

# State model degree reduction of a multistate complex system structured by a Coloured Petri Nets (CPN)

Wang Zhouhang, Atli Maen, Adjallah Kondo

► **To cite this version:**

Wang Zhouhang, Atli Maen, Adjallah Kondo. State model degree reduction of a multistate complex system structured by a Coloured Petri Nets (CPN). IFAC Proceedings Volumes, Elsevier, 2012, 45 (31), pp.103-108. 10.3182/20121122-2-ES-4026.00036 . hal-03089298

**HAL Id: hal-03089298**

**<https://hal.univ-lorraine.fr/hal-03089298>**

Submitted on 28 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# State model degree reduction of a multistate complex system structured by a Coloured Petri Nets (CPN)

WANG Zhouhang<sup>1</sup>, ATLI Maen<sup>1</sup>, ADJALLAH Kondo H<sup>2</sup>

<sup>1</sup>Université de Lorraine, 34 cours Léopold, CS 25233, 54052 Nancy cedex, France

<sup>2</sup>Ecole Nationale d'Ingénieurs de Metz, 1 route d'Ars Laquenexy, CS65820, 57078 Metz Cedex 3, France

This paper introduces a method of modelling multistate repairable multi-unit systems' behaviours subjected to stochastic degradation processes, using the coloured Petri nets (CPN). This solution prevents the combinatorial explosion problem that reduces the use of Markov graphs for studying such systems. The method combines the ability to model the state and evaluate the operational performances of a complex system. The technique also simplifies the study of  $k$ -out- $n$  structure systems, while allowing extraction of the minimal cuts. The method provided is grounded on theorems used to design an algorithm of systematic construction of models.

Keywords: Coloured Petri Net, multistate unit, multi-unit system, availability, performance, model.

## 1. Introduction

For decision support to inspecting or replacing a component in complex systems, the dependability models of single-unit systems subjected to stochastic deteriorations have been widely investigated for systems' availability. For example, authors in [8] considered a multi-level control limit model, while to derive a decision model, authors in [9] have exploited theories of regenerative processes and semi-regenerative processes. Other strategies, for example in [17] and [10], use monitoring of physical behaviour signals of units or health condition features, or combine both of them, to evaluate the level of degradation or performance for real-time decision support. This approach can quickly become very complicated when dealing with repairable multi-unit systems because of combinations of interactions among states [6]. Without strong restrictive hypotheses, state models' analytical development is cumbersome and even impossible in some instances. Instead, for dependability studies, researchers prefer to rely on digital simulation. The latter offers the advantage of allowing assessing the cost of maintenance actions, whatever is the maintenance policy while considering factors such as the quality of the technical activities, the system degradation being continuously monitored ([14][11][18][7][13]).

For studying the evolution of the state of a multi-unit system, authors in ([2][15][16]) proposed state models (e.g. Markov graphs, Bayesian belief networks, Stochastic Petri nets) at the system level, and embedding analytical models based on probability theories enabling estimating the level of degradation or reliability of each unit [12]. Indeed, Markov processes are appropriated to describe the multi-unit degradation states ([3][11]). In this paper, the coloured Petri net combining the abilities of modelling and evaluating a multistate multi-unit system is presented. In the remainder of this paper, Section 2 recalls the background of a Petri net model. The multistate multi-unit system's modelling problem based on different structures is formulated in Section 3 (single unit, parallel units, series units and composite structure units). We introduce new concepts as *impact factor* that may refer to Birnbaum [4], *stop function* and *stop constraints*. The minimal cut sets and  $k$ -out-of- $n$  structure system are then discussed respectively in Sections 4 and 5, before summarizing our contribution in Section 6.

## 2. Petri Net modelling

Petri nets are graphical and mathematic modelling tool applicable to many systems ([15][16][19][5]). They are efficient for describing concurrent, distributed, parallel, asynchronous and stochastic processes ([16] [17]). So, they are well adapted for modelling and studying multistate multi-unit systems state and performances. In our study, the system's performance relates to a system's availability and the degradation of its component rate or reliability level. Under continuous deterioration or degradation, we considered that the state of a component change after it's rate of deterioration or degradation level passes specified thresholds  $L_1$  and  $L_2$  ([8][9][3]). The level  $L_1$  defines the threshold for preventive maintenance, and  $L_2$  defines the threshold for corrective maintenance action.

Let consider the Coloured Petri Nets (CPNs), which allow us to introduce the token types. Namely, tokens are differentiated using colours. This section discusses four case studies of Petri nets modelling of the system, such as single-unit, parallel-units, series-units, and mixed parallel-series-units (complex) systems.

## 2.1. Modelling of a single unit

Let consider a system consisted of single-unit  $E$  shown in figure 1-a). This unit embodies three states, according to the diagram shown in figure 1-b). The first state represents the unit in good health; the second corresponds to a degraded unit, and the last stands for a failing unit. The figure 1-b) models the three possible states for one single unit where: unit's condition is good at beginning; then the unit is degraded, which defines the second state; finally, the unit fails as mentioned by the last state.

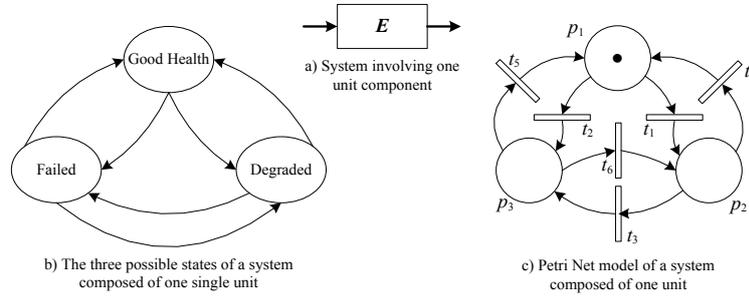


Figure 1: Representing the three possible cases for one single unit.

The Petri net shown in figure 1-c) defines a model for a single unit. The place  $p_1$  (respectively  $p_2$  and  $p_3$ ) represents the first state (good health) (respectively the degraded state and the failing state). A token represents the current state of the unit. For example, in figure 1-c), unit  $E$  is in good health as place  $p_1$  contains one token. Similarly, a token in  $p_2$  (respectively  $p_3$ ) depicts a degraded unit  $E$  (respectively failing unit). The crossing above level  $L_1$  (respectively level  $L_2$ ) is represented by transition  $t_1$  (respectively  $t_3$ ). Although the degradation level is lower than level  $L_1$ , a failure sometimes occurs randomly: this kind of failure is modelled by transition  $t_2$ . Also, unit  $E$  may fail while its degradation level lies between  $L_1$  and  $L_2$ , modelled by transition  $t_3$ . Finally, transition  $t_4$  triggers a preventive maintenance action, while transition  $t_5$  triggers corrective maintenance or replacement action.

## 2.2. System of parallel units

Let suppose a system consisting of two parallel units, as shown in figure 2-a). The system will not fail whether one unit fails. However, there will be some impacts on system performance. The system will stop working if and only if all units fail (all tokens pass in  $p_3$ ). Figure 2-b) depicts the states of such a system, where the number of states  $N$  computes as follows:

$$N = 3^n \quad (1)$$

where  $n$  is the number of units. In the sequel, we present definitions, theorems, and rules for the complete Petri Net model's systematic construction algorithm.

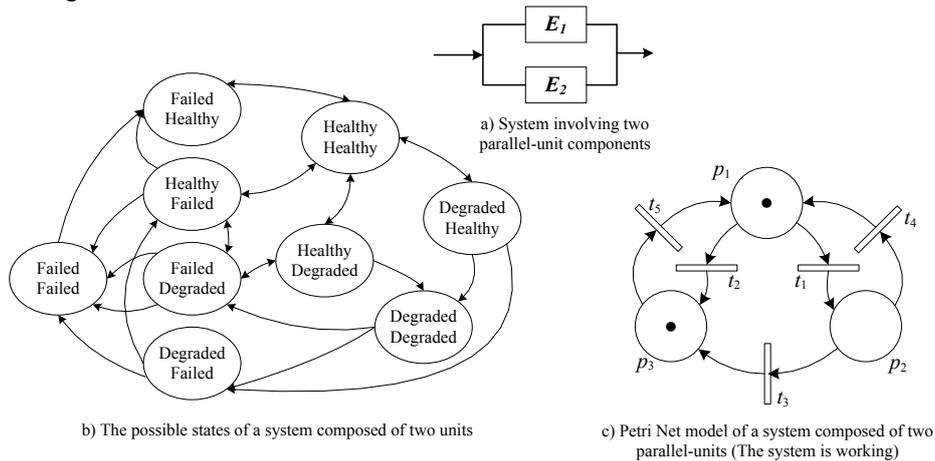


Figure 2: System of two parallel units.

**Definition 1:** The *impact factor* of a unit is a real positive number, which takes a value over  $]0, 1]$  relating to its state's influence on the system's operation. An *impact factor* value 1 implies the unit failure causes complete stop of the system, and a value 0 means the unit failure does not influence the system operation. Trivial, the whole system takes the max *impact factor*  $S$  equal to 1, and no unit in the system can take an impact factor equal to zero (a unit taking a value zero does not belong to this system).

**Theorem 1:** Let consider a system or subsystem which consists of  $n$  branches,  $S \equiv S_1 // S_2 // \dots // S_n$ , and let the *impact factor* of this system (respectively for the branches) denote by  $Im S$  (respectively  $Im S_1, Im S_2, \dots, Im S_n$ ). To simplify, we denote  $S$ 's impact factor (respectively  $S_1, S_2, \dots, S_n$ ). Each branch takes an impact factor equals to  $S_1 = S_2 = \dots = S_n = S/n$ . We suppose that all the units have the same *impact factor* so that we may rewrite:

$$S_i = S / n \quad (3)$$

For example, in figure 2:  $E_1=E_2=1/2$ .

**Definition 2:** The weight of a unit, denoted  $w_i$ , is a binary number, which takes either value 1, in case of failure, or the value 0 otherwise. In other words, each token in  $p_3$  yields a weight equals to one for the related unit.

$$w_i = \begin{cases} 1, & \text{if token in } p_3 \\ 0, & \text{otherwise} \end{cases}$$

**Definition 3:** The *Stop Function*  $F$  is a function that indicates the whole system's current performance by computing the *failed units' impact factor*.

**Rule 1:** The Stop Function of a system or subsystem consists of  $n$  branches computed as follows:

$$F = \sum_{i=1}^n w_i \cdot S_i \quad (4)$$

### 2.3. System of units in series

Let suppose a system consisting of two series-units, as shown in figure 3-a). When a unit fails, the system completely fails. Therefore, the number of possible states is less than  $N=3^n$  (compare Figures 2-b and 3-b). In other words, if a token passes in  $p_3$  (a unit fails), the other token can pass neither through  $t_1$  nor  $t_2$ . This feature cannot be modelled without introducing an arc called an inhibitor arc [16]. An inhibitor arc connects a place to a transition. It is represented by a dashed line terminating with a small circle instead of an arrowhead at the transition (Figure 3-c). The inhibitor arc disables the transition when the input place has a token and enables it when the input place has no token. The objective of this arc is to specify whether the system has failed completely or not.

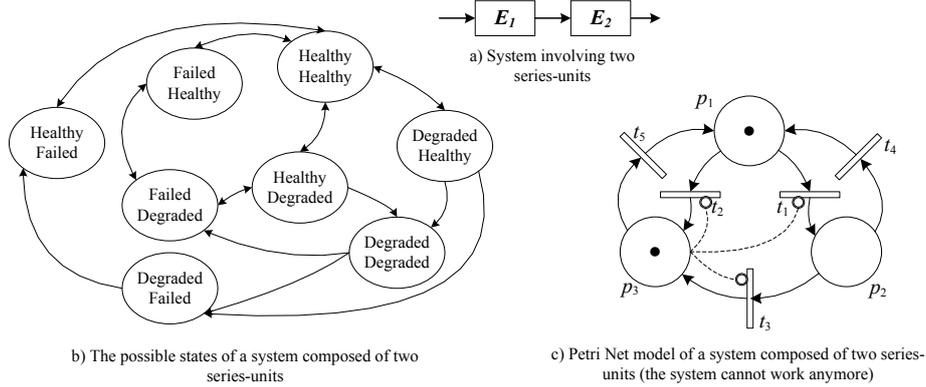


Figure 3: System of two units in series.

**Theorem 2:** Let consider a system or subsystem that consists of  $m$  series-units,  $S \equiv S_1 \cdot S_2 \cdot \dots \cdot S_m$ , and let the *impact factor* of this system (respectively for the units) denote by  $Im S$  (respectively  $Im S_1, Im S_2, \dots, Im S_n$ ). To simplify, we denote the *impact factor* by  $S$  (respectively  $S_1, S_2, \dots, S_m$ ). Each unit takes an impact factor equals to  $S_1 = S_2 = \dots = S_m = S$ .

**Rule 2:** The stop function of a system or subsystem consisting of  $m$  series-units is computed as follows:

$$F = \text{Max}(w_1 \cdot S_1, w_2 \cdot S_2, \dots, w_m \cdot S_m) \quad (6)$$

### 2.4. Model of a multistate multi-unit system

We consider a multi-unit system arranged in series-parallel-series and parallel-series-parallel structures, as shown in figure 4. This complex multistate multi-unit system may be broken down into subsystems; similarly, a subsystem may also be broken down into sub-sub-systems, etc., as depicted in Figure 4. We have broken down the system up to a single unit level where a uniform coloured token represents each unit. The coloured token helps distinguish the token that has passed the transition and subsequently identifies the degraded or failed unit.

As the system has a hybrid architecture (series-parallel-series), the use of inhibitor arcs is inappropriate in the Petri Net model of series-units. Therefore, we consider that transitions  $t_1, t_2$  and  $t_3$  are controlled transitions [1]. We added stop constraints to these transitions defined in Theorem 3.

**Theorem 3:** Let consider *Stop Constraint* (SC) attached to the controlled transitions to detect the transit ability (firing condition). When the stop function of a system or subsystem is equal to or greater than its impact factor (SC for  $S_i$ : if  $F_i \geq S_i$ ), this system or subsystem will stop.

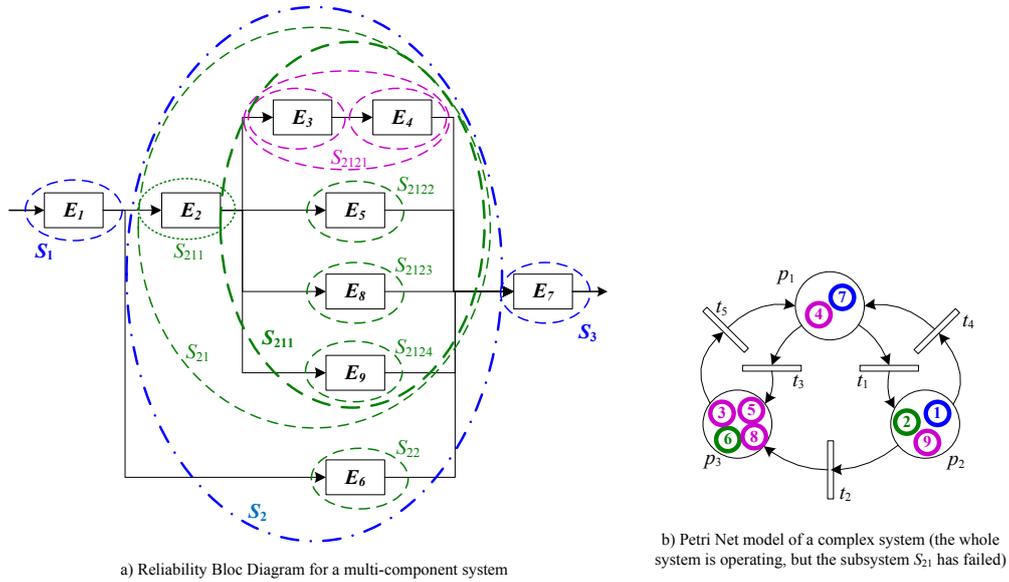


Figure 4: Complex System.

Let apply Theorem 3 to compute the stop constraints SCs for the system depicted in Figure 2:

- a) SC for  $S$ : if  $F \geq 1$ .
- b) SC for  $S_{21}$ : if  $F_{21} \geq 1/2$ .
- c) SC for  $S_{2121}$ : if  $F_{2121} \geq 1/8$ .

Note: there is no need to compute the SC for the branch that has a single unit.

We may rewrite the Theorem 3 to simplify by using many rules:

**Rule 3:** Each token passes to  $P_3$ , update the according to stop functions to this token.

**Rule 4:** If one of the stop constraints SCs is validated, the controlled transitions will disable against the passage of tokens subject to this SC.

Thus, an appropriate algorithm has been designed to generate a complete Petri net model of the system.

### 3. Determining the minimal cut sets

Cut sets are the unique combinations of component failures that can cause a system failure. Specifically, a cut set is a minimal cut set if the remaining events collectively are no longer a cut set when any primary event is removed from the set. A minimal cut sets have traditionally been used to estimate reliability for complex reliability block diagrams (RBD) or fault trees that cannot be simplified by a combination of the simple constructions (parallel, series,  $k$ -out-of- $n$ ).

Consider the previous RBD in figure 4. Let the equivalent *Fault Tree* (all minimal cut sets) shown in figure 5.

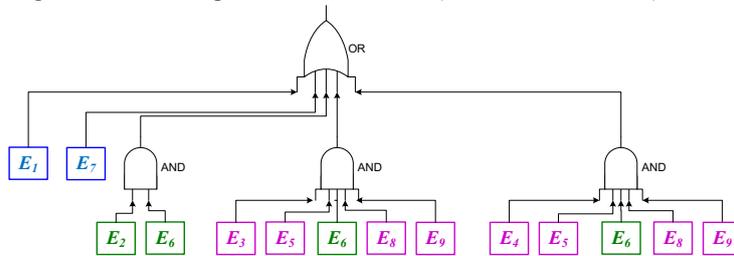


Figure 5: Intermediate fault tree equivalent to the diagram shown in Figure 4

The construction of the Fault Tree may be very tedious. A straightforward approach is available using minimal cut sets. First, the RBD is constructed, and then the minimal cut sets are obtained. Note that the analysis can be performed based on the RBD once built.

*Corollary* from Theorem 3 and the concept of a unit *impact factor*: one concludes that each set of single units forms a minimal cut set if it has the two following features:

- A. The *sum of impact factors* equals to one exactly.
- B. The *stop function* of this set equals to one exactly.

Each unit with an impact factor equal to one may be trivially called first-order minimal cut set (as  $E_1$  and  $E_7$ , figure 4). The table below shows the system's minimal cut sets depicted in Figure 4 with the fault tree in Figure 5.

Set	Minimal Cut set	Reasoning
$E_1$	<b>Yes</b>	$E_1=1$ $F_1=1$
$E_1, E_2$	No	$E_1+E_2=3/2>1$ , although $F_{(E_1, E_2)}=1$
$E_2, E_6$	<b>Yes</b>	$E_2+E_6=1$ $F_2=1$
$E_3, E_4, E_5, E_8, E_6$	No	$E_3+E_4+E_5+E_8+E_6=1$ $F_{(E_3, E_4, E_5, E_8, E_6)}=7/8<1$

#### 4. Determining the constraint of a $k$ -out-of- $n$ structure system

An  $n$ -units system that *works* (or is “healthy”) if and only if at least  $k$  of the  $n$ -units work (or are healthy) is called a  $k$ -out-of- $n$ : $G$  system. An  $n$ -units system that *fails* if and only if at least  $k$  of the  $n$  components fail is called a  $k$ -out-of- $n$ : $F$  system. Based on these two definitions, a  $k$ -out-of- $n$ : $G$  system is equivalent to an  $(n-k+1)$ -out-of- $n$ : $F$  system. The term  $k$ -out-of- $n$  system is often used to indicate either a  $G$  system or an  $F$  system or both. Both parallel and series systems are special cases of the  $k$ -out-of- $n$  system. A series system is equivalent to a 1-out-of- $n$ : $F$  system and to an  $n$ -out-of- $n$ : $G$  system while a parallel system is equivalent to an  $n$ -out-of- $n$ : $F$  system and to a 1-out-of- $n$ : $G$  system. Since the stop constraint represents the complete stop (all the units fails, i.e.  $n$ -out-of- $n$ : $F$  system), one can rewrite the stop constraint of a system  $S_i$  as follows:

$$F_i \geq n/nS_i \quad (7)$$

where  $S_i$  is the system impact factor. Whether the system is  $k$ -out-of- $n$ : $F$  system, the stop constraint will be changed to take the following form:  $F_i \geq k/nS_i$ .

However, if we have a  $k$ -out-of- $n$ : $G$  system, the stop function will be

$$F_i \geq (n-k+1)/nS_i \quad (8)$$

Let take the subsystem  $S_{212}$  shown in Figure 4-a), where at least 2 branches from the 4 branches must work, i.e. a 2-out-of-4: $G$  system. The stop function of this subsystem takes the form (8).

SC for  $S_{212}$ :

$$F_{212} \geq (4-2+1)/4 \times 2 \Rightarrow F_{212} \geq 3/8$$

One must consider this influence on the whole system since all the system and the subsystems related to  $S_{212}$  will be merged in one way or another. The stop constraints for  $S_{21}$ ,  $S_2$  and  $S$  will be SC for  $S_{21} \geq \text{Min}(1/2, 3/8) = 3/8$ , SC for  $S_2 \geq (1/2 + 3/8) = 7/8$  and SC for  $S \geq (1/2 + 3/8) = 7/8$ .

#### 5. Conclusion

In this paper, we introduced a new method. We suggested an algorithm to build a graphical and mathematical model for studying evolutions of the state and performances of multi-unit multistate systems. New concepts such as *impact factor*, the *unit's weight*, *stop function* and *stop constraint*; are used as basics in this algorithm, which allow us to simplify and study complex multi-unit systems. The proposed method comprises two main parts: the first one is graphical, which is a Petri net model, while the second part is a mathematical description based on equations and functions. The function is updated upon the state of the system. The constraint function is updated upon the state of the system. At each step, the stop constraints are evaluated to detect a possible change in the system's state according to the graphical Petri Nets model operation. The method provided is grounded on theorems used to design the algorithms of the models' construction.

#### Reference

- [1] Atli, M., Achour, Z., Sava, A., & Rezg, N. (2010-b). Supervisory Control of Partially Observable Marked Graph Based on Marking Exclusion Constraint. Proc. 5th IFAC Conf. on Management and Control of Production Logistics, Coimbra, Portugal, Portugal, Sep. 08-10, 64–69.

- [2] Ben-Arieh D, Kumar R.R., Tiwarib MK. (2004) "Analysis of assembly operations' difficulty using enhanced expert high-level coloured fuzzy Petri net model", *Robotics and Computer-Integrated Manufacturing*, 20, 385–403.
- [3] Ben-Arieh D, Ranjan K R & Tiwari M K. (2004) "Analysis of assembly operations' difficulty using enhanced expert high-level coloured fuzzy Petri net model", *Robotics and Computer-Integrated Manufacturing*, 20, 385–403.
- [4] Birnbaum Z.W. (1969), "On the importance of different components in a multi-component system", Washington Univ Seattle Lab of Statistical Research, Technical Report no54, Contract No-477(38), May 20, 1968.
- [5] CecilJA, Srihari A & Emerson CR (1992) "A review of Petri net based applications in manufacturing". *International Journal of Advanced Manufacturing Technology*, 7, 168–77.
- [6] Chen D & Trivedib K. S. (2005) "Optimization for condition-based maintenance with semi-Markov decision process", *Reliability Engineering and System Safety*, 90, 25–29.
- [7] Dekker R. (1996) "Applications of maintenance optimization models: a review and analysis", *Reliability Engineering & System Safety*, 51(3), 229–240.
- [8] Grall A, Berenguer B, Dieulle L. (2002) "A condition-based maintenance policy for stochastically deteriorating systems", *Reliability Engineering and System Safety*, 76, 167–180.
- [9] Grall A, Dieulle L, Bérenguer C, & Roussignol M. (2002), "Continuous-time predictive-maintenance scheduling for a deteriorating system", *IEEE Trans. on reliability*, 51(2), 141–150.
- [10] Jardine A. K.S., Lin D., Banjevic D. (2006), "A review on machinery diagnostics and prognostics implementing condition-based maintenance", *Mechanical Systems and Signal Processing*, 20(7), 1483-1510.
- [11] Jiang ZB., Zuo MJ., Fung RYK., Tu PYL. (2001) "Coloured Petri Nets with changeable structures (CPN-CS) and their applications in the modelling of one-of-a-kind production (OKP) systems", *Computer & Industrial Engineering*, 41, 279–308.
- [12] Kececioglu, D. (1991) "Reliability Engineering Handbook", 2, Prentice Hall, Inc., New Jersey.
- [13] Lam C.T & Yeh R.H. (1994) "Optimal maintenance-policies for deteriorating systems under various maintenance strategies", *IEEE Trans. on reliability*, 43(3), 423–430.
- [14] Levitin G, Lisnianski A, (2000) Optimization of imperfect preventive maintenance for multistate systems, *Reliability Engineering and System Safety*, vol. 67, 193–203.
- Marseguerra M, Zio E, Podofillini L. (2002) "Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation", *Reliability Engineering and System Safety*, 77, 151–166.
- [15] Murata T, (1989) "Petri Nets: properties, analysis and applications". *Proc. of the IEEE*, 77(4), 541-580.
- [16] Wang ZH, Adjallah KH. (2010) "Data and information collection process modelling for product lifecycle and health management", *International Conference on Product Lifecycle Management, PLM'10*.
- [17] Wang, H. (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139, 469–489.
- [18] Zha XF, Lim SYE, Fok SC (1998). Integrated knowledge-based Petri nets intelligent flexible assembly planning. *Journal of Intelligent Manufacturing*, 9(3): 235–53.
- [19] Zhang WX. (1989) "Representation of assembly and automatic robot planning by petri net". *IEEE Trans. on Systems Man and Cybernetics*, 29(2): 418–22.