these new applications, however, it is necessary to continue this research study.

5. Acknowledgments.

The authors would like to thank Dr. Kishor Trivedi of Duke University for kindly granting us and providing all the support for his computational tool “Stochastic Petri Nets Package” (SPNP), and the Advanced Scientific Computing Laboratory of the University of São Paulo (LCCA/USP) for providing the computing infrastructure applied in the simulation part of this work.

6. References

- Ahlström, K. & Torim, J. (2002). Future Architecture of Flight Control Systems. *IEEE AESS Systems Magazine*, Vol.17, Issue 12, (December, 2002) 21-27, ISSN 0885-8985.
- Asarin, E., Maler, O. & Pnueli, A. (1995). Reachability Analysis of Dynamical Systems Having Piecewise-Constant Derivatives. *Theoretical Computer Science*, Vol.138, Issue 1, (February, 1995) 33-35, ISSN 0304-3975.
- Bobbio, A. (1988). System Modelling with Petri Nets, *Proceedings of the Ispra Course 1988*, pp. 103-143, ISBN 0-7923-0837-9, Madrid-Spain, September 1988, Kluwer, Spain.
- Ciardo, G., Nicol, D.M. & Trivedi, K.S. (1999). Discrete-Event Simulation of Fluid Stochastic Petri Nets. *IEEE Transactions on Software Engineering*, Vol. 25, n. 2, (March/April, 1999) 207-217, ISSN 0098-5589.
- Computer Science Department (2000). On the Simulation of Stochastic Petri Nets. *College of William and Mary. England*. Available at <http://citeseer.ist.psu.edu/307709.htm>
- German, R. (2000). *Performance Analysis of Communication Systems: Modelling with Non-Markovian Stochastic Petri Nets*, John Wiley & Sons, ISBN 0471492582, England
- Gordon, G. (1978). *System Simulation*, Prentice-Hall, USA
- Gribaudo, M. et al (2002). Model-Checking Based on Fluid Petri Nets for the Temperature Control System of the ICARO Co-generative Plant, *Proceedings of Safecom 2002*, pp. 273-283, ISBN 3-540-44157-3, Catania - Italy, September 2002, Springer, Heidelberg.
- Gribaudo, M. et al (2003). Fluid Petri Nets and Hybrid model-checking: a comparative case study. *Reliability Engineering and System Safety*, Vol.81, n. 3, (September, 2003) 239-257, ISSN 0951-8320.
- Hammett, R. (2003). Flight-Critical Distributed Systems: Design Considerations. *IEEE AESS Systems Magazine*, Vol.18, Issue 6, (June, 2003) 30-36, ISSN 0885-8985.
- Heck, B.S. et al (2003). Software Technology for Implementing Reusable, Distributed Control Systems. *IEEE Control Systems Magazine*, Vol.23, Issue 1, (January, 2003) 21-35, ISSN 0272-1708.

- Horton, G. et al (1998). Fluid Stochastic Petri Nets: Theory, applications and solution techniques. *European Journal of Operational Research*, Vol.105, Issue 1 (February, 1998) 184-201, ISSN 0377-2217.
- ICAO (1999). *Manual of Technical Provisions for the Aeronautical Telecommunication Network (Doc.9705/AN956)*, International Civil Aviation Organization, ISBN 92-9194-003-8, Montreal.
- ICAO (2007). *Global Air Navigation Plan for CNS/ATM Systems (Doc.9750)*, International Civil Aviation Organization, ISBN 9789291949304, Montreal.
- Knight, J.C. (2002a). Safety Critical Systems: Challenges and Directions, *Proceedings of the 24rd International Conference on Software Engineering (ICSE 2002)*, pp. 547-550, Orlando - USA, May 2002, ACM.
- Knight, J.C. (2002b). Software Challenges in Aviation Systems, *Proceedings of Safecom 2002*, pp. 106-112, ISBN 3-540-44157-3, Catanya - Italy, September 2002, Springer, Heidelberg.
- Kirner, T.G. (1997). Quality Requirements for Real-Time Safety-Critical Systems. *Control Engineering Practice*, Vol.5, n. 7, (July, 1997) 965-973, ISSN 0967-0661.
- Labeau, P.E. et al (2000). Dynamic Reliability: towards an integrated platform for probabilistic risk assessment. *Reliability Engineering and System Safety*, Vol.68, n. 3, (June, 2000) 219-254, ISSN 0951-8320.
- Leveson, N.G. & Stolzy, J.L. (1987). Safety Analysis Using Petri Nets. *IEEE Transactions on Software Engineering*, Vol.SE13, n. 3, (March, 1987) 386-397, ISSN 0098-5589.
- Leveson, N.G. (1995). *Safeware: System, Safety and Computers*, Addison-Wesley Publishing Company, ISBN 0201119722, USA
- Muppala, J.K., Ciardo, G. & Trivedi, K. (1994). Stochastic Reward Nets for Reliability Prediction. *Communications in Reliability, Maintainability and Serviceability: An International Journal*, Vol.1, n. 2, (March, 1994) 9-20.
- Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, Vol.77, n. 4, April, 1989) 541-579, ISSN 0018-9219.
- Murray, R.M. et al (2003). Future Directions in Control in an Information-Rich World. *IEEE Control Systems Magazine*, Vol.23, Issue 2, (March, 2003) 20-33, ISSN 0272-1708.
- Peterson, L.J. (1981). *Petri Net Theory and the Modelling of Systems*, Prentice-Hall, ISBN 0136619835, USA
- Philippi, S. (2003). Analysis of fault tolerance and reliability in distributed real-time system architectures. *Reliability Engineering and System Safety*, Vol.82, n. 2, (November, 2003) 195-206, ISSN 0951-8320.
- Ripley, B.D. (1987). *Stochastic Simulation*, Wiley & Sons, ISBN 978-0-470-00960, USA
- Tipsuwan, Y. & Chow, M.Y. (2003). Control Methodologies in Networked Control Systems. *Control Engineering Practice*, Vol.11, n. 10, (October, 2003) 1099-1111, ISSN 0967-0661.
- Trivedi, K. & Kulkarni, V.G. (1993). FSPN: Fluid Stochastic Petri nets, *Proceedings of the 1993 Petri Nets Conference*, Chicago-USA
- Trivedi, K.S. (1999). *SPNP User's Manual Version 6.0*, Duke University, USA.
- Tuffin, B., Chen, D.S. & Trivedi, K. (2001). Comparison of Hybrid Systems and Fluid Stochastic Petri Nets. *Discrete Event Dynamic Systems: Theory and Applications*, Vol.11, n. 1&2, (January/February, 2001) 77-95, ISSN 1573-7594.

- Varaya, P. (1999). Design Simulation and Implementation of Hybrid Systems, *Proceedings of Application and Theory of Petri Nets 1999*, pp. 1-5, ISBN 3-540-66132-8, Williamsburg-USA, June 1999, Springer
- Vismari, L.F. & Camargo Junior, J.B. (2008). An Absolute-Relative Risk Assessment Methodology Approach to Current Safety Critical Systems and its Application to the ADS-B based Air Traffic Control System, *Proceedings of 27th International Symposium on Reliable Distributed Systems*, pp. 95-104, ISBN 9780769534107, Napoli-Italy, October 2008, IEEE Computer Society, Los Alamitos
- Wolter, K. (2000). Modelling Hybrid Systems with Fluid Stochastic Petri Nets, *Proceedings of International Conference on Automation on Mixed Process: Hybrid Dynamic Systems*, pp. 287-294, Dortmund - Germany, September 2000, Shaker Verlag, Dortmund.

IntechOpen

Petri Nets Applications

Edited by Pawel Pawlewski

ISBN 978-953-307-047-6

Hard cover, 752 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

Petri Nets are graphical and mathematical tool used in many different science domains. Their characteristic features are the intuitive graphical modeling language and advanced formal analysis method. The concurrence of performed actions is the natural phenomenon due to which Petri Nets are perceived as mathematical tool for modeling concurrent systems. The nets whose model was extended with the time model can be applied in modeling real-time systems. Petri Nets were introduced in the doctoral dissertation by K.A. Petri, titled „Kommunikation mit Automaten“ and published in 1962 by University of Bonn. During more than 40 years of development of this theory, many different classes were formed and the scope of applications was extended. Depending on particular needs, the net definition was changed and adjusted to the considered problem. The unusual “flexibility” of this theory makes it possible to introduce all these modifications. Owing to varied currently known net classes, it is relatively easy to find a proper class for the specific application. The present monograph shows the whole spectrum of Petri Nets applications, from classic applications (to which the theory is specially dedicated) like computer science and control systems, through fault diagnosis, manufacturing, power systems, traffic systems, transport and down to Web applications. At the same time, the publication describes the diversity of investigations performed with use of Petri Nets in science centers all over the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Lucio Flavio Vismari and Joao Batista Camargo Junior (2010). Assessing Risks in Critical Systems Using Petri Nets, Petri Nets Applications, Pawel Pawlewski (Ed.), ISBN: 978-953-307-047-6, InTech, Available from: <http://www.intechopen.com/books/petri-nets-applications/assessing-risks-in-critical-systems-using-petri-nets>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen