SUPERVISORY SYNTHESIS FOR SAFE PETRI NETS

Carmen Bratosin, Hassane Alla, Simona Iuliana Caramihai

Automatic Control and Computers Faculty, "Politehnica" University of Bucharest,
Splaiul Independentei, No. 313, Sector 6, Bucharest 77206, Romania.
Laboratoire d'Automatique de Grenoble, ENSIEG - BP 46, 38402
St Martin d'Heres Cedex, France.
carmen_toma_ro@yahoo.com, hassanne.alla@inpg.fr, sic@ics.pub.ro

Abstract: This paper presents an approach based on the reachability tree for the computation of a structural controller for the class of safe Petri Nets. The proposed approach computes the less restrictive controller for the safe Petri Nets. The first step determines the set of frontier states that have to be forbidden, ensuring that the plant will respect the specifications (the method is applied to the closed loop Petri Net model). Equivalence between this set of states and a set of linear constraints is formally proved. This equivalence is crucial since it provides a link between the supervisory control theory and the Petri net invariant method. A set of control places is added to the closed loop Petri model and these control places will guarantee the desired behavior of the plant.

Keywords: Safe Petri Nets, Supervisory Synthesis, Reachability Graph, Maximally Permissive Supervisor

1. INTRODUCTION

In the last decade many researchers have tried to develop a controller synthesis for Petri Nets (PNs). The main idea is to find an efficient way to construct control places that do not let the plant to reach forbidden states.

Ramadge and Wonham (1987) defined the concepts of DES control theory and they formalized the sufficient and necessary conditions for the existence of a maximally permissive controller, i.e. the less restrictive controller for the evolution of the plant. If for languages and automata models the researchers have found maximally permissive solutions of the control problem (Ramadge and Wonham, 1987; Kumar, 1991), for Petri Nets the field is still open.

In Giua et *al* (1992), inequality constraints have been defined and named General Mutual Exclusions Constraints (GMEC). Also, the concept of monitor has been introduced. A monitor is a place whose initial marking represents the available units of a resource and whose outgoing and incoming transitions represent, respectively, the acquisition and release of units of the resource. A simple computation method for monitor places was proposed. For PNs with all transitions controllable,

the GMECs may be easily enforced on the net plant. The method is applicable for safe and conservative PNs. But the method is not always applicable for PNs with uncontrollable transitions (i.e. that cannot be inhibited by an external action). In Giua et *al* (1993], a modified method was proposed for marked graphs with uncontrollable transitions.

A method for the computation of a maximally permissive controller using theory of region is presented in Ghaffari and Xie (2000). The main idea is to reduce the reachability graph to the set of admissible markings, i.e., markings that respect the specifications. Control places are added to the original network, the closed loop plant respecting the specifications. The theory of region is used on the reachability graph for the computation of the control places. The advantage of this method is that gives a maximally permissive solution of the control problem, but the computation method is difficult and does not guarantee a solution.

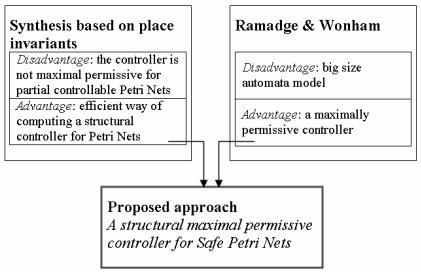

Fig. 1. Proposed approach

The goal of our work is to propose a controller synthesis for the safe Petri Net (SPN). The idea is to find a relation between the theory presented by Ramadge and Wonham and the computation of control places for PN models based on place invariants (Figure 1). Since the controller synthesis from the theory of Ramadge and Wonham founds the maximally permissive controller, the relation between the two theories will ensure a simple way (using place invariants) to compute a structural maximally permissive controller for PN models.

In this paper, the classical technique for finding the set of forbidden states from the reachability graph, using the specifications models, is used. After finding these states linear constraints are deduced. A major difficulty is to find an equivalence between constraints and the forbidden states in order to ensure that the computed supervisor is the less restrictive one i.e., maximally permissive supervisor. For general PNs this property is rarely verified. But for safe PNs, except in very particular cases that will be specified this equivalence is true.

The paper is organized as follows. Section 2 presents the proposed supervisory synthesis. Section 3 defines the sufficient and necessary conditions for which the maximally permissive supervisor is found. The method of computation the control places is presented in section 4. Finally, Section 5 concludes with directions for future research.

2. SUPERVISORY SYNTHESIS

In this section, the proposed approach is presented. The approach's goal is to determine a supervisor ensuring the specification obedience for the supervised plant. The supervisor is computed for the closed loop plant. After the construction of the closed loop model, the set of forbidden states are determined. In the end, for each forbidden state, a control place is computed. The necessary and sufficient condition for the realization of the maximally permissive supervisor is given.

2.1 Definitions

The following definitions will be used for better understanding the approach.

The reachability graph is an automata $\mathbb{R} = \{S, \Sigma, \delta, m_0\}$ where S is the states set, Σ is the events set, $\delta : S \times \Sigma \rightarrow S$ the evolution function, s_0 is the initial state. This graph corresponds to all the possible evolutions of the PN.

Definition 1: An event is called **uncontrollable** if their occurrence may not be inhibited by an external action. The set of uncontrollable events is Σ_u. Now, the set $\Sigma = \Sigma_c \cup \Sigma_u$, where Σ_c is the set of controllable events.

Let Σ^* denote the set of all finite string over Σ including the null string ε. The language generated by $\mathbb{R}$ is: $L_R := \{w \mid w \in \Sigma^* \text{ and } \delta(m_0, w) \text{ is defined}\}$

The specification for the PN that generates $\mathbb{R}$ is a pair (S, γ), where $\gamma: S \rightarrow \{0, 1\}$. The specifications restrict the behavior of the system, i.e., the desired language of the systems is $L_{Rd} := \{w \mid w \in L_R \text{ and } \gamma(\delta(m_0, w)) = 1\}$

Definition 2: The set of forbidden states is noted $\mathcal{M}_F$ and is expressed: $\mathcal{M}_F = \{m \in S \mid \exists\, w \in L_R \setminus L_{Rd} \text{ and } m = \delta(m_0, w)\}$ i.e., a state is **forbidden** if it is a reachable state and if it violates the specifications.

Definition 3: The set of dangerous states is: $\mathcal{M}_D = \{m \in S \mid \exists\, w \in \Sigma_u^* \text{ and } \delta(m, w) \in \mathcal{M}_F\}$ i.e., a **dangerous state** is a reachable marking from which there is at least a sequence of uncontrollable transitions who leads to a forbidden state. We will consider, also, that a forbidden state is a dangerous state.

Definition 4: The set $\mathcal{M}_A$ of admissible states is the greatest set of reachable states so as: $\mathcal{M}_A \cap \mathcal{M}_D = \varnothing$

If $m \in \mathcal{M}_A$ and $\delta(m, e) \in \mathcal{M}_F$ than $e \in \Sigma_c$, i.e., the passage from an admissible state to a dangerous state is made by the firing of a controllable transition.

Furthermore, $\mathcal{M}_A$ is the most permissive behavior respecting the specifications.

Definition 5: A supervisor is **maximally permissive** (MPC) if all the admissible markings of $\mathcal{M}_A$ are reachable under supervision and all the firings of a transition, which cause the evolution of the plant from an admissible state to a non-admissible one, are inhibited.

Definition 6: The set of border states is: $\mathcal{M}_B = \{m \in \mathcal{M}_D \mid \exists\, e \in \Sigma_c \wedge \exists\, m_a \in \mathcal{M}_A$, s.t. $\delta(m_a,e)=m\}$

i.e., the set of dangerous states which are reached by the firing of a controllable transition from an admissible state.

For finding the maximally permissive supervisor it is just necessary and sufficient to not allow the plant to reach the border states. All the others states of M_D will be no longer reachable

2.2 Closed Loop Plant Model

Usually, a plant is forced to respect several specifications. The specifications may concern: an order for the incoming parts into the plant, transfer order for some of the machines, production cycles etc. In this paper, the specifications are defined like the set of sequencial constraints to which the plant will be constrained. This type of specification can be easily modeled by a PN.

The closed loop model model corespond to the plant under the influence of the specifications. The model is obtained by the synchronization of the PN plant model and the PN specifications. The technique is a structural one, which does not depend of the initial marking. The synchronization consists in the fusion of the transitions with the same semnification.

2.3 From border states to linear constraints

For the forbidden states that are identified from the closed loop PN structure, the quasi-PN notion is introduced.

Remark 1: Let T_j be an uncontrollable transition of the plant PN. The following notation is used:

$^\bullet T_j^p \Leftrightarrow$ the set of plant places that are input places for T_j

$^\bullet T_j^s \Leftrightarrow$ the set of specificat ion places that are input places for T_j

For a plant component (i.e., place, transition, state etc.) the notation X^p is used, and for the specification component the notation is X^s.

Definition 7: A **quasi-PN** is a closed loop model PN for which the firing rules are:

- if T_j is a controllable transition it is fired if $\forall P_i \in {}^\bullet T_j^p \cup {}^\bullet T_j^s \quad m(P_i) = 1$
- if T_j is an uncontrollable transition it is fired if $\forall P_i \in {}^\bullet T_j^p \quad m(P_i) = 1$

Property 1: A state is a **forbidden state** if from this state an uncontrollable transition is not firable in the closed loop PN model and is firable in the equivalent quasi-PN.

Proof: If an uncontrollable transition is firable in the quasi-PN this means that it is enabled by the plant. If it is not enabled in the PN this means that it is not enabled by the specifications. The state in this case does not respect the specification and, from definition 3, it is a forbidden states, i.e. $\exists\, e \in L_R \wedge e \notin L_{Rd}$ (a valid firing sequence exists for the plant model, but does not exist for the closed loop model).

The objective is to compute control places such the states of the set $\mathcal{M}_B$ are never reachable under control. For applying the method based on place invariants, linear constraints must be determined. For each border state a linear constraints is computed using the following property.

Property 2: Let $M_j = (P_{j1}P_{j2} \ldots P_{jr}) \in \mathcal{M}_B$ (the state is given only by the marked places) be a forbidden border state with r marked places for a closed loop safe PN. The constraint:

$$\sum_{i=1}^{r} m(P_{ji}) \le r - 1 \qquad (1)$$

will force the closed loop safe PN model not to reach the border state M_j.

Proof: For the forbidden border state M_j, $\sum_{i=1}^{r} m_j(P_{ji}) = r > r-1 \Leftrightarrow M_j$ is an unreachable state for the PN under constraint (1).

Using property 2, for each border state a constraint can be found. The set of the constraints is denoted as C.

For the computation of the MPC, an equivalency between the set $\mathcal{M}_B$ and the set C must exist ($\mathcal{M}_B \Leftrightarrow C$). The implication $\mathcal{M}_B \Rightarrow C$ is obvious from Property 2. The problem is that the implication C => $\mathcal{M}_B$ is not always true.

In the following section, examples are given for better understandings when a constraint can forbid admissible markings. At the end of the section, the sufficient and necessary condition for the computation of the MPC is given.

3. NECESSARY AND SUFFICIENT CONDITION

Two examples are given for showing the possible cases that can occur in the computation of the supervisor.

Example 1: Let us consider a manufacturing system composed by two machines. The beginning of the jobs on each machine is associated with the events c_1, for the first machine, and, c_2 for the second one. These events are controllable. The events f_1 and f_2 correspond to the end of the job on the machines and they are uncontrollable events. The plant specification imposes an alternation of the events f_1 f_2. The closed loop PN of the plant is presented in Figure 2. The places P_1, P_2, P_3, and P_4 are the places that model the plant. The places P_5 and P_6 represent the specification PN model.

Analyzing the reachability graph, the set of forbidden states (Property 1) is obtained: $\mathcal{M}_F$ = { M_{f1}=[0 1 1 0 1 0]^T, M_{f2}= [0 1 0 1 1 0]^T , M_{f3}= [1 0 0 1 0 1]^T , M_{f4}= [0 1 0 1 0 1]^T }.

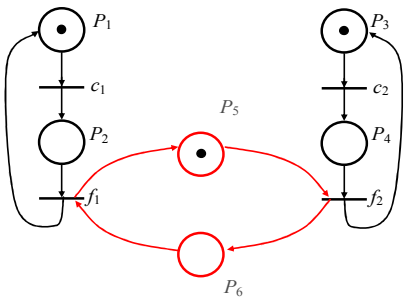

Fig. 2. Closed loop PN for example 1

The first two states, M_{f1} and M_{f2}, are forbidden because from this states the uncontrollable event f_1 is fired in the reachability graph of the quasi-PN, but not fired in the reachability graph of the closed loop PN. The states M_{f3} and M_{f4} are also forbidden states (uncontrollable event f_2).

Because the states M_{f1} and M_{f2} are reached by the firing of the transition associated with the controllable event c_1, and M_{f3} and M_{f4} are reached by the firing of the transition associated with the controllable event c_2, the set of forbidden states is also the set of the border states (Definition 6) $\mathcal{M}_F$ = $\mathcal{M}_B$.

From these forbidden states, the following set of constraints is deduced (property 2):

$$C = \{m(P_2)+m(P_3)+m(P_5) \le 2; m(P_2)+m(P_4)+m(P_5) \le 2;$$
$$m(P_1)+m(P_4)+m(P_6) \le 2; m(P_2)+m(P_4)+m(P_6) \le 2;\}$$

Each constraint forbids only the corresponding state. It can be verified that all the admissible states respect the set of constraint C: $\mathcal{M}_A$ = { M_{a1}=[1 0 1 0 1 0]^T , M_{a2}= [0 1 1 0 0 1]^T , M_{a3}= [1 0 0 1 1 0]^T , M_{a4}= [1 0 1 0 0 1]^T }.

In this example, the equivalence between the set $\mathcal{M}_B$ and the set C ensures the computation of the MPC. The case presented above is the usual case for real systems such as manufacturing plants. Some particular cases may also appear.

Even if a constraint interdicts more than one state, in the most cases the MPC is still found. If all the states that are equivalent with one constraint are forbidden states, the supervisor will be a MPC. But, it can be found cases when a constraint forbids states from the set of admissible states. Example 2 presents this case.

Example 2: In Figure 3 is presented a closed loop model of a plant composed by two machines that are synchronized on the event uncontrollable f (end of the job on the two machines). The events c_1 and c_2 represents the beginning of the tasks on the first machine and the second machine, and are controllable events.

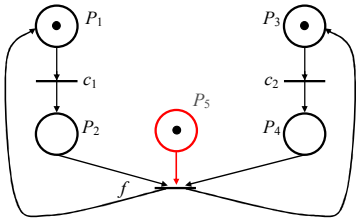

Fig. 3. Closed loop PN for example 2

The specification is represented by place P_5, and specifies that the event f must occur just once. From the reachability graph, the forbidden and border state is M_1 = [0 1 0 1 0]^T, because from this state in the plant PN the event f occur, but the place P_5 is not marked. This means that the end of job on the two machines is not allowed by the specification. The

constraint for this state is: $m(P_2)+m(P_4)\leq 1$. But this constraint forbids also the state $M_2=[\,0\;1\;0\;1\;1\,]^T$ which is an admissible state. In this case the MPC cannot be computed.

Remark 2*:* The example presented above, is a case rarely met in real systems.

Let us give first the following definition.

Definition 8: Let us consider a PN with *n* places and M_j, M_k two reachable markings for a given initial state. We say that M_k is greater than M_j ($M_k \gtrsim M_j$) if:

$$M_k \gtrsim M_j \Leftrightarrow \forall\, i\in[1,n]\ m_k(P_i)\geq m_j(P_i) \text{ and } \exists\, l\in[1,n]\ \ m_k(P_l)>m_j(P_l) \qquad (2)$$

In the first example the computation of the MPC is ensured by the fact that all the states are different in terms of markings. In the example 2, we have shown that are cases when a constraint will forbid more that one marking. In the safe PN, these cases appear when states that are greater than a border state exist.

Now, the necessary and sufficient condition that ensures the computation of the MPC can be given.

Property 3: The maximal permissive supervisor can be computed if and only if $\forall\ M_j\in\ \mathcal{M}_B\ \nexists\ M_k\in\mathcal{M}_A$ such that $M_k \gtrsim M_j$.

Proof: *Necessary condition*: Suppose that $\exists\ M_k\in\mathcal{M}_A$ and $\exists\ M_j\in\ \mathcal{M}_B$ such that $M_k\gtrsim M_j$. Because M_j is a border state, a linear constraint is constructed using property 2:

$$\sum_{i=1}^{r} m(P_i)\leq r-1$$

The constraint forbids M_j because $\sum_{i=1}^{r} m_j(P_i)=r>r-1$. Since $M_k\gtrsim M_j$, then $\sum_{i=1}^{r} m_k(P_i)=r>r-1$. The state M_k is not longer reachable under control. The MPC cannot longer be computed, the computed supervisor forbids an admissible state. This means that the supposition is false.

Sufficient condition: If $\forall\ M_j\in\mathcal{M}_B\ \nexists\ M_k\in\mathcal{M}_A$ such that $M_k \gtrsim M_j$, this means that all the admissible states will respect the constraint obtained from $\forall\ M_j\in\mathcal{M}_B$. Since all the admissible states are reached under the control, the MPC is computed.

Property 3 gives the class of safe PN for which the MPC is computed. If a safe PN has all the states different (there is no relation $\gtrsim$ between its states), then the supervisor will be a MPC.

A special class of safe PN is the class of conservative safe PNs.

Property 4: For a conservative safe PN the order relation $\gtrsim$ cannot exist between its states.

Proof: In a conservative safe PN each reachable state *M* verifies a marking invariant:

$$\sum_{i=1}^{n} l_i m(P_i)=b \qquad (3)$$

where l_i and b are integer positive values. It can easily be observed that in a conservative PN the relation order given in Definition 7 never exists.

The condition and the class of safe PNs for which the maximally permissive supervisor is computed were given in this section. The only step for obtaining the supervised PN is the computation of the control places that is presented in the next section.

4. COMPUTATION OF THE CONTROL PLACES

The control places are computed using Yamalidou et *al.* (1996) method. One advantage of this method is that it is a numerically efficient manner of finding a supervisor. The goal of the method is to force the plant to respect constraints of the form (4).

$$\sum_{i=1}^{n} l_i m(P_i)\leq \beta \qquad (4)$$

where $m(P_i)$ represents the ith element *i* of a vector of state *M*, and l_i and β are integer, positive values.

For a PN with *n* places and *m* transitions, the incident matrix of the net is $D^p\in Z^{n\times m}$. If we have n^c constraints of the form (5), a matrix inequality can be written:

$$Lm^p\leq b \qquad (5)$$

where $m^p\in Z^n$, is the state vector of the plant, $L\in Z^{nc\times n}$, and $b\in Z^{nc}$. Assuming that the closed loop plant has the following Petri net structure:

$$W=\begin{bmatrix} W^p \\ W^c \end{bmatrix}\quad m_0=\begin{bmatrix} m_0^p \\ m_0^c \end{bmatrix} \qquad (6)$$

the supervisor can be computed like conforming with the following theorem.

Theorem 1 (Yamalidou et *al.* (1996)**) Invariant based supervisory synthesis**.

If

$$b - Lm_0^p \geq 0 \qquad (7)$$

then a PN supervisor, $D^c \in Z^{nc \times m}$ with initial state $m_0^c \in Z^{nc}$

$$W^c = -LW^p \qquad (8)$$

$$m_0^c = b - Lm_0^p \qquad (9)$$

enforces constraint (7) when included in the closed loop plant, assuming that the plant's transitions are controllable and observable.

If inequality (7) is not true, then the constraints can not be enforced by any supervisor since the initial marking of the plant lies outside the range allowed by the constraints.

Theorem 1 can be applied in our presented approach, even if the PN contains uncontrollable events. The constraints forbid only the border states (states that are reached by the firing of controllable transitions), the control places will not interdict the firing of uncontrollable transitions.

Example 4: For the PN from the example 1, the set of constraints is:

$$C = \{m(P_2)+m(P_3)+m(P_5) \leq 2; m(P_2)+m(P_4)+m(P_5) \leq 2;$$
$$m(P_1)+m(P_4)+m(P_6) \leq 2; m(P_2)+m(P_4)+m(P_6) \leq 2;\}$$

A reduction of the constraints number can be made using the place invariants of the plant PN ($m(P_1)+m(P_2)=1$, $m(P_3)+ m(P_4)=1$). The resulted constraints are: $m(P_2)+ m(P_5) \leq 1$; $m(P_4)+ m(P_6) \leq 1$.

Remark 3: An algorithm for the systematic reduction of constraints was developed and will be presented in a future paper.

The computation of the control places is made using theorem 1. The supervised PN is obtained (figure 4).

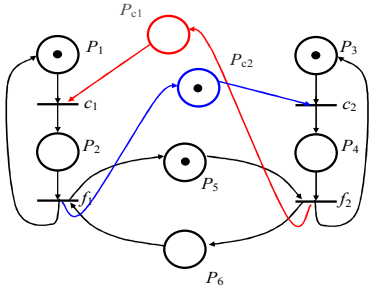

Fig. 4. Supervised PN

5. CONCLUSIONS

In this paper was presented an approach based on the reachability tree for the computation of a structural supervisor for the class of safe Petri Nets. The equivalence between a set of border states and a set of linear constraints was formally proved. This equivalence is crucial since it provides a link between the supervisory control theory and the Petri net invariant method. A set of control places is added to the closed loop Petri model and these control places will guarantee the desired behavior of the plant. The approach is applicable for a class of safe PN, class that is determined by Property 3. The problem of uncontrollable transitions is resolved using the Kumar algorithm for finding the set of border states.

REFERENCES

Wonham, W.M. and J.G. Ramadge (1987). On the Supremal Controllable Sublanguage of a Given Language. In: *Siam J. Control and Optimization*, **Vol. 25. No. 3,** pp. 637-659.

Kumar, R. (1991). *Supervisory Synthesis Techniques for Discreet Event Dynamical Systems.* PhD thesis, University of Texas AT Austin.

Giua, A., F. DiCesare and M. Silva, (1992). Generalized Mutual Exclusion Constraints for Nets with Uncontrollable Transitions. In: *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (Chicago, USA)*, pp. 974-799.

Giua, A., F. DiCesare and M. Silva (1993). Petri Net Supervisors for Generalized Mutual Exclusion Constraints. In: *Proc. 12th IFAC World Congress (Sidney, Australia)*, **Vol. 1**, pp. 267-270.

Yamalidou, K., J.O. Moody, M.D. Lemmon, and P.J. Antsaklis (1996). Feedback control of Petri nets based on place invariants. In: *Automatica*, **vol. 32**, **no. 1**, pp. 15 – 28.

Ghaffari, A., N. Rezg and X. Xie (2001). Design of a live and maximally permissive Petri Net controller using the theory of region. In: *IEEE Trans. Robotics & Automation* **Vol. 19(1)**, pp. 137 – 141.