Proc. IFAC WODES04: 7th Workshop on Discrete Event Systems (Reims, France), Sep 2004.

# CONTROL OF SAFE ORDINARY PETRI NETS WITH MARKING SPECIFICATIONS USING UNFOLDING

**Alessandro Giua** * **Xiaolan Xie** **

* *DIEE, Università di Cagliari, Piazza d'Armi, 09123 Cagliari, Italy.* giua@diee.unica.it
** *INRIA/MACSI Team, ISGMP – Bat A, Ile du Saulcy, 57045 Metz, France.* xie@loria.fr

Abstract: In this paper we deal with the problem of controlling a safe place/transition nets so as to avoid a set of forbidden markings $\mathcal{F}$. We say that a given set of markings has property REACH if it is closed under the reachability operator. We assume that all transitions of the net are controllable and that the set of forbidden markings $\mathcal{F}$ has the property REACH. Under these assumptions we show that using the technique of unfolding is possible to efficiently design a maximally permissive supervisor to solve this control problem. The supervisor takes the form of a set of control places to be added to the unfolding.

Keywords: Petri nets, unfolding, control of discrete event systems.

## 1. INTRODUCTION

The use of *partial order methods* for the efficient verification of *concurrent systems* is a technique that has been used by several authors in the last 10-15 years. In particular, a Petri net is a natural model for this approach because it has primitives to explicitly capture the notion of precedence and independence between events.

The interleaving of concurrent sequences often leads to the well-known problem of *state space explosion* that hinders the applicability of all those Petri net analysis techniques, e.g. the reachability graph, that are based on the exhaustive search over the set of reachable markings. However, the sets of states introduced by concurrency are for the most part intermediate markings that are irrelevant to determine the properties of the system: what matters is the unique marking reached by the firing of all these concurrent sequences.

As Valmari (1994) has lucidly explained, this fact has motivated research along at least two different lines.

(a) A first approach is to let one (or at least as few as possible) interleaving represent all its equivalent interleavings: the notion of *stubborn set* (Valmari, 1991) and *persistent set* (Godefroid, 1996) is inspired by this idea.

(b) A second approach consists in replacing the reachability graph by a net structure which captures the concurrent executions, and does not explicitly show individual interleavings. This technique is based on the *unfolding* of a (bounded) Petri net into an *occurrence net*. A finite prefix of the unfolding can be used to characterize the set of all reachable markings without having to enumerate them (McMillan, 1995; Esparza *et al.*, 2002). Recently, this approach has also been extended to unbounded nets (Neumair, 2002). Note that the occurrence net is much simpler that the original Petri net and can usually be validated using structural analysis.

Although these two type of techniques have proved to be a powerful instrument in the *verification* of concurrent systems, the application of these techniques to the *control* of discrete event systems has not received a lot of attention. We recall here some contributions in this area.

Hellgren *et al.* (1999) have used persistent sets to design supervisors for deadlock avoidance.

Observability and diagnosis are closely related to control: Aghasaryan *et al.*, (1988) were the first to use unfolding for fault detection and diagnosis in distributed systems. This approach has also been extended in two subsequent papers by Benveniste *et. al* (2003a and 2003b).

Recently, in a series of papers He and Lemmon (2000a, 2000b, 2002) have presented an original approach based on unfolding for liveness verifica-

tion and enforcing. However we have shown (Xie and Giua, 2004) that some key results of these papers need to be refined. As a result, although we still strongly believe that unfolding is an interesting and potentially fruitful technique for Petri net control, the applicability of unfolding for Petri net supervision is still an open issue.

In the paper we consider discrete event systems modeled by safe place/transition nets. The control problem we consider can be framed within the theory of Supervisory Control (Ramadge and Wonham, 1989). In particular, we consider a control specification that requires avoiding a set of forbidden marking $\mathcal{F}$. In the current state of investigation, we assume that all transitions are controllable, i.e., they can be disabled by a controlling agent called supervisor that must enforce the specification.

We use a set of finite prefixes of the unfolding, that we call order 1, to characterize the reachability set of the original net.

We restrict our attention to a special class of forbidden marking specification that have a property we call REACH: once a forbidden marking is reached, all markings reachable from it will also be forbidden. This has a nice advantage over the unfolding structure: if a configuration (i.e., a set of transition firings) is forbidden, any larger configuration should also be forbidden. We show that in this case a simple control structure - that consists in a set of places to be added to the order 1 prefix - can be used to implement a maximally permissive control policy that enforces the specification.

The approach we present in the paper requires an exhaustive enumeration of the set of forbidden markings. It has however the advantage of allowing one to construct a maximally permissive supervisor in the form of a "controlled" occurence net (i.e., an occurence net with the addition of control places) using a procedure where where the set of markings of the plant needs not be exhaustively enumerated. The closed loop system in this approach can also be represented by this controlled occurence net.

## 2. BACKGROUND ON PETRI NETS

In this section we recall the formalism used in the paper. A more detailed introduction to Petri nets can be found in (Murata, 1989).

The Petri net model considered in this paper is an *ordinary Place/Transition net* (P/T net) denoted $N = (P, T, F)$, where $P$ is a set of $m$ places; $T$ is a set of $n$ transitions; $F \subseteq (P \times T) \cup (T \times P)$ is the flow function that specifies the arcs from places to transitions and from transitions to places.

The *preset* and *postset* of a node $x \in P \cup T$ are denoted ${}^\bullet x \triangleq \{x' \,|(x', x) \in F\}$ and $x^\bullet \triangleq \{x' \,|(x, x') \in F\}$ while ${}^\bullet x^\bullet = {}^\bullet x \cup x^\bullet$. Node $x$ is a *source* (resp., *sink*) if ${}^\bullet x = \emptyset$ (resp.; $x^\bullet = \emptyset$).

Given two nodes $x, x' \in P \cup T$ we define the following relations.

- Node $x$ *precedes* $x'$ (denoted $x \preccurlyeq x'$) if there exists a directed path from $x$ to $x'$ . If we require that the path has length greater than zero we write $x \prec x'$.
- Nodes $x$ and $x'$ are in *conflict* (denoted $x \# x'$) if there exist two different transitions $t, t' \in T$ such that: $t \preccurlyeq x$, $t' \preccurlyeq x'$, ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$. In this case, in fact transitions $t$ and $t'$ are in conflict because they have a common input place, and the conflict propagates to all nodes following them. A node $x$ is in *self-conflict* if $x \# x$ holds.
- Nodes $x$ and $x'$ are *concurrent* (denoted $x \approx x'$) if neither $x \preccurlyeq x'$, nor $x' \preccurlyeq x$, nor $x \# x'$ hold.

Note that given two nodes $x$ and $x'$ it may hold that: ($x \prec x'$ and $x' \prec x$ and $x \# x'$).

A *marking* $M : P \to \mathbb{N}$ that assigns to each place of a $P/T$ net a non–negative integer number of tokens, represented by black dots. A $P/T$ *system* or *net system* $\langle N, M_0 \rangle$ is a net $N$ with an initial marking $M_0$.

A transition $t$ is *enabled* at $M$ iff $M(p) > 0$ for all $p \in {}^\bullet t$. If $t$ is enabled, it may *fire* yielding the marking $M' = M + C(\cdot, t)$. We write $M \ |\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at $M$, and we write $M \ |\sigma\rangle \ M'$ to denote that the firing of $\sigma$ yields $M'$. We can associate to a sequence $\sigma$ a *firing vector* $X : T \to \mathbb{N}$ such that $X(t) = k$ if the transition $t$ is contained $k$ times in $\sigma$.

A marking $M$ is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence $\sigma$ such that $M_0 \ |\sigma\rangle \ M$. The set of all markings reachable from $M_0$ defines the *reachability set* of $\langle N, M_0 \rangle$ and is denoted $R(N, M_0)$.

The *incidence matrix* of a net is an $m \times n$ matrix $C$ where; $C(p, t) = 1$ if $(t, p) \in F$ and $(p, t) \notin F$, $C(p, t) = -1$ if $(p, t) \in F$ and $(t, p) \notin F$, else $C(p, t) = 0$.

We denote $PR(N, M_0)$ the *potentially reachable set*, i.e., the set of all markings $M \in \mathbb{N}^m$ for which there exists a vector $X \in \mathbb{N}^n$ that satisfies the *state equation* $M = M_0 + C \cdot X$, i.e., $PR(N, M_0) \triangleq \{M \in \mathbb{N}^m \mid \exists X \in \mathbb{N}^n : M = M_0 + C \cdot X\}$. It holds that $R(N, M_0) \subseteq PR(N, M_0)$.

A place $p$ is *k-bounded* if for all $M \in R(N, M_0)$ it holds $M(p) \leqslant k$. A place 1-*bounded* is called *safe*. A net system $\langle N, M_0 \rangle$ is said $k$-bounded (resp., safe) if all its places are $k$-bounded (resp., safe). A marking $M$ of a safe net system is a binary vector and can also be seen as a set of places $M = \{p \in P \,|M(p) = 1\}$.

In the rest of the paper for sake of simplicity we will consider only safe net systems but the results presented in this paper can easily be extended to arbitrary bounded nets.

## 3. UNFOLDING

In this section we informally recall how it is possible, given a safe net system $\langle N, M_0 \rangle$, to *unfold*

it constructing a *labelled occurrence net* $\tilde{N}(M_0)$. This occurrence net, that is also commonly called the *unfolding of* $\langle N, M_0\rangle$, has a structure that depends both on $N$ and on $M_0$. A formal description of the unfolding procedure requires a long and tedious series of definitions: we prefer to present the key concepts here. Any of the references (McMillan, 1995; Esparza *et al.*, 2002; He and Lemmon, 2002; Benveniste *et al.*, 2003a) contains a more comprehensive and accurate discussion.

An *occurrence net* is an ordinary P/T net with a special structure: (a) starting from any node, all backward paths are finite, i.e., eventually they reach a source node; (b) each place has at most one input arc; (c) no node is in self-conflict. It is easy to show that in an occurrence net if $x$ and $x'$ are two distinct nodes, one and only one of the following conditions holds: $x \prec x'$, or $x' \prec x$, or $x \# x'$, or $x \approx x'$.

To the unfolding $\tilde{N}(M_0) = (\tilde{P}, \tilde{T}, \tilde{F})$ a *labelling function* $\ell : (\tilde{P} \to P) \cup (\tilde{T} \to T)$ is also associated: it maps each node of the unfolding into a node of the original net $N$. Note that usually a node $p$ or $t$ of $N$ may correspond to more than one node of the unfolding, i.e., $\ell^{-1}(p) \subset \tilde{P}$ and $\ell^{-1}(t) \subset \tilde{T}$.

The labelling function can also map set of nodes into set of nodes. In particular, in the following procedure given a set of places $P' \subseteq P$ of the original net, we write $P' = \hat{\ell}(\tilde{P}')$ to denote that the set of places $\tilde{P}'$ of the unfolding has the same cardinality of $P'$ and $P' = \left\{ p \in P \,\middle|\, \tilde{p} \in \tilde{P}', p = \ell(\tilde{p}) \right\}$, hence each place of $\tilde{P}'$ maps into a place of $P'$ but no two places in $\tilde{P}'$ map into the same place of $P'$.

*Procedure 1.* **(Unfolding of a safe net system $\langle N, M_0\rangle$ into an occurrence net $\tilde{N}(M_0)$)**

(1) Add to the unfolding a set of source places $\tilde{P}_0$ with $\hat{\ell}(\tilde{P}_0) = \{p \in P \,|\, M_0(p) = 1\}$.
(2) Let $i := 0$.
(3) Let $\tilde{P}_{\text{exp}} := \tilde{P}_i$
(4) If $\tilde{P}_i = \emptyset$ then STOP.
(5) Let $i := i + 1$.
(6) Let $\tilde{P}_i := \emptyset$.
(7) For all transitions $t \in T$
  For all sets of places $\tilde{P}' \subseteq \left(\tilde{P}_{\text{exp}} \setminus \tilde{P}_i\right)$ such that all following 3 conditions are verified:
    - $\hat{\ell}(\tilde{P}') = {}^\bullet t$,
    - all places in $\tilde{P}'$ are concurrent,
    - $\tilde{P}' \cap \tilde{P}_{i-1} \neq \emptyset$,
  (a) Add to the unfolding a new transition $\tilde{t}$ with $\hat{\ell}(\tilde{t}) = t$.
  (b) Add to the unfolding a set of new places $\tilde{P}''$ with $\hat{\ell}(\tilde{P}'') = t^\bullet$.
  (c) Add an arc from each place in $\tilde{P}'$ to $\tilde{t}$ and from $\tilde{t}$ to each place in $\tilde{P}''$.
  (d) Let $\tilde{P}_i := \tilde{P}_i \cup \tilde{P}''$.
  (e) Let $\tilde{P}_{\text{exp}} := \tilde{P}_{\text{exp}} \cup \tilde{P}''$.
(8) Goto 4.

In the procedure at step 1 we add to the unfolding a copy of each place of the original net marked by the initial marking: all places in this set $\tilde{P}_0$ are ranged on the tier 0 and represent the source nodes of the occurrence net. The index $i$ initialised at step 2 denotes the tier on which the places of each set $\tilde{P}_i$ are ranged.

The set $\tilde{P}_{\text{exp}}$ initialised at step 3 keeps track of the places that can be used to expand the unfolding. Strictly speaking, in this version of the procedure this set needs not be defined because it always coincides with the set of places in the unfolding. However we will need it when the procedure is modified to construct a finite prefix (that we call order 1 unfolding) as explained in the following.

Each time a new tier is added we check for all transitions $t$ of the original net if there exists in the unfolding a set $\tilde{P}'$ with the following properties:

- it is a copy of the set of input places of $t$, hence a marking that marks in the unfolding all places in $\tilde{P}'$ corresponds to a marking of the original net that enables $t$;
- all places in $\tilde{P}'$ are concurrent, hence they can simultaneously be marked (note that the safeness of the original net ensures that two places on the unfolding with the same label cannot be concurrent);
- at least one place in $\tilde{P}'$ belongs to the lastly added tier, so that the marking of the unfolding that marks all these places has not been considered in the previous steps.

For all such sets $\tilde{P}'$ we add to the net a new copy of transition $t$, a new copy of all its output places $\tilde{P}''$ and the relative arcs.

The procedure given above is not an algorithm because it is not guaranteed to halt in a finite number the steps. In fact the unfolding of a net that admits repetitive sequences is infinite.

Note that we can consider an unfolding both as a net and as a marked net where the initial marking assigns to each source place in $\tilde{P}_0$ a token, so we need not specify its initial marking and simply write $R(\tilde{N}(M_0))$ to denote its reachability set.

The unfolding is a safe net so we can represent a marking with the set of non-empty place: we write $\tilde{M}_0 = \tilde{P}_0$ and in general $\tilde{M} = \left\{ \tilde{p} \in \tilde{P} \,\middle|\, \tilde{M}(\tilde{p}) = 1 \right\}$. It is also possible to apply the mapping $\hat{\ell}$ to markings.

*Definition 2.* To each marking $\tilde{M}$ of the unfolding corresponds a marking of the original net $M = \hat{\ell}(\tilde{M}) \triangleq \left\{ p \in P \,|\, p = \ell(\tilde{p}) \,, \tilde{p} \in \tilde{M} \right\}$. This leads to an equivalence relation among markings in $R(\tilde{N}(M_0))$ and if $\hat{\ell}(\tilde{M}) = \hat{\ell}(\tilde{M}')$ we write $\tilde{M} =_P \tilde{M}'$.

A firing vector $\tilde{X}$ of the unfolding is a binary vector that can also be seen as a set of transitions $\tilde{X} = \left\{ \tilde{t} \in \tilde{T} \,\middle|\, \tilde{X}(\tilde{t}) = 1 \right\}$.

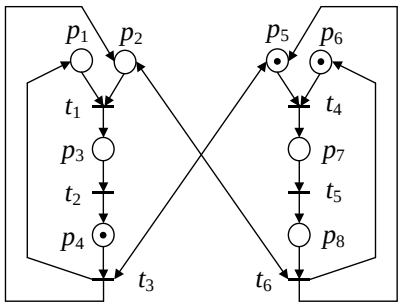


Fig. 1. A safe Petri net.

*Definition 3.* Given a transition $\tilde{t} \in \tilde{T}$, the minimal firing vector of the unfolding that contains it is called a local configuration; it can be show that this vector is unique and we denote it $[\tilde{t}]$. The marking reached firing configuration $\tilde{X}$ (resp., $[\tilde{t}]$) will be denoted $\tilde{M}(\tilde{X})$ (resp., $\tilde{M}([\tilde{t}])$).

It is also clear that each marking $\tilde{M}$ reachable in $\tilde{N}(M_0)$ corresponds to a unique configuration in $\tilde{N}(M_0)$ (the unfolding net is acyclic) that we sometimes denote $\mathrm{Conf}(\tilde{M})$.
Given a net system $\langle N, M_0\rangle$, McMillan (1995) presented a technique to construct a finite prefix of its unfolding. Following Lemmon and He (2002), we consider a slightly different construction of the finite prefix.

*Definition 4.* (Order 1 unfolding). The order 1 unfolding, denoted $\tilde{N}_1(M_0)$, is a finite prefix of the unfolding obtained by Procedure 1 stopping the construction of the unfolding when we reach a *cut-off transition* $\tilde{t}$, i.e., a transition such that:

- EITHER the firing of the local configuration of $\tilde{t}$ brings back to the initial marking, i.e., $\tilde{M}([\tilde{t}]) =_P \tilde{M}_0$;
- OR there exists another transition $\tilde{t}'$ with the following properties:
  (a) $\tilde{t}'$ has a smaller configuration than $\tilde{t}$: $[\tilde{t}'] \subset [\tilde{t}]$;
  (b) the markings reached firing the two configurations are equivalent, i.e., $\tilde{M}([\tilde{t}']) =_P \tilde{M}([\tilde{t}])$.

In the following we call $\tilde{t}'$ the mirror transition of $\tilde{t}$ in $\tilde{N}_1(M_0)$.

It should be noted that what we call order 1 unfolding is a net slightly larger than McMillan finite prefix, because our condition (a) is stronger: McMillan requires only that $\mathrm{card}([\tilde{t}']) < \mathrm{card}([\tilde{t}])$.

*Algorithm 5.* The order 1 unfolding can be constructed using a modified version of Procedure 1 where the instruction 7.(e) is changed to
*7.(e')* If $t$ is not a cut-off transition, then let $\tilde{P}_{\mathrm{exp}} := \tilde{P}_{\mathrm{exp}} \cup \tilde{P}''$.

In this case when a cut-off transitions is added to the unfolding, a copy of its output places is also added but they will not be used to expand the unfolding any further. With this small change, Procedure 1 always stops in a finite number of steps if the original system is safe.

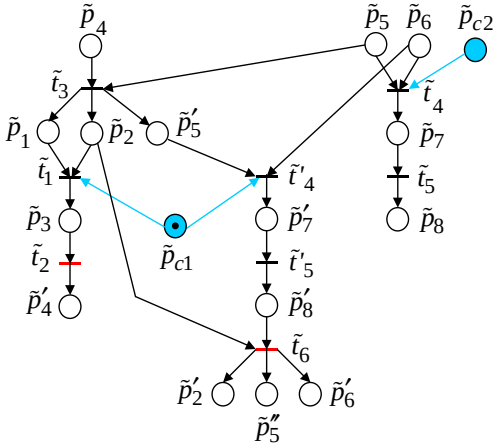


Fig. 2. The order 1 unfolding of the net in Fig. 1 with control places.

*Example 6.* Consider the net shown in Fig. 1. Its order 1 unfolding is shown in Fig. 2 (ignore the cyan filled places denoted $p_{ci}$). Note that we have also added to the unfolding the cut-off transitions (transition $t_2$ on tier 3 and transition $t_6$ on tier 4, marked in red) and their output places. Transition $t_5$ on tier 2 is not not a cut-off transition: after its firing the unfolding cannot proceed because a deadlock is reached. ■

The following result follows from an original result presented by McMillan (1995).

*Lemma 7.* The image through the labelling function of the reachability set of the order 1 unfolding $\tilde{N}_1(M_0)$ is the reachability set of the original system, i.e.,

$$R(N, M_0) = \ell(R(\tilde{N}_1(M_0))) \triangleq \left\{ M \in \mathbb{N}^m \,\middle|\, M = \ell(\tilde{M}) \,, \tilde{M} \in R(\tilde{N}_1(M_0)) \right\}.$$

*Proof.* McMillan showed this result holds for the total unfolding and also for the finite prefix. Since $\tilde{N}_1(M_0)$ is larger than McMillan finite prefix, the result follows immediately. □

## 4. A SPECIAL CLASS OF FORBIDDEN MARKINGS PROBLEM

We consider a control problem where the set of forbidden marking $\mathcal{F}$ has a special structure.

*Definition 8.* A set $\mathcal{F} \subseteq R(N, M_0)$ has property REACH wrt a net system $(N, M_0)$ if

$$M \in \mathcal{F} \text{ and } M' \in R(N, M) \Rightarrow M' \in \mathcal{F}.$$

Thus property REACH implies that the set is closed under the reachability operator.
Meaningful examples of sets that have property REACH are the following: (a) the set of deadlock

markings; (b) the set of markings from which there exists no firing sequence containing a given transition; (c) the set of markings that are not co-reachable, i.e., from which it is not possible to reach a given final marking; (d) the set of markings from which the initial marking is not reachable, i.e., from which no control law can ensure reversibility; (e) the set of markings from which there exists no firing sequence containing all transitions, i.e., from which no control law can ensure liveness.

In the following we focus on the optimal control of $\mathcal{F}$. Property REACH will allow us to use unfolding to design optimal controllers, as we show in the following section.

*Theorem 9.* Given a set $\mathcal{F}$ with property REACH and a marking $\tilde{M}$ such that $\hat{\ell}(\tilde{M}) \in \mathcal{F}$, if $\tilde{M}$ is reachable with configuration $\tilde{X}$, then any larger configuration $\tilde{X}' \geqslant \tilde{X}$ leads to a marking $\tilde{M}'$ such that $\hat{\ell}(\tilde{M}') \in \mathcal{F}$ .

*Proof.* If $\tilde{M}$ is reachable with configuration $\tilde{X}$, and $\tilde{M}'$ is reachable with configuration $\tilde{X}'$ then:

$$\tilde{M} = \tilde{M}_0 + \tilde{C}\tilde{X}, \quad and \quad \tilde{M}' = \tilde{M}_0 + \tilde{C}\tilde{X}'.$$

This implies $\tilde{M}' = \tilde{M} + \tilde{C}(\tilde{X}' - \tilde{X})$ with $\tilde{X}' - \tilde{X} \in \mathbb{N}^{\tilde{n}}$ hence $\tilde{M}'$ is reachable by $\tilde{M}$ (the unfolding is an acyclic net). Thus $\hat{\ell}(\tilde{M}')$ is reachable from $\hat{\ell}(\tilde{M})$ and (by REACH) it belongs to $\mathcal{F}$. □

Based on this property, given a forbidden marking set $\mathcal{F}$ with property REACH we now present a maximally permissive control policy ensuring that no marking in $\mathcal{F}$ is reached. This control policy will be "implemented" in the unfolding net by places with output arcs and no input arcs.

## 5. CONTROL POLICY FOR $\mathcal{F}$

For marking $\tilde{M} \in \mathrm{R}(\tilde{N}(M_0))$ such that $\hat{\ell}(\tilde{M}) \in \mathcal{F}$ let $\tilde{X}$ be the unique configuration that yields it.

*Definition 10.* The set of control transitions of $\tilde{M}$ is $\tilde{X}_c = \left\{ \tilde{t} \in \tilde{X} \,\middle|\, \nexists \tilde{t}' \in \tilde{X}, \tilde{t} \in [\tilde{t}'] \right\}$.

In plain words, these are all transitions inputting into the places that belong to $\tilde{M}$ and that do not precede any other such transition. It is easy to prove that all these transitions are concurrent. In fact, since $\tilde{X}_c \subseteq \tilde{X}$ and $\tilde{X}$ is a firable sequence, no two transitions can be in conflict. Furthermore, all transitions preceding another one in the set $\tilde{X}$ are removed by construction, thus we are left with only concurrent transitions in $\tilde{X}_c$.

We will use the following control structure to prevent reaching $\tilde{M}$.

*Definition 11.* Given a marking $\tilde{M}$ with set of control transitions $\tilde{X}_c$, the control place $\tilde{p}_c$ for $\tilde{M}$ is a new place initially marked with $\left|\tilde{X}_c\right| - 1$ tokens and with an arc going to each transition in $\tilde{X}_c$. The incidence matrix of the control place is $\tilde{C}(\tilde{p}_c, \tilde{t}) = -1$ if $\tilde{t} \in \tilde{X}_c$, else $\tilde{C}(\tilde{p}_c, \tilde{t}) = 0$.

*Theorem 12.* The control strategy corresponding to control places for all $\hat{\ell}(\tilde{M}) \in \mathcal{F}$ is maximally permissive, i.e., it does not prevents the unfolding to reach a marking $\tilde{M}'$ with $\hat{\ell}(\tilde{M}') \notin \mathcal{F}$, if the set $\mathcal{F}$ has property reach.

*Proof.* By construction, each control place for a marking $\hat{\ell}(\tilde{M}) \in \mathcal{F}$ corresponding to configuration $\tilde{X}$ forbids the configuration $\tilde{X}$ and all larger configurations $\tilde{X}' \geqslant \tilde{X}$. From Theorem 9, the control place only prevents from reaching markings $\hat{\ell}(\tilde{M}) \in \mathcal{F}$. As a result, the control strategy corresponding to control places for all $\hat{\ell}(\tilde{M}) \in \mathcal{F}$ is maximally permissive. □

We now show that such a controller can be constructed from order 1 unfolding $\tilde{N}_1(M_0)$ including all its cut-off transitions and their output places.

We first construct the control places that prevent reaching markings in $\mathcal{F}$ in $\tilde{N}_1(M_0)$.

*Algorithm 13.* **Control places for $\mathcal{F}$**

(1) Determine a reachable marking $\tilde{M}$ such that $\hat{\ell}(\tilde{M}) \in \mathcal{F}$ and such that no marking $\tilde{M}'$ with $\hat{\ell}(\tilde{M}') \in \mathcal{F}$ and $\mathrm{conf}(\tilde{M}') \subset \mathrm{conf}(\tilde{M})$ exists.
(2) If no such marking exists, then stop.
(3) Add to $\tilde{N}_1(M_0)$ the control place for $\tilde{M}$.
(4) Goto 1.

The net obtained by adding these control places to the order 1 unfolding is called $\tilde{N}_{1,c}(M_0)$. This net is not necessarily an occurrence net because the control places may contain more than one token.

*Example 14.* Given the net in Fig. 1, assume we want to forbid the set of markings $M \in R(N, M_0)$ such that $M(p_3) + M(p_4) + M(p_7) + M(p_8) = 2$. Clearly $\mathcal{F} = \{\{p_3, p_7\}, \{p_4, p_7\}, \{p_3, p_8\}, \{p_4, p_8\}\}$, and it is not difficult to show that this forbidden set has property REACH for this net.

In the following table for each of the forbidden markings $M$ we have shown the corresponding unfolding marking(s) $\tilde{M}$, the corresponding set of control transitions $\tilde{X}_c$ and finally the control place $\tilde{p}_c$. A symbol * (resp., **) in the last column denotes a configuration already forbidden by place $\tilde{p}_{c1}$ (resp., $\tilde{p}_{c2}$) hence no new place has to be added to the net for preventing it.

| $M$ | $\tilde{M}$ | $\tilde{X}_c$ | $\tilde{p}_c$ |
|---|---|---|---|
| $\{p_3, p_7\}$ | $\{\tilde{p}_3, \tilde{p}'_7\}$ | $\{\tilde{t}_1, \tilde{t}'_4\}$ | $\tilde{p}_{c1}$ |
| $\{p_4, p_7\}$ | $\{\tilde{p}_4, \tilde{p}_7\}$ | $\{\tilde{t}_4\}$ | $\tilde{p}_{c2}$ |
| | $\{\tilde{p}'_4, \tilde{p}'_7\}$ | $\{\tilde{t}_2, \tilde{t}'_4\}$ | * |
| $\{p_3, p_8\}$ | $\{\tilde{p}_3, \tilde{p}'_8\}$ | $\{\tilde{t}_1, \tilde{t}'_5\}$ | * |
| $\{p_4, p_8\}$ | $\{\tilde{p}_4, \tilde{p}_8\}$ | $\{\tilde{t}_5\}$ | ** |
| | $\{\tilde{p}'_4, \tilde{p}'_8\}$ | $\{\tilde{t}_2, \tilde{t}'_5\}$ | * |

Note that place $\tilde{p}_{c2}$ contains no token because its corresponding set of control transitions is a singleton: this means that transition $\tilde{t}_4$ on tier 1 should never fire. ■

*Definition 15.* Let $\tilde{M}$ be a marking of an unfolding net such that $\text{conf}(\tilde{M}) = [\tilde{t}] \cup E$ and $[\tilde{t}] \cap E = \Phi$, i.e. $\text{conf}(\tilde{M})$ is an extension of $[\tilde{t}]$ denoted as $[\tilde{t}] \oplus E$ (Ezparza *et al.*, 2002). If $\tilde{t}$ is a cut-off transition with $\tilde{t}'$ as its mirror transition in $\tilde{N}_1(M_0)$, we define the mirror marking of $\tilde{M}$ as the marking of the configuration $[\tilde{t}'] \oplus E'$ where $E'$ is the equivalent extension of $E$ for $[\tilde{t}']$.

The concept of mirror marking can also be extended to a control places.

*Definition 16.* Let $\tilde{M}$ be a marking of $\tilde{N}_{1,c}(M_0)$ such that $\text{conf}(\tilde{M}) = [\tilde{t}] \cup E$ and $[\tilde{t}] \cap E = \Phi$. If $\tilde{t}$ is a cut-off transition with $\tilde{t}'$ as its mirror transition, the mirror marking of a control place $p_c$ is $\tilde{M}(\tilde{p}_c) - \tilde{C}(\tilde{p}_c, \cdot)([\tilde{t}] - [\tilde{t}'])$.

According to the previous definition, after a cut-off transition fires, the control places should get back all those tokens that have been taken by the firing of $[\tilde{t}] - [\tilde{t}']$.

*Algorithm 17.* The control policy for $\mathcal{F}$ uses the net $\tilde{N}_{1,c}(M_0)$ and can be defined as follows.

(1) The plant and the net $\tilde{N}_{1,c}(M_0))$ are initialised with the respective initial marking.
(2) Compute a control pattern as follows: if $\tilde{T}_e$ is the set of transitions enabled in $\tilde{N}_{1,c}(M_0)$, the set of transitions that are enabled by the controller on the plant is $T_e = \ell(\tilde{T}_e)$.
(3) If a transition t fires in the plant, the unique transition $\tilde{t} \in \ell^{-1}(t)$ enabled in $\tilde{N}_{1,c}(M_0)$ is fired. After the firing of $\tilde{t}$, the marking of the unfolding is set to the related mirror marking if $\tilde{t}$ is a cut-off transition.
(4) Goto 2.

*Theorem 18.* The control policy of Algorithm 17 is maximally permissive.

*Proof.* Similar to the proof of Theorem 12, a marking $\tilde{M}$ of $\tilde{N}_1(M_0)$ is forbidden by control places of Algorithm 13 if and only if $\hat{\ell}(\tilde{M}) \in \mathcal{F}$. Since a marking $\tilde{M}$ obtained by a cut-off transition $\tilde{t}$ is replaced by its mirror marking $\tilde{M}'$, we need to prove that $\tilde{M}'$ is also permitted by control places. This is true since $\tilde{M}$ is accepted by control places which implies $\hat{\ell}(\tilde{M}) \notin \mathcal{F}$ and $\tilde{M} =_P \tilde{M}'$. □

## 6. CONCLUSIONS

In this paper we have used the technique of unfolding to design maximally permissive supervisors for safe Petri nets assuming that the specification is given by a set of forbidden markings with property REACH.

There are some lines for future research that are still open. In many cases it may be possible to find equivalent control structure to be added to the original net rather than to the unfolding. The approach may also be extended to nets with uncontrollable transitions.

## 7. REFERENCES


(Aghasaryan *et al.*, 1988) A. Aghasaryan, E. Fabre, A. Benveniste, R. Boubour, C. Jard, "Fault detection and diagnosis in distributed systems : an approach by partially stochastic Petri nets," *Discrete Events Dynamical Systems*, Vol. 8, p. 203-231, June 1998.

(Benveniste *et al.*, 2003a) A. Benveniste, E. Fabre, S. Haar, C. Jard, "Diagnosis of asynchronous discrete event systems, a net unfolding approach," *IEEE Trans. on Automatic Control*, Vol. 48, No. 5, pp. 714-727, May 2003.

(Benveniste *et al.*, 2003b) A. Benveniste, E. Fabre, S. Haar, "Markov nets: probabilistic models for distributed and concurrent systems," *IEEE Trans. on Automatic Control*, Vol. 48, No. 11, pp. 1936-1950, November 2003.

(Esparza *et al.*, 2002) J. Esparza, S. Römer, W. Vogler, "An improvement of McMillan's unfolding algorithm," *Formal Methods in System Design*, Vol. 20, pp. 285–310, 2002.

(Godefroid, 1996) P. Godefroid, *Partial-order methods for the verification of concurrent systems - an approach to the state-explosion problem*, Lecture Notes in Computer Science, Vol. 1032, Springer-Verlag, 1996.

(He and Lemmon, 2000a) K.X He, M.D. Lemmon, "Liveness verification of discrete-event systems modeled by n-safe ordinary Petri Nets," *Proc. 21$^{st}$ Int. Conf. on Application and Theory of Petri Nets* (Aarhus Denmark), Lecture Notes in Computer Science, Vol. 1825, pp. 227-243, Springer, 2000.

(He and Lemmon, 2000b) K.X He, M.D. Lemmon, "Liveness enforcing supervision of discrete-event systems modeled by $n$-safe Petri nets," *Proc. IFAC Int. Conf. on Control System Design* (Bratislava Slovakia), June 2000.

(He and Lemmon, 2002) K.X. He, M.D. Lemmon, "Liveness-enforcing supervision of bounded ordinary Petri nets using partial order methods," *IEEE Trans. on Automatic Control*, Vol. 47, No. 7, pp. 1042-1055, July 2002.

(Hellgren *et al.*, 1999) A. Hellgren, M. Fabian, B. Lennartson, "Deadlock detection and controller synthesis for production systems using partial order techniques," *Proc. of the 1999 IEEE Int. Conf. on Control Applications* (Kohala Coast, Hawaii, USA), Aug 1999.

(McMillan, 1995) K.L. McMillan, "A technique of state space search based on unfolding," *Formal Methods in System Design*, Vol. 6, No. 1, pp. 45-65, 1995.

(Murata, 1989) T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541–580, April 1989.

(Neumair, 2002) C. Neumair, "Finite unfoldings of unbounded Petri nets," *Petri Net Newsletter* No. 63, pp. 5-10, October 2002.

(Ramadge and Wonham, 1989) P. Ramadge, W.M. Wonham, "Control of discrete event system," *Proceedings of the IEEE*, Vol. 77, No. 1, pp. 81-98, January 1989.

(Valmari, 1991) A. Valmari, "Stubborn sets for reduced state space generation," *Advances in Petri Nets '90*, Lecture Notes in Computer Science, Vol. 483, pp. 491-515, Springer, 1991.

(Valmari, 1994) A. Valmari, "State of the art report: stubborn sets," *Petri Net Newsletter* No. 46, pp. 6-14, April 1994.

(Xie and Giua, 2004) X. Xie, A. Giua, "Counterexamples to «Liveness-enforcing supervision of bounded ordinary Petri nets using partial order methods»," *IEEE Trans. on Automatic Control*, Vol. 49, 2004. To appear.