Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Control synthesis approach for DES modelled by Petri Nets

R. Bekrar*, N. Messai, N. Essounbouli, A. Hamzaoui*, B. Riera****

** CReSTIC, IUT de Troyes, 9 rue de Québec, BP 396, 10026 Troyes Cedex, France*
(e-mail: {rebiha.bekrar, najib.essounbouli, abdelaziz.hamzaoui}@ univ-reims.fr)
*** CReSTIC, Université de Reims Champagne Ardenne, UFR Sciences Exactes et Naturelles, Moulin. de la House, BP 1039, 51687, Reims Cedex 2, France*
(e-mail: {nadhir.messai, bernad.riera}@univ-reims.fr).

Abstract: This paper presents a control synthesis approach for discrete event systems modelled by ordinary Petri Nets (PN) to solve a forbidden state problem. The PN herein considered are transitions controllable and contain measurable and non measurable transitions. The proposed PN controller is synthesised using the influence paths of the forbidden transitions. The latter are deduced from the PN reachability graph. The designed PN controller is maximally permissive because it prevents the occurrence of forbidden markings and guarantees exactly the desired behaviours. The proposed approach is illustrated by an example.

1. INTRODUCTION

Petri Nets (PN) is an appropriate tool for studying discrete events dynamical systems (DES). Thanks to their graphical representation and algebraic formulation, PN have been extensively used in modelling, identification, simulation, performances evaluation and control of DES.

This paper is considered as a complementary phase of our work (Bekrar *et al*, 2006a, 2006b) on the identification of DES using PN. The proposed identification method has the advantage to provide a unique PN that can generate all the observed behaviours of the system. Nevertheless, the identified PN can, sometimes, generate some states, which are not observed in the set of the input/output signals used for the identification task. Thus, we suggested completing the obtained PN model by adding control places in order to avoid the undesirable states. This leads to design a PN controller that guarantees the desired behaviours.

A look at the literature shows that this control problem is considered either as a Forbidden State Problem (FSP) or as a Forbidden State Transitions Problem (FSTP). Several techniques have, also, been proposed to treat both problems. Hence, formal treatment for solving the FSP and the FSTP using the theory of regions are proposed in (Ghaffari *et al*, 2002), where necessary and sufficient conditions for solution existence are established. The proposed approach is valid for bounded PN with controllable and uncontrollable transitions. In addition, a method to solve the FSP for live bounded marked graphs with uncontrollable transitions using General Mutual Exclusion Constraints (GMEC) is introduced in (Ghaffari *et al*, 2003). The case of unobservable transitions is considered in (Achour *et al* 2004) and the existence of a maximally permissive PN controller for the FSP of bounded PN under partial observation is treated in (Achour *et al*, 2005). Based on the supervisory theory of (Ramadge and Wonham, 1987), Lee *et al* (2006) have proposed an algorithm for elaborating constraint synchronous reachability graph and calculating forbidden and authorised sequences. Then, the theory of regions is adapted to calculate the control places to be added to the initial model.

Based on the unfolding technique, Giua and Xie (2004) proposed a method to treat the control problem of safe PN by enforcing the marking constraints. The developed maximally permissive supervisor takes the form of control places to be added to the unfolding. The solution of deadlock problem is presented in (Giua and Xie, 2005). Finally, an optimisation approach to deal with PN control problem for enforcing GMEC using Integer Linear Programming techniques is introduced in (Basile *et al*, 2007a). The classical partition of the events set into controllable and uncontrollable events is replaced by associating a control and an observation cost to each event. Afterwards, the FSP specified by GMEC in backward conflict-free and free choice uncontrollable subnets (BCFCNs) is treated in (Basile *et al*, 2007b).

The analysis of these works shows that they can not be exploited directly in our case because, it is impossible to characterise the forbidden states by constraints. Indeed, the control specification is mainly important to synthesise a controller. However, we have neither the control specification nor the system description because the identified PN model that we want to control is obtained using only the measurable inputs and outputs system signals. Let's note that, we can determine the forbidden states uniquely by comparing the observed system states and those reachable by the identified PN model.

This paper deals with the supervisory control problem of DES modelled by ordinary PN. The PN herein considered are transitions controllable and contain measurable and non measurable places. The latter represent the non measurable outputs system signals. Indeed, the fact that all PN transitions are controllable than, the PN reachability graph does not contains dangerous markings. In addition, we suppose that the initial marking is the unique marked state. Finally, we consider that the PN reachability graph does not contain deadlock states because the input and the output signals observed during the identification phase represent only the normal behaviours of the considered system.

Moreover, the proposed PN controller is synthesised by determining a set of forbidden markings and using the influence paths, of the forbidden transitions, which are deduced from the PN reachability graph. The designed PN controller permits to avoid the occurrence of forbidden markings, generated by the non controlled PN model, and guarantees a set of desired behaviours.

This paper is organised as follows. Section 2 reviews some definitions related to PN. Section 3 presents the considered FSP and introduces the control specifications. Section 4 presents a new algorithm to design the PN controller and section 5 illustrates the proposed algorithm by an explicative example.

2. PN NOTATIONS AND DEFINITIONS

This section introduces brief presentations of some definitions used in this paper. A Petri Net (PN) with m places and n transitions is defined as $N(P,T,Pre,Post)$ where, $P=\{p_i\}_{i=1,\dots,m}$ is not empty finite set of places, $T=\{t_j\}_{j=1,\dots,n}$ is a not empty finite set of transitions, such that $P\cap T=\varnothing$. $Pre: P\times T\rightarrow\mathbb{N}$ is the pre-incidence application. $Pre(p_i,t_j)$ is the weight of the arc from place p_i to transition t_j and $W_{PR}=(w_{ij}^{PR})_{i=1,\dots,m,j=1,\dots,n}\in\mathbb{N}^{m\times n}$ with $w_{ij}^{PR}=Pre(p_i,t_j)$ is the pre-incidence matrix.

$Post: P\times T\rightarrow\mathbb{N}$ is the post-incidence application. $Post(p_i,t_j)$ is the weight of the arc from transition t_j to place p_i and $W_{PO}=(w_{ij}^{PO})_{i=1,\dots,m,j=1,\dots,n}\in\mathbb{N}^{m\times n}$ with $w_{ij}^{PO}=Post(p_i,t_j)$ is the post-incidence matrix. The PN incidence matrix W is defined as: $W=W_{PO}-W_{PR}\in\mathbb{N}^{m\times n}$.

The set of input (respectively output) places of a transition t_j is noted by ${}^*t_j$ (respectively t_j^*). Similarly, the set of input (respectively output) transitions of a place p_i is noted ${}^*p_i$ (resp. p_i^*). A PN is said ordinary if the weights of all arcs are equal to 1.

The state of a PN is given by its current marking which is a mapping $M: P\rightarrow\mathbb{N}$ that assigning to each place of the net a non negative integer number of so-called tokens. A marked PN is noted (N,M_0) where M_0 is the initial marking. The marking of a place p_i at a marking M is denoted by $M(p_i)$. A transition $t_j\in T$ is enabled at a marking M if and only if for each $p_i\in {}^*t_j$, it holds: $M(p_i)\geq Pre(p_i,t_j)$. We note by $M[t_j>$ where $t_j\in T$ is enabled at the marking M. When fired, t_j produces a new marking M', denoted by $M[t_j>M'$. The marking M is said reachable from (N,M_0) if there exist a firing sequence σ such that $M_0[\sigma>M$. The set of reachable markings from M_0 denoted by $R(N,M_0)$.

3. FORBIDDEN STATE PROBLEM

We consider the basic supervisory control problem for designing PN controller that restricts the reachability set of PN model. In the remainder of this paper we assume that, this PN model is identified from the I/O sequences describing all the behaviours of the system according to the algorithm presented in (Bekrar *et al*, 2006b). Hence, our goal is to add some control places in order to guarantee that all the behaviours generated by the identified PN model coincide with those generated by the real system.

Note also that, each $M_i'\in R_G'$ is composed of two parts as follows: $M_i'=\begin{bmatrix}M_{im}'\\ M_{i\bar{m}}'\end{bmatrix}$ where, the first one represents the marking of measurable places and the second one represents the estimated marking of the non measurable places.

The problem herein considered is a FSP. However, it will be considered as a FSTP (see Ghaffari *et al*, 2002). To solve this problem we develop an algorithm that consists in: (1) Identifying the set of the forbidden markings. (2) Determining the set of the forbidden state transitions. (3) Designing the PN controller.

The two first steps of this algorithm are presented in this section and the design of the PN controller will be detailed in the next section.

3.1. Identification procedure of forbidden markings

Contrary to studies proposed to treat the FSP where the forbidden states are defined explicitly by constraints, in our case we must calculate them using the behaviours of the real system and those of the PN model (Fig. 1). The behaviours of the considered system are represented by the set of its reachable states called ξ. When, those generated by the PN modelling the system are represented by its reachability graph called R_G'.

Definition 1: A marking $M_i'\in R_G'$ reachable by the PN that we want to control is said forbidden marking if it's measurable part is not equivalent to any state E_i in ξ (i.e., $\nexists E_i\in\xi: M_{im}'=E_i$) otherwise it is a legal marking.

The set of the forbidden markings can be obtained using the following procedure:

Inputs: ξ and R_G',

Output: M_{fr}.

Begin

1. Initialise the set of forbidden marking: $M_{fr} := \varnothing$.

2. For each marking $M_i^{'} \in R_G^{'}$ do :

   2.1. *If* there exists a state $E_i \in \xi$ such that $M_{im}^{'} = E_i$ then :

   - $M_i^{'}$ is authorised marking.

   2.2. *Else* $M_i^{'}$ is forbidden marking.

   - Update the set of forbidden markings: $M_{fr} := M_{fr} \cup \{M_i^{'}\}$.

   End If.

   End For.

End.

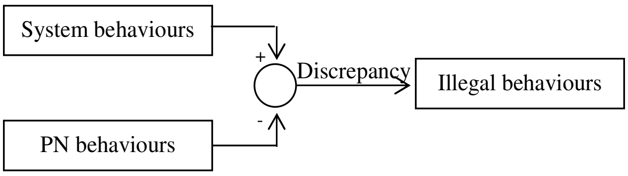

Fig.1. Determining the forbidden behaviours

Once the set of the forbidden markings is obtained, we should determine the set of the transitions leading to or firing from a forbidden marking in order to prevent the firing of these transitions.

3.2. Determination procedure of forbidden state transitions

A state transition of the PN reachability graph, which fires from a marking $M_i^{'} \in R_G^{'}$ and leads to a marking $M_j^{'} \in R_G^{'}$ ($M_i^{'}[t_i > M_j^{'}$) is said forbidden state transition if and only if one of the following conditions is verified:

(a). $M_i^{'}$ is forbidden marking or,

(b). $M_j^{'}$ is forbidden marking.

Proof: the occurrence of forbidden marking can be prevented by avoiding the firing of the transition leading to this marking. Moreover, it is clear that a transition firing from a forbidden marking is a forbidden transition. So, each transition leads to or fires from a forbidden marking is considered as a forbidden transition.

Hence, the set of forbidden state transitions can be obtained using the following procedure:

Inputs: ξ and $R_G^{'}$.

Output: Ψ the set of forbidden state transitions.

Begin

1. Initialise the set of forbidden state transitions: $\Psi = \{\varnothing\}$.

2. For each state transition $(M_i^{'} \xrightarrow{t_i} M_j^{'}) \in R_G^{'}$ such that $M_i^{'}, M_j^{'} \in R_G^{'}$ do:

   2.1. *If* one of the conditions (a) and (b) is verified then: $(M_i^{'} \xrightarrow{t_i} M_j^{'})$ is forbidden state transition.

   - Update Ψ the set of forbidden state transitions: $\Psi := \Psi \cup \{(M_i^{'} \xrightarrow{t_i} M_j^{'})\}$.

   2.2. *Else*, $(M_i^{'} \xrightarrow{t_i} M_j^{'})$ is an authorised state transition.

   End If

   End for

End.

One the forbidden state transitions are determined, the influence paths of forbidden transitions will be calculated and used to design a PN controller. More details about this step will be introduced in the next section.

4. PN CONTROLLER DESIGN

The supervisory control problem considered in this paper can be solved by adding control places defined as follows:

Definition 2: A control place p_{ci} of PN model $(N, M_0^{'})$ is defined by: (i) $M_0^{'}(p_{ci})$: its initial marking, (ii) $Post(p_{ci},.)$: the weighting vectors of the arcs connecting the transitions of $(N, M_0^{'})$ to p_{ci} and, (iii) $Pre(p_{ci},.)$: the weighting vectors of the arcs connecting p_{ci} to the transitions of $(N, M_0^{'})$.

Remark 1: Since the considered PN are ordinary then the arcs weighting values are equal to 1.

Thus, the PN controller design problem consists in determining a set of control places $\{p_{c1},..., p_{ck}\}$ that must be added to the initial PN model $(N, M_0^{'})$ such that, the controlled PN (N_c, M_{0c}) generate the desired behaviours and prevents the forbidden ones.

In order to solve this problem, we propose to use the influence paths of forbidden transitions, calculated from the reachability graph of the PN that we want to control, to determine the control places to be added. Note that, the influence paths of critical place have been used by Ghaffari *et al* (2003) and Basile et *al* (2007b) to treat a FSP specified by GMEC.

To design a PN controller, we must answer to the following questions: (1) do we add control places as much as forbidden state transitions? (2) What are the inputs and the outputs transitions of each added control place? (3) What is the initial marking value of each added control place? In order to answer to these questions, we introduce in the following some definitions which will be used in the remainder of this paper.

Definition 3: An influence path $\pi_i = t_i, M_i, t_j, ..., t_k, M_f$ of a forbidden transition t_i, in the reachability graph of the PN that we want to control, is a directed path composed of a succession of forbidden markings and forbidden transitions. It connects the forbidden transition t_i to a legal marking M_f

such that all other transitions and markings in the path are forbidden.

In the next of the paper the first and the last transition of each path π_i will be respectively noted $t_{in}(\pi_i)$ and $t_{fi}(\pi_i)$.

Remark 2: A forbidden transition t_i can have several influence paths and its influence paths set will be noted $\Pi_i = \{\pi_i^1, \pi_i^2, ..., \pi_i^k\}$, with $k \in \{1,..,|\Pi_i|\}$ and $|\Pi_i|$ is the cardinal of Π_i.

Definition 4: An influence path π_j of a forbidden transition t_j is said included in an influence path π_i of a forbidden transition t_i, noted $\pi_j \subset \pi_i$, if the following conditions are checked simultaneously:

1. $\forall\, t_k \in \pi_j \mid t_k \in \pi_i$, 2. $\forall M_i \in \pi_j \mid M_i \in \pi_i$, 3. π_i can be written using π_j as follows: $\pi_i = t_i M_i ....\pi_j$.

The first and the second conditions allow to verify the inclusion relation of each element of π_j in π_i. These conditions means that, each forbidden transition and each forbidden marking in π_j belongs to π_i. But, these conditions permit to confirm the inclusion relation of each element of π_j in π_i without regarding the secession order of these markings and transitions. For this raison, if the third condition is not verified the inclusion relation becomes intersection relation between two influence paths.

Remark 3: One forbidden transition can have several influence paths that check the inclusion property between them. For each transition $t_i \in \pi_i^j$, we note $|t_i \cap \pi_i^j|$ the number of times that the transition t_i appears in π_i^j.

Based on these definitions, we developed an algorithm that adds control places preventing the occurrence of the forbidden markings generated by the PN model. This algorithm is divided into the following steps:

1. The first one consists in generating the reachability graph $R_G^{'}$ of the PN model that represents all the possible markings reachable from the initial marking $M_0^{'}$, 2. The second step allowing to identify the set of forbidden markings and determining the set of the equivalent forbidden state transitions Ψ, 3. In the third step, we calculate the influence paths of each forbidden transition in Ψ. 4. In this step we verify the inclusion relation between the influence paths of the same transition and between each two distinct forbidden transitions in order to obtain the final set of influence paths $\Pi^{'}$. 5. Finally, we use $\Pi^{'}$ to synthesise a PN controller.

Hence, the PN controller is worked out using the following algorithm:

Inputs: ξ the system states set, $(N, M_0^{'})$ PN model.

Output: controlled PN.

Begin

1. Generate the PN reachability graph $R_G^{'}$ of the PN model $(N, M_0^{'})$.
2. Determine the set of forbidden markings M_{fr} and conclude the set of forbidden state transitions Ψ by executing the procedures introduced in 3.1 and 3.2.
3. For each transition t_i in Ψ do:
   3.1. Calculate the set of the influence paths Π_i.
   3.2. Update Π_i by verifying the inclusion relation between the influence paths of the same forbidden transition as follows:
   - *For* each $\pi_i^m \in \Pi_i$ and $\pi_i^n \in \Pi_i$ do:
     - *If* $\pi_i^m \subset \pi_i^n$ then:
       - Eliminate π_i^m from Π_i.
       - Update Π_i such as: $\Pi_i := \Pi_i \setminus \{\pi_i^m\}$.

       End if.

     End For.

   End For.
4. Put all the updated influence paths of all forbidden transitions in the same set $\Pi^{'}$ such as $\Pi^{'} = \{\Pi_i \mid i \in \{1,...,k\}\}$ (*k* is the forbidden transitions index).
5. Update $\Pi^{'}$ by verifying the inclusion relation between each two influence paths of each two distinct transitions as follows:
   - *For* each $\pi_i^q \in \Pi^{'}, \pi_j^l \in \Pi^{'}$ do:
     - *If* $\pi_i^q \subset \pi_j^l$ then:
       - Eliminate π_i^q from $\Pi^{'}$,
       - Update $\Pi^{'}$ such as $\Pi^{'} := \Pi^{'} \setminus \{\pi_i^q\}$.

       End if.

     End For.
6. *For* each influence path π_i^r in $\Pi^{'}$ do:
   6.1. Determine $t_{in}(\pi_i^r)$ and $t_{fi}(\pi_i^r)$ (the first transition and the last transition) of the considered influence path.
   6.2. Add a control place p_{ci} to the actual PN model.
   6.3. Connect the added control place p_{ci} to the actual PN model by arcs of weight equals to 1. $t_{in}(\pi_i^r)$ will be output of p_{ci} and $t_{fi}(\pi_i^r)$ will be input of p_{ci}.

6.4. Mark p_{ci} with initial marking equal to $|\, t_{in}(\pi_i^r) \cap \pi_i^r \,|$.

6.5. If there is several influence paths which have the same first and last transitions then, mark the control place p_{ci} with initial marking equal to $\max(|\, t_{in} \cap \Pi^{'})$.

End For.

End.

This algorithm allowing to add control places to the PN that we want to control by analysing the influence paths of the forbidden transitions. The number of the control places to be added is not necessary equivalent to the number of the forbidden state transitions. In this algorithm, each influence path π_i^r in the final influence paths set $\Pi^{'}$ characterise a control place to be added. Indeed, each added control place is an input of the first transition and an output of the last transition in π_i^r. The initial marking of each added control place is the number of appearance times of this transition in its influence path. This represents in reality the firing number of this transition.

The control method proposed in this paper is a structural. It addition, the controlled behaviours do not contain neither dangerous nor blocking markings. This implies that the controlled behaviours of the system are maximally permissive within the specifications.

5. ILLUSTRATIVE EXAMPLE

To illustrate the proposed algorithm, let as consider the example of a system characterised by the identified PN of Fig. 2 and the set of behaviours given by:

$$R_G = \left\{ \underbrace{\begin{bmatrix}0\\0\\3\\2\end{bmatrix}}_{M_0}, \underbrace{\begin{bmatrix}1\\0\\2\\1\end{bmatrix}}_{M_1}, \underbrace{\begin{bmatrix}0\\1\\2\\1\end{bmatrix}}_{M_2} \underbrace{\begin{bmatrix}0\\0\\2\\1\end{bmatrix}}_{M_3}, \underbrace{\begin{bmatrix}2\\0\\1\\0\end{bmatrix}}_{M_4}, \underbrace{\begin{bmatrix}1\\1\\1\\0\end{bmatrix}}_{M_5}, \underbrace{\begin{bmatrix}1\\0\\1\\0\end{bmatrix}}_{M_6} \right\}$$

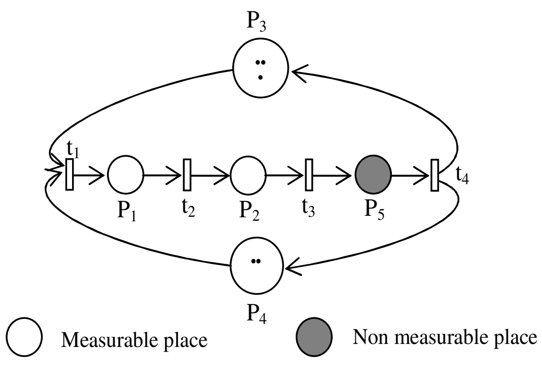

Fig. 2. The PN model.

Firstly we elaborate the PN reachability graph which is represented in Fig. 3.

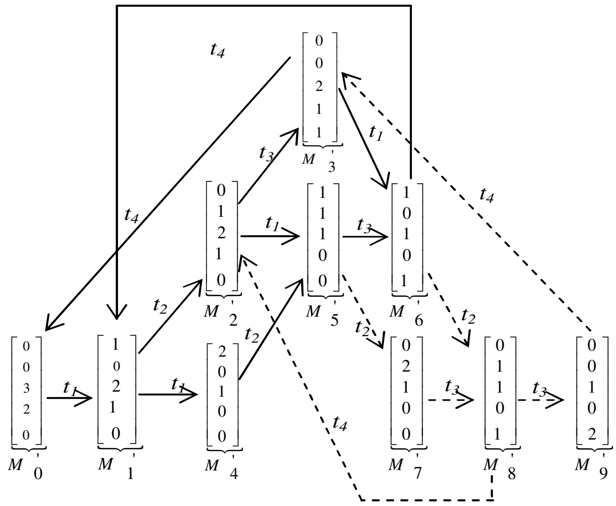

Fig. 3. The PN reachability graph.

The set of forbidden markings is: $M_{fr} = \{ M_7^{'}, M_8^{'}, M_9^{'} \}$. These markings must be removed from $R_G^{'}$ together with their arcs coming from other markings to these forbidden markings or going from these forbidden markings to other markings in $R_G^{'}$. The set of the equivalent forbidden state transitions is:

$$\Psi = \{ (M_5^{'} \xrightarrow{t_2} M_7^{'}), (M_7^{'} \xrightarrow{t_3} M_8^{'}), (M_8^{'} \xrightarrow{t_3} M_9^{'}), (M_9^{'} \xrightarrow{t_4} M_3^{'}), (M_6^{'} \xrightarrow{t_2} M_8^{'}), (M_8^{'} \xrightarrow{t_4} M_2^{'}) \}$$

On the other hand, the influence paths of forbidden transition in Ψ are:

The influence paths of t_2 are: $\pi_2^1 = t_2 M_7^{'} t_3 M_8^{'} t_3 M_9^{'} t_4 M_3^{'}$, $\pi_2^2 = t_2 M_8^{'} t_3 M_9^{'} t_4 M_3^{'}$, $\pi_2^3 = t_2 M_7^{'} t_3 M_8^{'} t_4 M_2^{'}$, $\pi_2^4 = t_2 M_8^{'} t_4 M_2^{'}$. Thus, $\Pi_2 = \{\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4\}$.

The influence paths of t_3 are: $\pi_3^1 = t_3 M_8^{'} t_3 M_9^{'} t_4 M_3^{'}$, $\pi_3^2 = t_3 M_9^{'} t_4 M_3^{'}$, $\pi_3^3 = t_3 M_8^{'} t_4 M_2^{'}$. Thus, $\Pi_3 = \{\pi_3^1, \pi_3^2, \pi_3^3\}$.

The influence path of t_4 are: $\pi_4^1 = t_4 M_3^{'}$, $\pi_4^2 = t_4 M_2^{'}$. Thus, $\Pi_4 = \{\pi_4^1, \pi_4^2\}$.

By updating the influence path sets $\Pi_{i, i\in\{2,3,4\}}$: we remark that $\pi_3^2 \subset \pi_3^1$ then $\Pi_3 = \{\pi_3^1, \pi_3^3\}$. Hence, the set of all influence paths of all forbidden transitions is: $\Pi^{'} = \{\Pi_2, \Pi_3, \Pi_4\}$.

In addition, by updating $\Pi^{'}$: we remark that: $\pi_4^2 \subset \{\pi_3^3, \pi_2^4, \pi_2^3\}$, $\pi_4^1 \subset \{\pi_3^1, \pi_2^2, \pi_2^1\}$, $\pi_3^3 \subset \pi_2^3$, $\pi_3^1 \subset \pi_2^1$.

So, $\Pi^{'} = \Pi_2 = \{\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4\}$.

Finally, we can remark that all the influence paths in $\Pi^{'}$ have the same input and output transitions. These transitions are respectively $t_{in} = t_2$ and $t_{fi} = t_4$. Therefore, we add control place p_{c1} with initial marking $M_{0c}(p_c) = \max(|\, t_2 \cap \Pi^{'}) = 1$.

This place is an input of t_2 and output of t_4 as depicted in the figure 4.

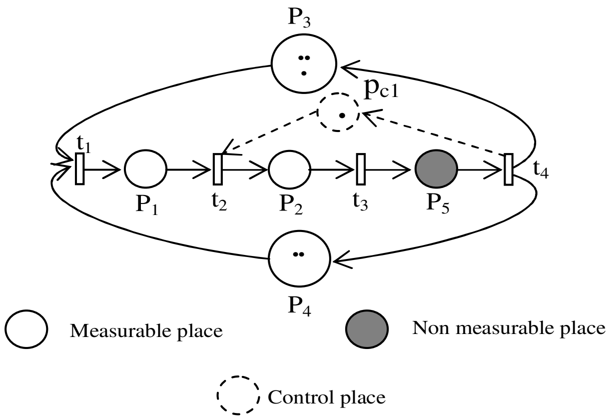

Fig. 4. The controlled PN model

6. CONCLUSION

This paper presents a structural method to solve a FSP of DES modelled by ordinary PN. The PN model to be controlled is marking partially measurable and transitions controllable. Using system behaviours and PN ones, forbidden states are identified and forbidden state transitions are determined. The influence paths of forbidden transitions are used to synthesis a PN controller. The later is maximally permissive within the specifications which, guarantees only desired behaviours.

Future research concerns the extension of this work to PN with uncontrollable transitions.

REFERENCES

Achour, Z., N. Rezg and X. Xie (2004), Supervisory control of partially observable marked graphs, *IEEE Transactions on Automatic Control*, **Vol. 49**, N°11, pp. 2007- 2011.

Achour, Z., N. Rezg and X. Xie (2005), On the existence of Petri Net controller for discrete event systems under Partial Observation, *16th IFAC World Congress.*

Basile, F., P.Chiacchio and A. Giua (2007a), An optimisation approach to Petri Net monitor design, *IEEE Transaction on Automation and Control*, **Vol. 52**, N° 2, pp. 306-311.

Basile, F., C. Carbone and P. Chiacchio (2007b), Feedback control logic for backward conflict free choice Nets, *IEEE Transaction on Automatic Control*, **Vol. 52**, N° 3, pp. 387-400.

Bekrar, R., N. Messai, N. Essounbouli, A. Hamzaoui, and B. Riera (2006a), Off-line identification for a class of discrete event systems using safe Petri Nets, *Proceeding of the 3rd IFAC Workshop on Discrete Event System Design (DESDes'06).*

Bekrar, R., N. Messai, N. Essounbouli, A. Hamzaoui, and B. Riera (2006b), Identification of discrete event systems using ordinary Petri Nets, *Proceeding of 2006 Workshop on Advanced Control and Diagnosis (IAR-ACD'06).*

Dotoli, M., P. F. Maria and A. M. Mangini (2006), An optimisation approach for identification of Petri Nets, *Proceedings of the 8th international Workshop on Discrete Event Systems*, pp. 332-337.

Ghaffari, A., N. Rezg and X. Xie (2002), Algebraic and geometric characterization of Petri Net controllers using the theory of regions, *Proceedings of 6th Worksop on Discrete Event Systems,* pp. 219- 224.

Ghaffari, A., N. Rezg and X. Xie (2003), Feedback control logic for forbidden state problems of marked graphs: application to a real manufacturing system, *IEEE transaction on Automatic Control*, **Vol. 48**, N° 1,pp. 18-29.

Giua, A., and X. Xie (2004), Control of safe ordinary Petri Nets with marking specifications using unfolding, *proceedings of the 7th IFAC Workshop on Discrete Event Systems*, pp. 61-66.

Giua, A., and X. Xie (2005), Control of safe ordinary Petri Nets using unfolding, *Discrete Event Dynamic Systems*, **Vol. 15**, N° 4, pp. 349-373.

Lee, E. J.,A. Toguyeni and N. Dangoumau (2006), A Petri Net based approach for the synthesis of parts' controllers for reconfigurable manufacturing systems, *Proceeding of SICE-ICASE International Joint Conference*, pp. 5567 – 5572.

Ramadge, P. J., and W. M. Wonham (1987), Supervisory control of class of Discrete Event Process, *SIAM J. Control and Optimisation*, **Vol. 25**, N° 1, pp. 206–230.