

WORKFLOWS IN CONSTRUCTIONS MODELLED WITH STOCHASTIC ARTIFICIAL SOCIAL SYSTEMS

Calin Ciufudean, Constantin Filote

Stefan cel Mare University of Suceava
9 University Str., Suceava, RO-720225, Romania
calin@eed.usv.ro

Daniel Popescu

Technical University of Constructions
66 P.Protopopescu Bvd., Bucharest, Romania
dpopescu@instal.utcb.ro

Abstract: This paper focuses on evaluation of the performance characteristics of workflows in constructions modeled with stochastic Petri nets (SPN). This goal is achieved by focusing on a new model for Artificial Social Systems (ASS) behaviors. ASS exists in practically every multi-agent system, and play a major role in the performance and effectiveness of the agents. This is the reason why we introduce a more suggestive model for ASS. To model these systems, a class of Petri nets is adopted, and briefly introduced in the paper. This class allows representing the flow of physical resources and control information data of the ASS's components.

In the analysis of SPN we use simulations in respect to timing parameters in a generalized semi-Markov process (GSMP). By using existing results on perturbation (e.g., delays in supply with raw materials, derangements of equipments, etc.) analysis and by extending them to new physical interpretations we address unbiased sensitivity estimators correlated with practical solutions in order to attenuate the perturbations. An important advantage of this approach is that one simulation is needed for evaluating the stochastic Petri nets and the perturbation analysis and to take advantage of the state of the art.

Key-words: Workflow, Artificial Social Systems, Stochastic Petri nets.

1. INTRODUCTION

An Artificial Social System (ASS) is a set of restrictions on agent's behavior in a multi-agent environment [1].

ASS allows agents to coexist in a shared environment and pursue their respective goals in the presence of other agents. A multi-agent system consists of several agents, where at given point, each agent is in one of several states.

In each of its states, an agent can perform several actions. The actions an agent performs at a given point may affect the way the state of this agent and the state of other agents will change.

A system of dependent automata consists of two or more agents, each of which may be in one of a finite number of different local states. We denote the set of local states of an agent i by P_i . The set (P_1, P_2, \dots, P_n) of states of the different agents is called system's configuration.

The set of possible actions an agent i can perform is a function of the local state. For every state $p \in P_i$ there is a set $A_i(p)$ of action that i can perform when in local state p .

The row actions (a_1, \dots, a_n) denote the actions the different agents perform at a given point and is called their joint action there. An agent's next state is a function of the system's current configuration and the joint action performed by the agents. A goal for an agent is identified with one of its states. That is the reason why an agent has plans how to attain its goal.

A plan for agent i in a dependent automata is a function $U(p)$ that associates with every state p of agent i a particular action $a \in A_i(p)$. A plan [2] is said to guarantee the attainment of a particular goal starting from an initial state, in a given dependent automata system, if by following this plan the

agent will attain the goal, regardless of what the other agent will do, and what are the initial states of the other agents. A dependent automata system is said to be social if, for every initial state p_0 and goal state p_g , it is computationally feasible for an agent to devise, on-line, an efficient plan that guarantees to attain the goal p_g state when starting in the initial state p_0 . For a proper behavior, a dependent automata system is modeled with a social law. Formally, a social law Q for a given dependent automata system consists of functions (A_1, A_2, \dots, A_N) , satisfying $A_i(p) \subset A_i(p)$ for every agent i and state $p \in P_i$. Intuitively, a social law will restrict the set of actions an agent is "allowed" to perform at any given state. Given a dependent automata system S and a social law Q for S , if we replace the functions A_i of S by the restricted functions A_i , we obtain new dependent automata system. We denote this new system by S^Q . In S^Q the agents can behave only in a manner compatible with the social law T [3], [4].

In controlling the actions, or strategies, available to an agent, the social law plays a dual role. By reducing the set of strategies available to a given agent, the social system may limit the number of goals the agent is able to attain. By restricting the behaviors of the other agents, however, the social system may make it possible for the agent to attain more goals and in some cases these goals will be attainable using more efficient plans than in the absence of the social system.

A semantic definition of artificial social systems gives us the ability to reason about such systems. For example, the manufacturer of the agents (e.g., robots) that are to function in the social system will need to reason about whether its creation will indeed be equipped with the hardware and the

software necessary to follow the rules. In order to be able to reason properly, we need a mathematical model and a description language [8], [9]. We chose the stochastic Petri nets model in order to model and simulate real conditions encountered in constructions workflow planning. We shall name on further accounts this model as Stochastic Artificial Social System.

Petri nets have been recognized as a powerful tool for modeling discrete event systems. Data networks, viewed as discrete systems, are analyzed with such models. In the Petri nets theory, mathematical tools are available for analysis of the qualitative properties including deadlock-freeness, boundedness, reversibility, s.a. [1]. However simulation remains the effective for performance evaluation. Perturbation (e.g., delays in supply with raw materials, derangements of equipments, etc.) analysis has been developed for evaluating sensitivity measures by using simulations [2]. A generalized semi-Markov process (GSMP) is the usual model for the stochastic processes of discrete-event simulations, and most existing perturbation analysis methods are based on the GSMP framework. Since GSMP's and stochastic Petri nets (SPN's) have been proven to have the same modeling power [3], existing perturbation analysis methods are expected to apply to SPN's. Petri nets models considered here are SPN's with random transition firing times and the sensitivity estimators can be obtained from a simulation run. Our perturbation analysis is based on work of [5] and [6] which provides unbiased gradient estimators for a broad class of GSMP's. In this study, unbiased estimators are applied by using an appropriate SPN representation. Under correct conditioning, the unbiased estimators are easily confirmed by the simulation run of the GSP representation. This confirms the importance of underlying stochastic process. Practical solutions are shown in the paper, in order to give a concrete utilization of the theoretical model realized with SPN. The remainder of this paper is organized as follows. Section 2 presets SPN's under consideration, section 3 presents unbiased estimators for general nets, and section 4 apply the theoretical approach to a data network perturbation analysis, and explicates some practical correlations between theory and practical implementation.

2. STOCHASTIC PETRI NETS

In an ordinary Petri net $PN = (P, T, F, M_0)$, where P and T are two disjointed sets of nodes named, respectively, places and transitions. $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs. $M_0: P \rightarrow \mathbb{N}$ is the initial marking.

Two transitions t_i and t_j are said to be in conflict if they have at least one common input place. A transition t is said to be conflict free if it is not in conflict with any other transition. A transition may fire if it is enabled. A transition $t \in T$ is said to be enable at marking M if for all $p \in {}^*t$, $M(p) \geq 1$. The SPN's considered here are ordinary Petri nets with un-timed and timed transitions. In this paper we assume that timed transitions are conflict free transitions, and they are single

server transitions. Untimed transitions can be in conflict therefore we say that a marking is said to be stable if no untimed transition is enabled. In the following assumptions we assume that the initial marking is a stable marking. We note by (M, T) a stable marking reachable from M by firing t . The new stable marking M^* is obtained from M according to some routing probability. The basic idea is that in order to guarantee that a stable marking can be reached; we must ensure that the respective circuit contains at least one timed transition. A SPN can be defined by the following elements [4]:

T_t	Set of timed transitions
T_n	Set of untimed transitions
M_s	Set of stable markings
$T(M)$	Set of transitions enable at marking M
$M_s(M, t)$	Set of stable markings reachable from M by firing transition t
$p(M^*, M, t)$	Probability of reaching a stable marking M^* from M when t fires. Obviously, we have: $p(M^*, M, t) = 0$ if $M^* \notin M_s(M, t)$.
$F_t(\cdot)$	Distribution function of the firing time of $t \in T_t$

The GSMP representation of the SPN can be characterized by the following parameters:

$X(t, k)$	Independent random variables, where $t \in T_t$, and $k \in \mathbb{N}$. Each $X(t, k)$ has distribution F_t and corresponds to the time of the k^{th} firing of transition t .
$U(t, k)$	Independent uniform random variables on $[0, 1]$, where $t \in T_t$, $k \in \mathbb{N}$. Each $U(t, k)$ corresponds to the routing indicator at the k^{th} completion of t .
t_n	n^{th} completed timed transition
M_n	Stable marking reached at the firing of t_n
S_n	Completion time of t_n
τ_n	Holding time of marking M_{n-1}
$r_n(t)$	Remaining firing time of transition t at S_n
$V(t, n)$	Number of instances of t among t_1, \dots, t_n .

The dynamic behavior of an SPN can be explained in the following way: at the initial marking M_0 , set $r_n(t) = X(t, 1)$, $\forall t \in T(M_0)$ and set $V(t, 0) = 0$, $\forall t \in T_t$. All other parameters t_{n+1} , τ_{n+1} , S_{n+1} , $V(t, n+1)$, M_{n+1} , r_{n+1} can be determined recursively as usually done in discrete event simulation. Recursive equations are given in [5]. The following routing mechanism is used in GSMP:

$$M_{n+1} = \varnothing(M_n, t_{n+1}, U(t_{n+1}, V(t_{n+1}, n+1))) \quad (1)$$

Where \varnothing is a mapping such that $P(\varnothing(M,t,U) = M^*) = P(M^*,M,t)$

3. PERTURBATION PARAMETERS MODELLED WITH SPN

Following the approach given in [5], we suppose that the distributions of firing times depend on a parameter Θ . Parameters defined in section 2 are, in the above assumption, functions of Θ . In perturbation analysis the following results hold [6], where performance measures under consideration are of the form $g(M_1, t_1, \tau_1, \dots, M_n, t_n, \tau_n)$ and a shorthand notation $g(\Theta)$ is used:

a) For each Θ , $g(\Theta)$ is a.s. continuously differentiable at Θ and the infinitesimal perturbation indicator is:

$$\frac{dg(\theta)}{d\theta} = \sum_{i=1}^n \frac{\partial g}{\partial \tau_i} \cdot \frac{d\tau_i}{d\theta} \tag{2}$$

b) If $d \in [g(\Theta)]/d\Theta$ exists, the following perturbation estimator is unbiased:

$$\sum_{i=1}^n \frac{\partial g}{\partial \tau_i} \cdot \frac{d\tau_i}{d\theta} + \sum_{k=1}^n f_k(h_k) \cdot G_k \tag{3}$$

Where $f_k = \frac{f_{t_{k+1}}(L_k(t_{k+1}))}{F_{t_{k+1}}(L_k(t_{k+1}) + y_k - F_{t_{k+1}}(L_k(t_{k+1})))}$ (4)

$$y_k = \min \{r_k(t) : \forall t \in T(M_k) - \{t_{k+1}\}\} \tag{5}$$

$$\tau_k = \frac{dL_k(t_{k+1})}{d\theta} - \frac{dX(t_{k+1})}{d\theta} \tag{6}$$

$L_k(t)$ is the age of time transition t at S_k ; $G_k = g_{pp,k} - g_{DNP,k}$. The sample path $(M_1(\Theta), t_1(\Theta), \tau_1(\Theta), \dots, M_n(\Theta), t_n(\Theta), \tau_n(\Theta))$ is the nominal path denoted by NP. $g_{DNP,k}$ is the performance measure of the k^{th} degenerated nominal path, denoted by DNP_k. It is identical to NP except for the sojourn time of the $(k+1)_{th}$ stable marking in DNP_k. $g_{pp,k}$ is the performance measure of a so-called k^{th} perturbed path, denoted by PP_k. It is identical to DNP_k up to time s_k . At this instant the order of transition t_k and t_{k+1} is reversed, i.e., the firing of t_{k+1} completes just before that of t_k in PP_k. We notice that by definition, DNP_k and PP_k are identical up to s_k . At s_k , the events t_k and t_{k+1} occur almost simultaneously, but t_k occurs first in DNP and t_{k+1} occurs first in PP_k. The commuting condition given in [6] guarantees that the two samples paths became identical after the firing of both t_k and t_{k+1} . Our goal is to introduce a correction mechanism in the structure of the SPN so that the transition t_k and t_{k+1} fire in the desired order, and the routing mechanism given in relation (1) is re-established. We will exemplify this approach on an example, and we will correlate the theoretical assumption with some practical mechanisms in order to verify the approach.

4. APPLICATION TO A QUEUING NETWORK

In Fig.1, we represented a workflow queuing network. The servers are s_1, s_2 , and for any of them, if the downstream buffer is full, the customer is blocked until the downstream buffer has one hole. For simplicity of the Petri net model, we consider the perturbation analysis of only one way in the workflow [10].

In the corresponding SPN of the system in Fig.1, the transitions t_1 and t_4 model the arrivals (see Fig.2). Transitions t_3, t_6, t_7, t_9 are un-timed transitions, and are used to model the materials departure between constructors.

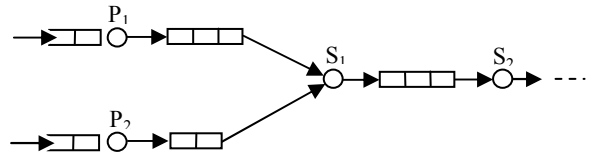


Fig. 1. A data queuing network with finite line capacity

The transitions t_2, t_5, t_8 model the service periods in the network. The holding times of the transitions t_2, t_5, t_8 in the SPN are identical to the service times of computers in the workflow. The un-timed transitions in Fig.2 are figured with bars, and the timed transitions are figures with boxes. First, in our discussion we will consider only the elements of the Petri net drawn with filled lines.

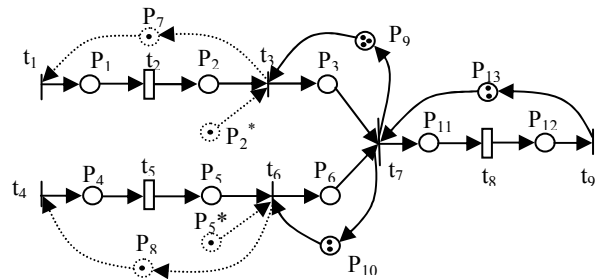


Fig. 2. The SPN model of the queuing network given in Fig.1

The information transmitted to p_{11} by firing t_7 is determined by u' (routing indicator defined in section 2, see relation (1)) when t_2 fires first and it is determined by u'' when t_5 fires first. Since u' and u'' are independent random variables, the commuting condition given in [6] does not hold (i.e., $\varnothing(\varnothing(M,t_2,u'),t_5,u'')$ can be different from $\varnothing(\varnothing(M,t_5,u''),t_2,u')$). In order to make true the commuting condition we added in Fig.2 the following elements:

- a) Locations p_2^* , respectively p_5^* and corresponding arcs play the role to ensure the desired order in firing transitions t_3 , respectively t_6 .
- b) Locations p_7, p_8, p_9, p_{10} , and p_{13} and corresponding arcs ensure a Kanban mechanism in the SPN, in order to achieve

the desired order in firing transitions t_3 and t_6 , and, for p_{13} , a delay in materials transmission to the output. Locations p_7 and p_8 (drawn with dotted lines) ensures the priorities in servicing of the materials flow arrivals (the arrival of the external raw materials).

For the average delay of demands ($g = \frac{4}{n} \sum_{i=1}^n M_i(p) \cdot \tau_i$) the perturbation estimator given in (2) is unbiased.

$$\frac{4}{n} \sum_{i=1}^n \frac{\partial g}{\partial \tau_i} \cdot \frac{d\tau_i}{d\theta} = \frac{4}{n} \sum_{i=1}^n M_{i-1}(p) \cdot \frac{d\tau_i}{d\theta} \quad (7)$$

$$g = \frac{4}{n} \sum_{i=1}^n L(M_{i-1}) \cdot \tau_i \quad (8)$$

Where $L(M_i) = M_i(p_1) + M_i(p_2) + M_i(p_3) + M_i(p_4) + M_i(p_5) + M_i(p_6) + M_i(p_8)$.

The perturbation estimator is equal to:

$$\frac{4}{n} \sum_{i=1}^n L(M_{i-1}) \cdot \frac{d\tau_i}{d\theta} \quad (9)$$

Assuming that firing times are exponentially distributed with mean equal to: Θ for t_1 and t_2 ; 1 for t_2 ; 0,86 for t_3 ; 0,75 for t_8 , we consider the average customer delay (Θ). The mean value of the gradient evaluated at $\Theta = 1.22$ and at $\Theta = 1.24$ is close to the central finite difference: $(E[g(1.24)] - E[g(1.20)]) / 0,04 = -10.27$. This result is acceptable, and we notice that additional values can be obtained by modifying the net structure as discussed before, and as it is drawn with dotted lines in Fig.2, by modifying the marking in the places p_7 and p_8 .

5. A FLEXIBLE CONSTRUCTION SYSTEM

5.1 The system description

The construction system considered in this paper consists of two cells linked together by a material system composed of two buffers A and B and a conveyor. Each cell consists of a machine to handle within cell part movement. Pieces enter the system at the load/unload station, where they are released from those two buffers, A and B, and then are sorted in cells (pieces of type “a” in one cell, and pieces of type “b” in the other cell). We notice that in the buffer A are pieces of types “a”, “b”, and others, where the number of pieces “a” is greater than the number of pieces “b”. In the buffer B there are pieces of types “a”, “b”, and others, where the number of pieces “b” is greater than the number of pieces “a”. The conveyor moves pieces between the load/unload station the various cells. The sorted piece leaves the system,

and an unsorted piece enters in the system, respectively in one of those two buffers A or B. The conveyor along with the central storage incorporates a sufficiently large buffer space, so that it can be thought of as possessing infinite storage capacity. Thus, if a piece routed to a particular cell finds that the cell is full, it is refused entry and is routed back to the centralized storage area. If a piece routed by conveyor is of a different type of the required types to be sorted, respectively “a”, and “b”, then that piece is rejected out of the system. We notice that once a piece is blocked from entry in a cell, the conveyor does not stop service; instead it proceed with its operation on the other pieces waiting for transport. At the system level, we assume that the cells are functionally equivalent, so that each cell can provide the necessary processing for a piece. Hence, one cell is sufficient to maintain production (at a reduced throughput). We say that the manufacturing system is available (or, operational) if the conveyor and at least one of the cells are available. A cell is available if its machine is available. Over a specified period of operation, owing to the randomly occurring subsystem failures and subsequent repairs, the cellular construction system (CCS) will function in different configurations and exhibit varying levels of performance over the random residence times in these configurations. The logical model of our manufacturing system is showed in Fig.3.

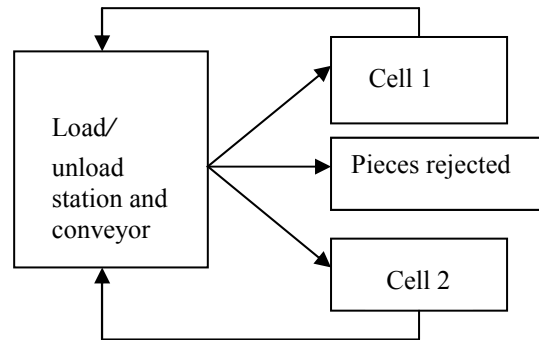


Fig. 3. Logical model for a manufacturing system

5.2 A Markov model for evaluating the availability of the system

For the flexible manufacturing system depicted in Fig.1, we assume that the machines are failure-prone, while the load/unload station and the conveyor are extremely reliable. Assuming the failure times and the repair times to be exponentially distributed, we can formulate the state process as a continuous time Markov chain (CTMC). The state process is given by $\{X(u), u \geq 0\}$ with state space $S = \{(ij), i \in \{0,1,2\}, j \in \{0,1\}\}$, where i denotes the number of machine working, and j denotes the status of the material handling system (load station and conveyor): up (1), and down (0). We consider the state independent (or, time dependent) failure case and the operation dependent failure case separately.

Time dependent failures: In this case, the component fails irrespective of whether the system is operational or not. All failure states are recoverable. Let r_a and r_m denote the repair rates of the material handling system, and a machine respectively. The state process is shown in Fig.4.a.

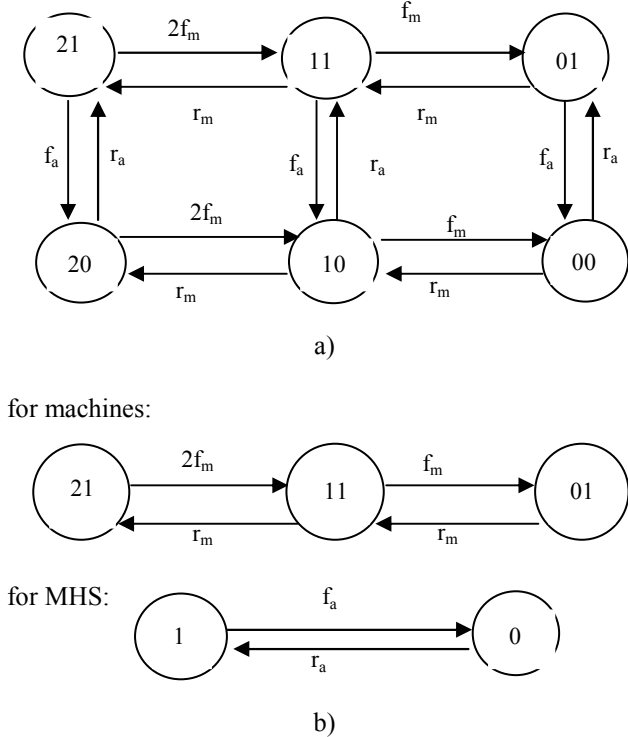


Fig.4. State process of a CCS with time-dependent failures, (a) State process for a state-independent failure model, (b) Decomposed failure/repair process

Because the failure/repair behavior of the system components are independent, the state process can be decomposed into two CTMCs as shown in Fig.2.b. Analytically, the state process is expressed by relations: $S_0 = \{(21), (11)\}$ and $S_F = \{(20), (10), (00)\}$. For each state in S_F no production is possible since the Material Handling System (MHS) or both the machines are down. In Fig.2.b the failure/repair behaviour of each resource type (machines or MHS) is described by a unique Markov chain. Thus, the transient state probabilities, $p_{ij}(t)$ can be obtained from relation:

$$p_{ij}(t) = p_i(t)p_j(t) \tag{10}$$

where $p_i(t)$ is the probability that i machines are working at time t for $i = 0,1,2$. The probability $p_i(t)$ is obtained by solving (separately) the failure/repair model of the machines. $P_j(t)$ is the probability that j MHS (load/unload station and conveyor) are working at instant t , for $j = 0,1$. Let f_a and f_m denote the failure rates of the MHS and of a machine respectively.

Operation-dependent failures: Assume that when the system is functional, the resources are all fully utilized. Since failures occur only when the system is operational, the state space is: $S = \{(21), (11), (20), (10), (01)\}$, with $S_0 = \{(21), (11)\}$, $S_F = \{(20), (10), (01)\}$. The Markov chain model is shown in Fig.5. Transitions representing failure will be allowed only when the resource is busy. Transitions rates can however be computed as the product of the failure rates and percentage utilization of the resource, and T_k^{ij} represents the average utilization of the k^{th} resource in the state (ij) .

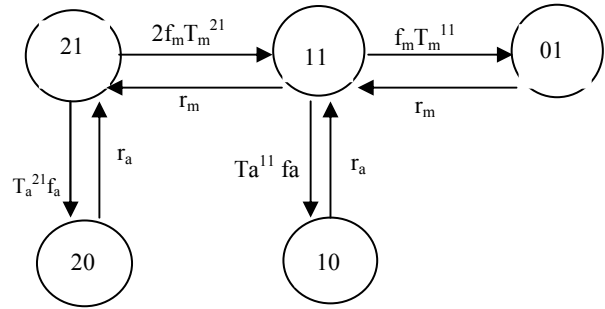


Fig.5. State process of a CCS with state-dependent failures

5.3 Numerical study

For the CCS presented in this paper, in the table 1 are given the failure/repair data of the system components We notice that T_k^{ij} (the average utilization of the system of the k^{th} resource in state (ij)), $T_k^{ij} = 1$ since the utilization in each operational state is 100% for all $i, j, k, i = \{0,1,2\}, j = \{0,1\}$, $k = 4$. The other notations used in table 1 are: f is the exponential failure rate of resources, r is the exponential repair rate of resources, N_p is the required minimum number of operational machines in cell p , $p = \{1,2\}$, and n_p is the total number of machines in cell p .

Table 1. Data for the numerical study

	R	F	N_p	n_p	T_k^{ij}
Machines	1	0,05	1	2	1
MHS	0,2	0,001	1	1	1

From Fig.2 and Fig.3 we calculate the corresponding infinitesimal generators and after that, the probability vector of CTMC. With relation (10) we calculate the availability of CCS. The computational results are summarized in Table 2 for the state process given in Fig.4 (CCS with time-dependent failures), and respectively in Table 3 for the state process given in Fig.5 (CCS with state-dependent failures). We consider the system operation over an interval of 24 hours (three consecutive shifts).

Table 2. Computational results for the CCS in Fig.4

Time hour	Machines	MHS	System Availability
0	1.0000	1.0000	1.0000
1	0.9800	0.9548	0.9217
4	0.9470	0.8645	0.7789
8	0.9335	0.8061	0.7025
12	0.9330	0.7810	0.6758
16	0.9331	0.7701	0.6655
20	0.9330	0.7654	0.6623
24	0.9328	0.7648	0.6617

Table 3. Computational results for the CCS in Fig.5

Time hour	Machines	MHS	System Availability
0	1.0000	1.0000	1.0000
1	0.9780	0.9528	0.9201
4	0.9450	0.8628	0.7762
8	0.9315	0.8039	0.7008
12	0.9310	0.7798	0.6739
16	0.9320	0.7688	0.6632
20	0.9318	0.7639	0.6598
24	0.9320	0.7636	0.6583

The results of the availability analysis of the construction system are illustrated in Fig.4, which depicts the availability of the system as a function of the time. The numbers $x = 1, 2$ indicate the system in Fig.4, respectively Fig.5. One can see from Fig.6 that the layout with CCS with time-dependent failures is superior to that with CCS with state-dependent failure

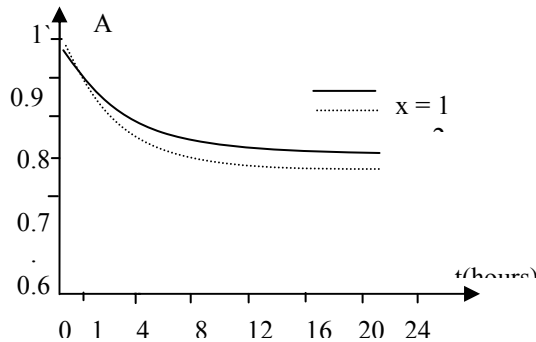


Fig.6 Availability analysis of the CCS

6. CONCLUSIONS

In this paper we analysed the perturbation estimators in digital systems modelled with stochastic Petri nets (SPN's). The approach presented in this paper (e.g., Stochastic Artificial Social Systems) can be used to analyze SPN's that model complex dynamic system interactions. Unbiased gradient estimators proposed in [4], [6] were used for the sensitivity analysis of the GSMP representation and some practical solutions for attenuating the perturbations influences were indicated. The proposed procedure was imagined for a

data network perturbation analysis. We estimate that this methodology can be applied to modelling and analysis of manufacturing systems, job scheduling in a chain management system, such as construction systems. Future research will focus to differential and fluid Petri nets in order to estimate throughput of complex systems.

An analytical technique for the availability evaluation of the construction systems was presented. The novelty of the approach is that the construction of large Markov chains is not required. Using a structural decomposition, the construction system is divided into cells. For each cell a Markov model was derived and the probability was determined of at least N_i working machines in cell i , for $i = 1, 2, \dots, n$ and j working material handling system (MHS) at time t , where N_i and j satisfy the system production capacity requirements. The model presented in this paper can be extended to include other components, e.g., tools, control systems. The results reported here can form the basis of several enhancements, such as conducting performance studies of with multiple part types.

REFERENCES

- [1] T. Murata, *Petri nets: Properties, analysis and applications*, Proc. IEEE, vol. 77, pp. 541-580, 1989.
- [2] F. Baccelli, Z. Liu, *Comparison properties of stochastic decision free Petri nets*, IEEE Trans. on Autom. Contr., vol. 37, No. 12, pp. 1905-1920, 1992
- [3] P. J. Haas, G. S. Shendler, *Stochastic Petri nets: Modeling power and limit theorems*, Probability Eng. Inform. Sci., vol. 5, pp. 477-498, 1991
- [4] M. C. Fu and J. Q. Hu, *Extensions and generalizations of smoothed perturbation analysis in a generalized semi-Markov process framework*, IEEE Trans. Automat. Contr., vol. 37, pp. 1483-1500, 1992
- [5] F. Archetti, A. Gaivoronski, A. Sciomachen, *Sensitivity analysis and optimisation of stochastic Petri nets*, J. Discrete Event Dynamic Systems: Theory Appl., vol. 3, pp.5-37, 1993
- [6] X. Xie, *Perturbation analysis of stochastic Petri nets*, IEEE Trans. on Autom. Control, vol. 43, No. 1, pp. 76-80, 1998
- [7] Sh. T. Yee, J. A. Ventura, *Phase-type approximation of Petri nets for analysis of manufacturing systems*, IEEE Trans. on Rob. and Autom., No. 3, pp. 318-322, 2000
- [8] C. Ciufudean, D. Popescu, "Modeling Digital Signal Perturbation with Stochastic Petri Nets", *Advances in Electrical and Computer Engineering*, vol. 4(II), no. 1 (21), pp. 71-75, Suceava, Romania, 2004.
- [9] C. Ciufudean, A.B. Larionescu, "Safety criteria for production lines modeled with Petri nets", *Advances in Electrical and Computer Engineering*, vol. 2(9), no.2(18), pp.15-20, 2002.
- [10] Ciufudean C., "Modeling the reliability of the interaction man-machine in railway transport", *The Annals of the "Stefan cel Mare" University of Suceava*, No.13, pp.80-84, 2000.

