Sequential Synthesis of Supervisory Policies for Discrete-Event Systems Modeled by Petri Nets

A. Raman and R. S. Sreenivas

***Abstract*— It is often of interest to synthesize a supervisory policy for enforcing complex properties on the behaviour of a Discrete-Event System (DES). One way of doing this is by decomposing complex properties into simpler objectives and then synthesizing supervisors for those simpler objectives in a sequential manner. This approach is particularly convenient if the supervised-system can be represented using the *same* modeling framework at each stage of this sequential process. An additional desirable feature could be that the supervisory policy remain the same even if the initial-state of the DES were to change.**

In this paper, we consider *Petri Net* (PN) models of *Discrete-Event Systems* (DES) under a *supervisory policy* that enforces a desired-property $\mathcal{B}$. We prove that the supervised-system can be modeled as a PN if and only if the supervisory policy is a *marking-monotone-$\mathcal{B}$-enforcing supervisory policy* (MM-$\mathcal{B}$ESP) over reachable markings. In the second half of the paper we describe a software tool for the synthesis of MM-$\mathcal{B}$ESPs, where the desired-property $\mathcal{B}$ is the PN-property of *liveness*, for arbitrary Petri Nets. We end the paper with an example that illustrates both the contributions.

I. Introduction

A *Discrete Event System* (DES) is a discrete-state system, where the discrete-state changes at discrete-time instants due to the occurrence of events. Suppose we have a directed graph representing a DES in which each node represents a state of the system, and each edge represents an event that takes the system from one state to another. Suppose some of the events in the system are controllable, in the sense that they can be prevented from occurring by a *supervisor*. The supervisory policy specifies which events (edges) to disable (resp. remove) at which states (resp. nodes) such that the DES (residual graph) satisfies a desired property $\mathcal{B}$. The simplest way of doing this is to use a procedure that tries all possible combinations of edges that can be removed. However, this approach is inapplicable when the graph is infinite in size. Even if we have a finite-model of the infinite-state DES, there is the additional requirement for the existence of a procedure that synthesizes the supervisory policy for this DES. Although well established methods exist for synthesizing policies that enforce certain properties (like liveness, safety, boundedness etc.), it is likely that there do not exist systematic procedures for the synthesis of policies that enforce complex objectives (like the combination of several objectives). One of the ways of accomplishing this is by decomposing the complex objective into simpler objectives and then synthesizing policies for enforcing these simpler objectives (cf. Figure 1).

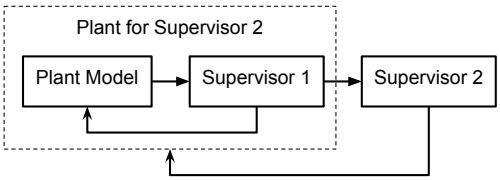

Fig. 1. Sequential Synthesis of Supervisory Policies.

The issue with this approach is that while there was a model, within a modeling-paradigm, of the original infinite-state DES, there may not exist a model of the supervised system within the same modeling-paradigm. For instance, Example 3.1 of [1] presents a plant DES that is a *Petri Net* (PN), where the desired behavior $\mathcal{B}$ corresponds to the requirement to be *non-blocking*. Giua and DiCesare have shown that there is no PN that can model the resulting supervised-system. In the context of Figure 1 of this paper, this result shows that while the plant for Supervisor 1 is a PN, the plant for Supervisor 2 cannot be PN in any sequential attempt to the synthesis of supervisory policies. The identification of a necessary and sufficient condition for a DES modeling paradigm (different from *Finite State Automata*) where the supervised-DES can also be represented using the same modeling paradigm, plays a critical role in the sequential synthesis of supervisory policies for DES. Thus, there are two aspects to this problem– (i) the algebraic framework under which the supervisor construction is carried out; and, (ii) the model. We discuss each of these points in the remainder of the section.

Modular problem specification and supervisor construction for DES was first discussed by Ramadge and Wonham in [2], [3]. They modeled the behavior of DES as a prefix closed language L over the set of event alphabet Σ, where each $u \in L$ is a possible event sample path. The behaviour of the language L is modeled by a *generator* G which is an automaton (Σ, Q, δ, q_0). Here Q represents the set of (possibly infinite) states, $\delta : \Sigma \times Q \rightarrow Q$ is the transition function and q_0 is the initial state. Control is modeled by partitioning the event space into controllable and uncontrollable events: $\Sigma = \Sigma_u \cup \Sigma_c$. For multiple objective controller synthesis, if each objective is specified in terms of a controlled language K_i, then the overall desired behaviour

This work was supported in part by the Arthur Davis Faculty Scholar Endowment at the University of Illinois at Urbana-Champaign.

The authors are with the Department of Industrial and Enterprise Systems Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Illinois, USA. raman12{rsree}@illinois.edu

is specified by the controlled language $\cap_i K_i$. There are two key points– controllability of the desired language and compatibility between multiple objectives. For prefix closed languages, if K_1 and K_2 are controllable, then $K_1 \cap K_2$ is also controllable (see Section IV in [3] for the definition of controllability). Compatibility between multiple objectives is formalized by the concept of nonconflicting languages. Two controlled languages $K_1, K_2 \in \Sigma^*$, are said to *nonconflicting* if $pr(K_1 \cap K_2) = pr(K_1) \cap pr(K_2)$ where $pr(\bullet)$ denotes the prefix of the string argument. That is, whenever the two languages K_1 and K_2 share a prefix, they also share a word containing this prefix.

Modeling of a supervised Petri net by another Petri net was first discussed in [4], where the authors gave an algorithm for constructing the PN model of the supervised system. They proved that such a construction can always be carried out for conservative PNs (cf. Theorem 4.1 in [4]). Reference [1] proved by a counter-example that not all supervised Petri nets can be modeled by a PN. Using the work in the aforementioned papers as a starting point, in this paper we first formally define the concept of *composition* of a PN model and a supervisory policy. We extend the algorithm in [4] to unbounded PNs, and prove that a composition of a PN model and a supervisory policy $\mathcal{P}$ that enforces a property $\mathcal{B}$ exists if and only if the supervisory policy is marking-monotone $\mathcal{B}$-enforcing supervisory policy (MM-$\mathcal{B}$ESP) over the reachable markings. That is, (i) if $\mathcal{P}$ permits a transition to fire at a smaller marking, then it permits the transition at all *reachable* larger markings as well, and (ii) $\mathcal{P}$ enforces property $\mathcal{B}$ at all *reachable* larger markings. We also generalize the concept of marking-monotonicity from the set of reachable markings to all markings in the marking space and prove the necessary and sufficient conditions for the existence of an MM-$\mathcal{B}$ESP. By definition, if $\mathcal{P}$ is an MM-$\mathcal{B}$ESP for an initial marking, then it is an MM-$\mathcal{B}$ESP for all larger initial markings as well. Since the policy does not change for a larger initial marking, the composition of the PN and the policy also stays the same for a larger initial marking. This is an important property to have while designing a system as analysis for various initial markings become easy.

A PN is *live* if it is possible to fire any transition, although not immediately, from any marking that is reachable from the initial marking. Let $\Delta(N)$ denote the set of initial markings for which a *Liveness Enforcing Supervisory Policy* ($\mathcal{L}$ESP) exists. The modular enforcement of liveness for PNs is complicated by the fact that the existence of an $\mathcal{L}$ESP heavily depends on the set of markings reachable under the supervision of an $\mathcal{L}$ESP (cf. Theorem 5.1 in [5]). A PN that is live can lose liveness if some markings that were originally reachable cannot be reached anymore (due to the action of some other supervisory policy)— which is something that is like to happen in a scheme like in Figure 1 where Supervisor 2 further trims the markings that were reachable under the supervision of Supervisor 1. This behaviour is different compared to other well studied properties like safety, boundedness, reachability of a marking etc. For instance, if a PN is bounded under the supervision of a policy, then trimming the set of reachable markings will not result in loss of boundedness. As the last contribution of the paper, we combine marking monotonicity with liveness and describe a software tool for the synthesis of marking-monotone $\mathcal{L}$ESPs for arbitrary PNs.

The paper is organized as follows. In Section II we formally discuss some preliminaries of PNs and introduce notations and definitions that we will use in the rest of the paper. Section III presents the main results of this paper. We describe the software tool for the synthesis of an MM-$\mathcal{L}$ESP in Section IV. Section V presents an example to illustrate the results. We conclude the paper with Section VI.

II. Notations

A *Petri net structure* $N = (\Pi, T, \Phi, \Gamma)$ is an ordered 4-tuple, where $\Pi = \{p_1, \ldots, p_n\}$ is a set of n *places*, $T = \{t_1, \ldots, t_m\}$ is a collection of m *transitions*, $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, and $\Gamma : \Phi \to \mathcal{N}^+$ is the *weight* associated with each arc. The *initial marking function* (or the *initial marking*) of a PN structure N is a function $\mathbf{m}^0 : \Pi \to \mathcal{N}^n$, which identifies the number of *tokens* in each place, where $\mathcal{N}$ ($\mathcal{N}^+$) denotes the set of non-negative (positive) integers. The marking can be interpreted as an integer-valued vector where the i-th component represents the token load of the i-th place $p_i \in \Pi$. We use the notation $\mathbf{m}$(p) to denote the tokens in place $p \in \Pi$ at marking $\mathbf{m}$.

We use $N(\mathbf{m}^0)$ to denote a PN structure N along with its initial marking $\mathbf{m}^0$. In graphical representations of PNs, the places, tokens, transition and arcs are represented by circles, filled circles, rectangles and directed edges respectively. For brevity, we only show non-unitary arc-weights alongside arcs in graphic representations of PNs in this paper. The set of transitions in the PN is partitioned into controllable- ($T_c \subseteq T$) and uncontrollable-transitions ($T_u \subseteq T$). The controllable (uncontrollable) transitions are represented as filled (resp. unfilled) rectangles.

We define the sets ${}^\bullet x = \{y | (y, x) \in \Phi\}$ and $x^\bullet = \{y | (x, y) \in \Phi\}$. A transition $t \in T$ is said to be *state-enabled* at a marking $\mathbf{m}^i$ if $\forall p \in {}^\bullet t, \mathbf{m}^i(p) \geq \Gamma(p, t)$. The set of state-enabled transitions at marking $\mathbf{m}^i$ is denoted by the symbol $T_e(N, \mathbf{m}^i)$. If $t_j \in T_e(N, \mathbf{m})$, then $\mathbf{m} \geq \mathbf{IN}_{\bullet, j}$, which is the j-th column of the $n \times m$ *input matrix* $\mathbf{IN}$. $\mathbf{IN}_{i,j} = \Gamma(p, t)$ if $p_i \in {}^\bullet t_j$, else it is zero. Similarly, the *output matrix* is an $n \times m$ matrix that encodes the firing of an enabled transition. $\mathbf{OUT}_{i,j} = \Gamma(t, p)$ if $p_i \in t_j^\bullet$, otherwise it is zero. The *incidence matrix* $\mathbf{C}$ of the PN N is an $n \times m$ matrix, where $\mathbf{C} = \mathbf{OUT} - \mathbf{IN}$.

A supervisory policy $\mathcal{P} : \mathcal{N}^n \times T \to \{0, 1\}$, is a function that returns a 0 or 1 for each marking and each transition. If $\mathcal{P}(\mathbf{m}^i, t_j) = 1$, we say the transition t_j is *control-enabled* (permitted to fire) at marking $\mathbf{m}^i$. A transition has to be state- and control-enabled before it can fire. We assume that $\forall t_j \in T_u \ \forall \mathbf{m}^i \in \mathcal{N}^n, \mathcal{P}(\mathbf{m}^i, t_j) = 1$. A state- and control-enabled transition can fire, which changes the marking $\mathbf{m}^i$ to $\mathbf{m}^{i+1}$ according to $\mathbf{m}^{i+1}(p) = \mathbf{m}^i(p) - \Gamma(p, t) + \Gamma(t, p)$.

A string of transitions $\sigma = t_1 \ldots t_k$, where $t_j \in T$ ($j \in \{1, \ldots, k\}$), is said to be a *valid firing string* from

$\mathbf{m}^i$ if 1) the transitions $t_1 \in T_e(N, \mathbf{m}^i)$, $\mathcal{P}(\mathbf{m}^i, t_1) = 1$, and 2) for $j \in \{1, 2, \ldots, k-1\}$, the firing of the transition t_j produces a marking $\mathbf{m}^{i+j}$ and $t_{j+1} \in T_e(N, \mathbf{m}^{i+j})$ and $\mathcal{P}(\mathbf{m}^{i+j}, t_{j+1}) = 1$. If $\mathbf{m}^{i+k}$ results from the firing of $\sigma \in T^*$ starting from the initial marking $\mathbf{m}^i$, we represent it symbolically as $\mathbf{m}^i \xrightarrow{\sigma} \mathbf{m}^{i+k}$. The symbol T^* denotes the set of all possible strings that can be constructed from an alphabet T.

Given an initial marking $\mathbf{m}^0$, the set of *reachable markings* for $\mathbf{m}^0$, which is denoted by $\mathfrak{R}(N,\ \mathbf{m}^0)$, is defined as the set of markings generated by all valid firing strings starting with marking $\mathbf{m}^0$ in the PN N. The set of reachable markings under the supervision of $\mathcal{P}$ in N from the initial marking $\mathbf{m}^0$ is denoted by $\mathfrak{R}(N,\ \mathbf{m}^0,\ \mathcal{P})$.

We use $\mathcal{B}$ESP as a shorthand to denote a property $\mathcal{B}$ enforcing supervisory policy. The set of initial markings for which property $\mathcal{B}$ can be enforced is defined as: $\mathfrak{D}(\mathcal{B}, N) = \{\mathbf{m}^0 \in \mathcal{N}^n : \exists$ a $\mathcal{B}$ESP for $N(\mathbf{m}^0)\}$. This can also be restated as: ($\exists$ a $\mathcal{B}$ESP for $N(\mathbf{m}^0)) \Leftrightarrow (\mathbf{m}^0 \in \mathfrak{D}(\mathcal{B}, N))$. A supervisory policy $\mathcal{P}$ is a *marking-monotone policy*, if $\forall \mathbf{m}^r \geq \mathbf{m}^s$, $\forall t \in T$, $(\mathcal{P}(\mathbf{m}^s, t) = 1) \Rightarrow (\mathcal{P}(\mathbf{m}^r, t) = 1)$. That is, if the policy permits a transition to fire at a smaller marking, then it will permit it at all larger markings as well. If a marking-monotone policy that is a $\mathcal{B}$ESP for $N(\mathbf{m}^s)$ is also a $\mathcal{B}$ESP for $N(\mathbf{m}^r)$ for all $\mathbf{m}^r \geq \mathbf{m}^s$, then it is called a *marking-monotone $\mathcal{B}$ESP* (MM-$\mathcal{B}$ESP). The set $\mathfrak{D}_M(\mathcal{B}, N)$ denotes the set of initial markings for which an MM-$\mathcal{B}$ESP exists. It follows that $\mathfrak{D}_M(\mathcal{B}, N) \subseteq \mathfrak{D}(\mathcal{B}, N)$. A set of markings $\mathcal{S} \subseteq \mathcal{N}^n$ is said to be *right-closed* if $((\mathbf{m}^1 \in \mathcal{S}) \wedge (\mathbf{m}^2 \geq \mathbf{m}^1) \Rightarrow (\mathbf{m}^2 \in \mathcal{S}))$. A right-closed set, $\mathcal{S}$, is uniquely identified by its finite set of minimal elements denoted by $\min(\mathcal{S})$. $\mathfrak{D}_M(\mathcal{B}, N)$ is right-closed by definition. For a given initial marking $\mathbf{m}^0$, a supervisory policy $\mathcal{P}$ is an *MM-$\mathcal{B}$ESP over reachable markings*, if $\forall \mathbf{m}^r, \mathbf{m}^s \in \mathfrak{R}(N, \mathbf{m}^0)$ such that $\mathbf{m}^r \geq \mathbf{m}^s$ (i) $\forall t \in T$, $(\mathcal{P}(\mathbf{m}^s, t) = 1) \Rightarrow (\mathcal{P}(\mathbf{m}^r, t) = 1)$; and, (ii) if $\mathcal{P}$ is MM-$\mathcal{B}$ESP for $N(\mathbf{m}^s)$, then it is an MM-$\mathcal{B}$ESP for $N(\mathbf{m}^r)$ as well. A set of markings, $\mathcal{S}$, is said to be *control invariant* if $\nexists t_u \in T_u$ such that for any $\mathbf{m}^1 \in \mathcal{S}$, $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ and $\mathbf{m}^2 \notin \mathcal{S}$. Sets $\mathfrak{D}(\mathcal{B}, N)$ and $\mathfrak{D}_M(\mathcal{B}, N)$ are control invariant. A supervisory policy $\mathcal{P}$ is said to *enforce a set* $\mathcal{S}$ if (i) $\mathcal{S}$ is control invariant; and, (ii) $\forall \mathbf{m}^0 \in \mathcal{S}$, $\forall t_c \in T_c$, $(\mathcal{P}(\mathbf{m}^0, t_c) = 1) \Leftrightarrow (\mathbf{m}^0 \xrightarrow{t_c} \mathbf{m}^1,\ \mathbf{m}^1 \in \mathcal{S})$.

A PN $N(\mathbf{m}^0)$ is said to be *live* if $\forall t \in T, \forall \mathbf{m}^i \in \mathfrak{R}(N, \mathbf{m}^0), \exists \mathbf{m}^j \in \mathfrak{R}(N, \mathbf{m}^i)$ such that $t \in T_e(N, \mathbf{m}^j)$. (cf. *level 4 liveness*, [6]). A transition t_k is *live* under the supervision of $\mathcal{P}$, if $\forall \mathbf{m}^i \in \mathfrak{R}(N, \mathbf{m}^0, \mathcal{P}), \exists \mathbf{m}^j \in \mathfrak{R}(N, \mathbf{m}^i, \mathcal{P})$ such that $t_k \in T_e(N, \mathbf{m}^j)$ and $\mathcal{P}(\mathbf{m}^j, t_k) = 1$. A policy $\mathcal{P}$ is a *liveness enforcing supervisory policy* ($\mathcal{L}$ESP) for $N(\mathbf{m}^0)$ if all transitions in $N(\mathbf{m}^0)$ are live under $\mathcal{P}$. The test for existence (resp. non-existence) of an $\mathcal{L}$ESP for an initial marking reduces to the decision-problem – "*Is* $\mathbf{m}^0 \in \mathfrak{D}(\mathcal{L}, N)$?" (resp. "*Is* $\mathbf{m}^0 \notin \mathfrak{D}(\mathcal{L}, N)$?"). For arbitrary PNs, the questions "*Is* $\mathbf{m}^0 \in \mathfrak{D}(\mathcal{L}, N)$?" and "*Is* $\mathbf{m}^0 \notin \mathfrak{D}(\mathcal{L}, N)$?" are not semi-decidable [7]. However, recent results have shown that "*Is* $\mathbf{m}^0 \in \mathfrak{D}_M(\mathcal{L}, N)$" is decidable for an arbitrary PN structure N [8].

III. Main Results

Definition 1: Let $\mathcal{P}$ be a $\mathcal{B}$ESP for $N_1(\mathbf{m}^0)$. $N_1(\mathbf{m}^0)$ and $\mathcal{P}$ are said to be *$\mathcal{B}$-composable* if there exists a Petri net N_2 such that $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$. We refer to $N_2(\mathbf{m}^0)$ as the *$\mathcal{B}$-preserving composition* of $N_1(\mathbf{m}^0)$ and $\mathcal{P}$.

Theorem 1: There exists an MM-$\mathcal{B}$ESP for $N(\mathbf{m}^0)$ if and only if there exists a subset $\widehat{\mathfrak{D}}(\mathcal{B}, N) \subseteq \mathfrak{D}(\mathcal{B}, N)$ such that:

1) $\mathbf{m}^0 \in \widehat{\mathfrak{D}}(\mathcal{B}, N)$.
2) $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is right-closed.
3) A supervisory policy that enforces the set $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is a $\mathcal{B}$ESP.

Proof: (If) Suppose $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is right-closed and $\mathbf{m}^0 \in \widetilde{\mathfrak{D}}(\mathcal{B}, N)$. Let $\{\widetilde{\mathbf{m}}^i\}_{i=1}^l$ denote the minimal elements of $\widehat{\mathfrak{D}}(\mathcal{B}, N)$. Consider a supervisory policy that enforces the set $\widehat{\mathfrak{D}}(\mathcal{B}, N)$, that is:

1) $\forall t_u \in T_u$, $(\mathbf{m}^0 \xrightarrow{t_u} \mathbf{m}^1) \Rightarrow (\mathbf{m}^1 \geq \widetilde{\mathbf{m}}^i$ for some $i \in \{1, \ldots, l\})$.
2) $\forall t_c \in T_c$, $(\mathcal{P}(\mathbf{m}^0, t_c) = 1) \Leftrightarrow (\mathbf{m}^0 \xrightarrow{t_c} \mathbf{m}^1$, $\mathbf{m}^1 \geq \widetilde{\mathbf{m}}^i$ for some $i \in \{1, \ldots, l\})$

Let $\widehat{\mathbf{m}}^0 \geq \mathbf{m}^0$ and consider a transition t such that $\mathbf{m}^0 \xrightarrow{t} \mathbf{m}^1$ and $\widehat{\mathbf{m}}^0 \xrightarrow{t} \widehat{\mathbf{m}}^1$. Suppose $\mathcal{P}(\mathbf{m}^0, t) = 1$. Then $(\mathbf{m}^1 \geq \widetilde{\mathbf{m}}^i) \Rightarrow (\widehat{\mathbf{m}}^1 \geq \widetilde{\mathbf{m}}^i)$. Therefore, a supervisory policy that enforces a right-closed set ($\widehat{\mathfrak{D}}(\mathcal{B}, N)$ in this case) will permit the transition t to fire at $\widehat{\mathbf{m}}^0$ as well, and hence is marking-monotone. By Item 3, the supervisory policy that enforces $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is a $\mathcal{B}$ESP. Therefore, a supervisory policy that enforces the set $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is an MM-$\mathcal{B}$ESP.

(Only If) Suppose there exists an MM-$\mathcal{B}$ESP for $N(\mathbf{m}^0)$. Let $\widehat{\mathfrak{D}}(\mathcal{B}, N) = \mathfrak{D}_M(\mathcal{B}, N)$. Then $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is right-closed by definition. Since there is an MM-$\mathcal{B}$ESP for $N(\mathbf{m}^0)$, it follows that $\mathbf{m}^0 \in \widehat{\mathfrak{D}}(\mathcal{B}, N)$. Suppose $\mathbf{m}^1 \in \widehat{\mathfrak{D}}(\mathcal{B}, N)$ and $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^2$ for some $t_u \in T_u$. Then we have that $\mathbf{m}^2 \in \widehat{\mathfrak{D}}(\mathcal{B}, N)$. If not, the supervisory policy will not be an MM-$\mathcal{B}$ESP for $\mathbf{m}^2$ (as $\widehat{\mathfrak{D}}(\mathcal{B}, N) = \mathfrak{D}_M(\mathcal{B}, N)$). In fact, using the same argument, the MM-$\mathcal{B}$ESP will disable any controllable transition whose firing takes the PN outside $\widehat{\mathfrak{D}}(\mathcal{B}, N)$. Therefore, the marking monotone policy that enforces the set $\widehat{\mathfrak{D}}(\mathcal{B}, N)$ is a $\mathcal{B}$ESP. ■

Let $\mathcal{T}_M(\mathcal{B}, N, t) \subseteq \mathfrak{D}_M(\mathcal{B}, N)$ be the set of markings such that $\forall \mathbf{m} \in \mathcal{T}_M(\mathcal{B}, N, t)$, $\mathcal{P}_M(\mathbf{m}, t) = 1$, where $\mathcal{P}_M$ is an MM-$\mathcal{B}$ESP that enforces the set $\mathfrak{D}_M(\mathcal{B}, N)$. Since $\mathcal{P}_M$ is an MM-$\mathcal{B}$ESP, $\mathcal{T}_M(\mathcal{B}, N, t)$ is right-closed.

Lemma 1: Let $\{\widetilde{\mathbf{m}}^i\}_{i=1}^k = \min(\mathfrak{D}_M(\mathcal{B}, N))$. For any $t_c \in T_c$, $\min(\mathcal{T}_M(\mathcal{B}, N, t_c)) = \{\max\{\mathbf{0}, \widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c\}\}_{i=1}^k$. Here $\mathbf{1}_c$ is the unit-vector whose c-th element (corresponding to transition t_c) is unity, the $\max$ operator acts element-wise, and $\mathbf{0}$ represents a vector of all zeros of appropriate size.

Proof: Recall that we need $\mathbf{m}^0 \in \mathcal{D}_M(\mathcal{B}, N)$ for $\mathcal{P}_M$ to enforce property $\mathcal{B}$. First we note that $\forall \widetilde{\mathbf{m}}^i \in \min(\mathfrak{D}_M(\mathcal{B}, N))$, $\max\{\mathbf{0}, \widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c\} \in \mathcal{T}_M(\mathcal{B}, N, t_c)$. Since $(\widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c) + \mathbf{C} \times \mathbf{1}_c = \widetilde{\mathbf{m}}^i$, we have $\max\{\mathbf{0}, \widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c\} + \mathbf{C} \times \mathbf{1}_c \geq \widetilde{\mathbf{m}}^i$. That is, the firing of t_c keeps the marking in the set $\mathfrak{D}_M(\mathcal{B}, N)$. That $\max\{\mathbf{0}, \widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c\}$ is the minimal element of $\mathcal{T}_M(\mathcal{B}, N, t_c)$ follows from the observation that it is the largest non-negative marking greater

than $(\widetilde{\mathbf{m}}^i - \mathbf{C} \times \mathbf{1}_c)$ and that $\widetilde{\mathbf{m}}^i$ is the minimal element of $\mathfrak{D}_M(\mathcal{B}, N)$. ■

Algorithm 1 COMPOSE(N_1, $\mathfrak{D}_M(\mathcal{B}, N_1)$)

1: $\Pi_2 = \Pi_1$
2: $T_2 = T_{1u}$
3: $\Gamma_2(t,p) = \Gamma_1(t,p)$, $\Gamma_2(p,t) = \Gamma_1(p,t)\ \forall t \in T_{1u}$
4: **for** $i \in \{1, \ldots |T_{1c}|\}$ **do**
5: **for** $j \in \{1, \ldots k\}$ **do**
6: $T_2 \leftarrow T_2 \cup \{t_i^j\}$
7: **for** $\forall p \in \Pi_2$ **do**
8: $\Gamma(p, t_i^j) = (\max\{\mathbf{0}, \widetilde{\mathbf{m}}^j - \mathbf{C} \times \mathbf{1}_i\})(p)$
9: $\Gamma(t_i^j, p) = (\max\{\mathbf{0}, \widetilde{\mathbf{m}}^j - \mathbf{C} \times \mathbf{1}_i\} + \mathbf{C} \times \mathbf{1}_i)(p)$
10: **end for**
11: **end for**
12: **end for**

Algorithm 1 presents a procedure for evaluating a $\mathcal{B}$-preserving composition, $N_2 = (\Pi_2, T_2, \Gamma_2)$, of a PN $N_1 = (\Pi_1, T_1, \Gamma_1)$ and an MM-$\mathcal{B}$ESP, $\mathcal{P}_M$, that enforces the set $\mathfrak{D}_M(\mathcal{B}, N_1)$. We need to construct N_2 such that $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P}_M)$. A supervisory policy has no control over the uncontrollable transitions. Therefore, intuitively, the behaviour of the system in the uncontrollable space should be the same for N_1 and N_2 (Steps 2 and 3). Lemma 1 identifies k minimal elements of the set $\mathcal{T}_M(\mathcal{B}, N, t)$ for a controllable transition t_i of N_1. Using this observation, each controllable transition, t_i, of N_1 is replaced by k-many controllable transitions, $\{t_i^j\}_{j=1}^k$ in N_2. The input arc-weights of $\{t_i^j\}_{j=1}^k$ correspond to these minimal elements of $\mathcal{T}_M(\mathcal{B}, N, t_i)$. The output arc-weights of $\{t_i^j\}_{j=1}^k$ correspond to effect of firing t_i. Steps 4 to 12 accomplish these tasks. T_{1c} and T_{1u} denote the set of controllable and uncontrollable transitions in N_1 respectively.

MM-$\mathcal{B}$ESPs are a generalized version of MM-$\mathcal{B}$ESPs-over-reachable-markings. While MM-$\mathcal{B}$ESPs consider the marking monotonicity over all markings, MM-$\mathcal{B}$ESPs over reachability only consider the markings that are reachable from the initial marking (that is, ignoring the markings that are not reachable from the initial marking). It follows that the existence of an MM-$\mathcal{B}$ESP implies the existence of an MM-$\mathcal{B}$ESPs over reachability. Lemma 1 and Algorithm 1 can be easily extended for MM-$\mathcal{B}$ESPs over reachable markings by constraining the analysis to reachability markings. We do not explicate the details in the interest of space.

Theorem 2: For an arbitrary Petri Net N_1: ($\exists$ a $\mathcal{B}$-preserving composition of $N_1(\mathbf{m}^0)$ and $\mathcal{P}$) $\Leftrightarrow$ ($\mathcal{P}$ is an MM-$\mathcal{B}$ESP over reachable markings).

Proof: ($\Rightarrow$) Suppose there exists a $\mathcal{B}$-preserving composition of $N_1(\mathbf{m}^0)$ and $\mathcal{P}$. Then $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$. Since we have to consider the unsupervised reachability graph of N_2, without loss of generality in the context of the proof, we assume all transitions in N_2 are uncontrollable. The condition $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$ implies that transitions $\{t_i^j\} \in T_2$ are state-enabled at a marking if and only if $\mathcal{P}(\mathbf{m}, t_c) = 1$ for some $t_c \in T_{1c}$. To see this, assume $\exists t_i^j \in T_2$ and a marking $\mathbf{m}$ such that $t_i^j \in T_e(N_2, \mathbf{m})$ but $\mathcal{P}(\mathbf{m}, t_i) = 0$, where $t_i \in T_{1c}$. Then $(\mathbf{m} + \mathbf{C} \times \mathbf{1}_i^j) \in \mathfrak{R}(N_2, \mathbf{m}^0) - \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$, which is a contradiction. In the same way if $\exists t_i \in T_{1c}$ and a marking $\mathbf{m}$ such that $\nexists t_i^j \in T_2$ such that $t_i^j \in T_e(N_2, \mathbf{m})$ but $\mathcal{P}(\mathbf{m}, t_c) = 1$, then $(\mathbf{m} + \mathbf{C} \times \mathbf{1}_i) \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P}) - \mathfrak{R}(N_2, \mathbf{m}^0)$, which is again a contradiction.

Therefore, $\forall \mathbf{m}$, $\forall t_u \in T_2$, $(t_u \in T_e(N_2, \mathbf{m})) \Leftrightarrow (\mathcal{P}(\mathbf{m}, t) = 1$ for some $t \in T_1)$. A transition that is enabled at a marking is also enabled at all larger markings. This means that $\mathcal{P}$ is a marking-monotone policy. Since N_2 is a $\mathcal{B}$-preserving composition, it means that $\mathcal{P}$ is an MM-$\mathcal{B}$ESP.

($\Leftarrow$) We prove that the PN N_2 obtained by the construction in Algorithm 1 is a $\mathcal{B}$-preserving composition of N_1 and a $\mathcal{B}$ESP $\mathcal{P}$. We use induction to prove that $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$. The base case is the initial marking $\mathbf{m}^0$. Consider a string σ that is a valid firing string from $\mathbf{m}^0$. Suppose $\mathbf{m}^0 \xrightarrow{\sigma} \mathbf{m}^1$ and $\mathbf{m}^0 \xrightarrow{\sigma_1} \mathbf{m}^2$, where $\sigma_1 \in pr(\sigma)$. Here we use $pr(\bullet)$ to denote the set of prefixes of the string argument. The induction hypothesis is that $\mathbf{m}^2 \in \mathfrak{R}(N_2, \mathbf{m}^0)$ and $\mathbf{m}^2 \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})\ \forall \sigma_1 \in pr(\sigma)$.

Since the input arc-weights of uncontrollable transitions are the same in N_1 and N_2, if $\exists t_u \in T_{1u} \cap T_e(N_1, \mathbf{m}^1)$, then $\exists t_u \in T_2 \cap T_e(N_2, \mathbf{m}^1)$. Since the output arc-weights of uncontrollable transitions are same in N_1 and N_2, if $\mathbf{m}^1 \xrightarrow{t_u} \mathbf{m}^3$, then $\mathbf{m}^3 \in \mathfrak{R}(N_2, \mathbf{m}^0)$ and $\mathbf{m}^3 \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$. Next consider a $t_i \in T_{1c}$. If $\mathcal{P}(\mathbf{m}^1, t_i) = 1$, then from Lemma 1, we have that $\mathbf{m}^1 \geq \widehat{\mathbf{m}}_i$ for some $\widehat{\mathbf{m}}_i \in \min(\mathcal{T}_M(\mathcal{B}, N, t_i))$. Then it follows from the construction in Algorithm 1, that $\exists j$ such that $t_i^j \in T_e(N_2, \mathbf{m}^1)$ (Step 8). Moreover, the firing of t_i^j adds $\mathbf{C}_i(p)$-many tokens in places $p \in \Pi_2$, which is equal to the number of tokens added in $p \in \Pi_1$. Therefore, if $\mathbf{m}^1 \xrightarrow{t_i} \mathbf{m}^4$, then $\mathbf{m}^4 \in \mathfrak{R}(N_2, \mathbf{m}^0)$ and $\mathbf{m}^4 \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$. On the other hand, if $\mathcal{P}(\mathbf{m}^1, t_i) = 0$, then from Lemma 1, we have that $\nexists \widehat{\mathbf{m}}_i \in \min(\mathcal{T}_M(\mathcal{B}, N, t_i))$ such that $\mathbf{m}^1 \geq \widehat{\mathbf{m}}_i$. Then it follows from the construction in Algorithm 1, that $\nexists j$ such that $t_i^j \in T_e(N_2, \mathbf{m}^1)$ (Step 8). This constitutes the induction step. ■

An important consequence of the above theorem is that if there exists an MM-$\mathcal{B}$ESP $\mathcal{P}$ for $N(\mathbf{m}^0)$, then there exists a composition of $\mathcal{P}$ and $N(\mathbf{m}^0)$. Moreover, due to the marking-monotone nature over the whole space of markings, the composition remains the same for any $\widehat{\mathbf{m}}^0 \geq \mathbf{m}^0$. This is a desirable feature in the design of systems as analysis for various initial markings becomes easy, without having to evaluate the composition for each of them separately.

Suppose we want to synthesize a supervisory policy that enforces the property $\wedge_{c=1}^{l} \mathcal{B}_c$ in a Petri net $N_1(\mathbf{m}^0)$. We assume that the existence of an MM-$\mathcal{B}_c$ESP over reachable markings is decidable for all $c \in \{1, \ldots, l\}$, and that there exists a procedure for synthesis. We also assume that the composed model and the supervisory policy that enforces $\wedge_{c=1}^{l} \mathcal{B}_c$ is independent of the order in which $\mathcal{B}_c$s are enforced. Algorithm 2 gives an *outline* of the procedure for the synthesis of a supervisory policy that enforces $(\wedge_{c=1}^{l} \mathcal{B}_c)$. A more specific procedure will depend on the properties that

we want to enforce. Note that in Algorithm 2 we use the set $\widehat{\gamma}$ as a proxy for the MM policy, which due to result in Theorem 1 does not lead to any loss of generality.

Algorithm 2 SEQSYNTH($\mathcal{B}_c$, N_c)

1: $\widehat{\gamma} = \mathcal{N}^n$
2: **while** $(\mathbf{m}^0 \in \widehat{\gamma}) \wedge (\{\mathcal{B}_c\}_{c=1}^{l}$ are not enforced) **do**
3: **for** $c = 1$ to l **do**
4: $\widehat{\gamma}$=SYNTHESIZE($\widehat{\gamma}, \mathcal{B}_c, N_c$)
5: N_{c+1}=COMPOSE($\mathcal{B}_c, N_c$)
6: **end for**
7: **end while**

The algorithm stops either when the initial marking drops out of the estimate $\widehat{\gamma}$ or when property $\wedge_{c=1}^{l}\mathcal{B}_c$ can be enforced by enforcing $\widehat{\gamma}$. However, whether a specific instance of the procedure (for given $\{\mathcal{B}_c\}_{c=1}^{l}$) will terminate or not will depend on the subroutine SYNTHESIZE. The subroutine SYNTHESIZE takes the current estimate of $\widehat{\gamma}$ and modifies it so that another property $\mathcal{B}_c$ can be enforced. For several properties the while loop in Algorithm 2 will terminate in a single iteration. But as discussed in Section I, a PN that is live can lose liveness if some markings that were originally reachable cannot be reached anymore. Therefore, if liveness is one of the $\mathcal{B}_i$s, then the while loop in Algorithm 2 might have to execute several times till all the properties are enforced. Due to the typical nature of liveness, in the next section present a software tool for synthesizing an MM-$\mathcal{L}$ESP for an arbitrary PN N.

IV. AN ALGORITHM FOR MM-$\mathcal{L}$ESP SYNTHESIS

Suppose $\mathbf{m}^0 \in \Omega$ and $\mathcal{P}_\Omega$ is an MM-policy that enforces the control-invariant set Ω. We can construct the coverability graph, $G(N(\mathbf{m}^0), \mathcal{P}_\Omega)$, of $N(\mathbf{m}^0)$ under the supervision of $\mathcal{P}_\Omega$, along the same lines as the coverability graph of a PN (cf. section 4.2.1, [6]). The policy $\mathcal{P}_\Omega$ is an MM-$\mathcal{L}$ESP for $N(\mathbf{m}^0)$ if and only if (i) $\mathbf{m}^0 \in \Omega$, and (ii) there is a closed-path $v \xrightarrow{\sigma} v$ in $G(N(\mathbf{m}^i), \mathcal{P}_\Omega)$ for each $\mathbf{m}^i \in \min(\Omega)$ such that all transitions appear at least once in σ, and the net-change in tokens in each place after the firing of σ is non-negative (i.e. $\mathbf{C}\mathbf{x}(\sigma) \geq \mathbf{0}$). References [9], [10] present the implementation details of an $\mathcal{L}$ESP synthesis algorithm for a class of PN structures using object-oriented concepts like encapsulation and polymorphism. Extending the $\mathcal{L}$ESP-synthesis code for MM-$\mathcal{L}$ESP synthesis required just a few modifications, which are described below.

The $\mathcal{L}$ESP-synthesis implementation was done within four major classes called PetriNet, NodeTable, MinimalElementsManager and MarkingVector, described in detail in [9] (cf. Figure 3). The PetriNet class has a member function doTheLoopTest() that checks for the path-requirement on the coverability graph as described in Figure 2. This test uses an ILP-based algorithm described in the appendix of [5] using the API of the MILP-solver lp_solve [11]. Among the other member functions of the PetriNet class is a member function computeMinimalElementsOfControlInvariantSet() which computes the (minimal elements

- If $\mathbf{m}^0 \notin \Omega_i$, the procedure terminates with the conclusion that there is no MM-$\mathcal{L}$ESP for $N(\mathbf{m}^0)$.
- If $\mathbf{m}^0 \in \Omega_i$, and Ω_i is not control invariant with respect to N, it is replaced by its largest control invariant subset, Ω_{i+1} where $\Omega_{i+1} \subset \Omega_i$. The process is repeated with $\Omega_i \leftarrow \Omega_{i+1}$.
- If $\mathbf{m}^0 \in \Omega_i$, and Ω_i is control invariant with respect to N, each minimal element of the control invariant, right-closed set Ω_i is tested for the path-requirement on its coverability graph.
- If all minimal elements satisfy this requirement, then the members of $\min(\Omega)$ are presented as a description of the MM-$\mathcal{L}$ESP for $N(\mathbf{m}^0)$.
- If there are minimal elements that do not meet the path-requirement, then each minimal element $\mathbf{m}^i$ that fails the requirement is "elevated" by n-many unit-vectors, where n is the number of places in N. The resulting markings define a right-closed set $\Omega_{i+1} \subset \Omega_i$. After this, the process is repeated with $\Omega_i \leftarrow \Omega_{i+1}$.

Fig. 2. Pseudo-code for the deciding if there is an MM-$\mathcal{L}$ESP for $N(\mathbf{m}^0)$.

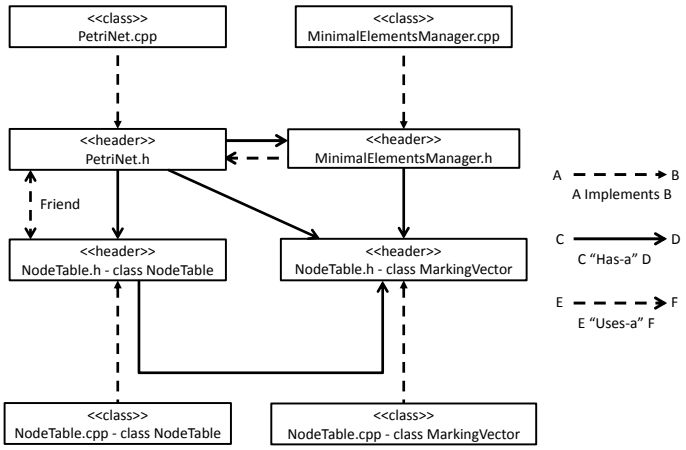

Fig. 3. The class structure of the Object-Oriented Implementation for $\mathcal{L}$ESP synthesis from [9], which was re-used for MM-$\mathcal{L}$ESP synthesis.

of) largest control-invariant subset of any right-closed set of markings that was not control invariant with respect to N. This member function was modified, in the implementation for MM-$\mathcal{L}$ESP synthesis, to check if $\mathbf{m}^0 \in \Omega_{i+1}$ before making the substitution of $\Omega_i \leftarrow \Omega_{i+1}$. The other classes (i.e. NodeTable, MinimalElementsManager and MarkingVector) and their member functions were kept with no change. The input and output structure of the earlier implementation was retained without change for the MM-$\mathcal{L}$ESP synthesis as well.

V. AN ILLUSTRATIVE EXAMPLE

We illustrate the procedure in Algorithm 1 and 2 using an example. We are interested in synthesizing a supervisory

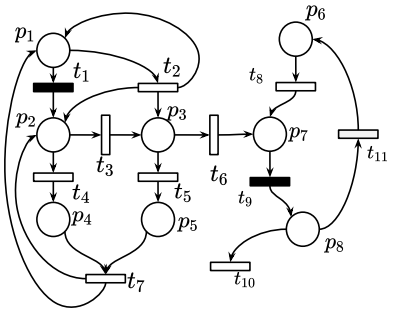

Fig. 4. Petri Net, N_1, to illustrate the procedure of Algorithms 1 and 2.

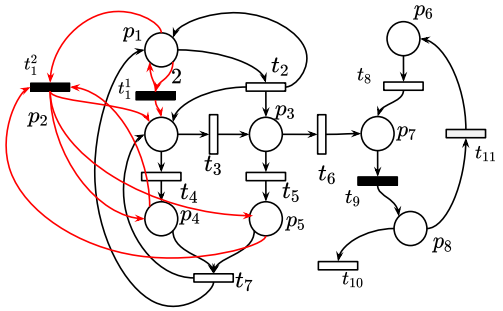

Fig. 5. Petri Net, N_2.

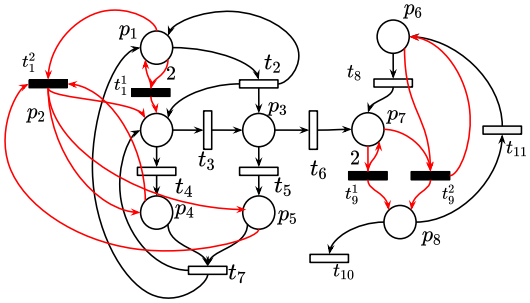

Fig. 6. Petri Net, N_3.

policy, $\mathcal{P}$, for the PN N_1 as shown in Figure 4. If $\mathbf{m}^0$ is the initial marking, then the objectives for supervisions are: (1) $N_1(\mathbf{m}^0)$ should be live; and (2) $\forall \mathbf{m}^1 \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$, $\mathbf{m}^1(p_6)+\mathbf{m}^1(p_7)+\mathbf{m}^1(p_8) \geq 1$. The set of initial markings for which an $\mathcal{L}$ESP exists, $\mathfrak{D}(\mathcal{L}, N_1)$, is given by the right-closed set with minimal elements $\widetilde{\mathbf{m}}_1$ = (1 0 0 0 0 0 0 0)^T and $\widetilde{\mathbf{m}}_2$ = (0 0 0 1 1 0 0 0)^T ([10]). Let $\mathbf{C}_1$ = (-1 1 0 0 0 0 0 0)^T denote the column of the incidence matrix $\mathbf{C}$ corresponding to transition t_1. Applying the result from Lemma 1, we get two minimal elements of $\mathcal{T}_M(\mathcal{L}, N_1, t_1)$, $\{\widehat{\mathbf{m}}_i\}_{i=1,2}$, as: $\widehat{\mathbf{m}}_1 = \max\{\mathbf{0}, \widetilde{\mathbf{m}}_1 - \mathbf{C_1}\}$ = (2 0 0 0 0 0 0 0)^T and $\widehat{\mathbf{m}}_2 = \max\{\mathbf{0}, \widetilde{\mathbf{m}}_2 - \mathbf{C_1}\}$ = (1 0 0 1 1 0 0 0)^T. Upon firing of t_1 from $\widehat{\mathbf{m}}_1$ and $\widehat{\mathbf{m}}_2$, we get the markings $\overline{\mathbf{m}}^1$ = (1 1 0 0 0 0 0 0)^T and $\overline{\mathbf{m}}^2$ = (0 1 0 1 1 0 0 0)^T respectively. The first thing to note is that $\overline{\mathbf{m}}_1, \overline{\mathbf{m}}_2 \in \mathfrak{D}(\mathcal{L}, N_1)$. The PN N_2 which is a composition of N_1 and the $\mathcal{L}$ESP $\mathcal{P}_\mathcal{L}$ is shown in Figure 5. Transition t_1 in N_1 is replaced by two new controllable transitions t_1^1 and t_1^2. The input (output) arc-weights of t_1^1 and t_1^2 correspond to $\widehat{\mathbf{m}}^1$ and $\widehat{\mathbf{m}}^2$ (resp. $\overline{\mathbf{m}}^1$ and $\overline{\mathbf{m}}^2$) respectively. Transition t_9 will always be enabled by the $\mathcal{L}$ESP $\mathcal{P}_\mathcal{L}$. Therefore, there is no change in it. It can be verified that $\mathfrak{R}(N_2, \mathbf{m}^0) = \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P}_\mathcal{L})$. Next we synthesize an $\mathcal{L}$ESP that enforces the property: $\forall \mathbf{m}^1 \in \mathfrak{R}(N_1, \mathbf{m}^0, \mathcal{P})$, $\mathbf{m}^1(p_6)+\mathbf{m}^1(p_7)+\mathbf{m}^1(p_8) \geq 1$. Let $\mathfrak{D}_M(\mathcal{C}, N_1) = \{\mathbf{m} \in \mathcal{N}^n : \mathbf{m}(p_6) + \mathbf{m}(p_7) + \mathbf{m}(p_8) \geq 1\}$. Since the tokens in place p_8 can be lost by the uncontrolled firing of t_{10}, transition t_9 should be controlled enabled if and only if the resulting marking is in $\mathfrak{D}_M(\mathcal{C}, N_1)$. By Lemma 1, the minimal elements of $\mathcal{T}_M(\mathcal{C}, N_1, t_1)$ are: $\{$(0 0 0 0 0 0 2 0)^T, (0 0 0 0 0 1 1 0)$^T\}$. For this particular example, the intersection of $\mathfrak{D}_M(\mathcal{L}, N_1)$ and $\mathfrak{D}_M(\mathcal{C}, N_1)$ gives us an estimate of the set of markings for which a supervisory policy that enforces both properties exists.

VI. Conclusion

In this paper, we formalized the concept of composition for a PN model and supervisory policy. A PN model of the supervised system was obtained by systematically replacing a controllable transition in the plant model by a set of transitions. We proved that there exists a composition of a PN model and a supervisory policy if and only if the supervisory policy is marking monotone over its reachable markings. We also presented a procedure for obtaining a Petri net model of the supervised system. One of the open problem is to identify different interpretations of composition and to obtain similar results.

References

[1] A. Giua and F. DiCesare, "Blocking and controllability of petri nets in supervisory control," *IEEE Transactions on automatic control*, vol. 39, no. 4, pp. 818–823, 1994.

[2] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM Journal on Control and Optimization*, vol. 25, no. 5, pp. 1202–1218, 1987.

[3] ——, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[4] A. Giua and F. DiCesare, "Supervisory design using petri nets," in *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE, 1991, pp. 92–97.

[5] R. S. Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled petri nets," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 928–945, 1997.

[6] J. L. Peterson, "Petri net theory and the modeling of systems," 1981.

[7] R. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice petri nets," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 435–449, 2012.

[8] C. Chen, A. Raman, H. Hu, and R. S. Sreenivas, "On liveness enforcing supervisory policies for arbitrary petri nets," *conditionally accepted, IEEE Transactions on Automatic Control*.

[9] S. Chandrasekaran, "Object-oriented implementation of the minimally restrictive liveness enforcing supervisory policy in a class of petri nets," *University of Illinios at Urbana-Champaign*, 2013.

[10] S. Chandrasekaran, N. Somnath, and R. Sreenivas, "A software tool for the automatic synthesis of minimally restrictive liveness enforcing supervisory policies for a class of general petri net models of manufacturing-and service-systems," *Journal of Intelligent Manufacturing*, vol. 26, no. 5, pp. 945–958, 2015.

[11] http://lpsolve.sourceforge.net, [Online; Last Accessed: June 25, 2019].