Petri Net Structural Analysis for Supervisory Control

Alessandro Giua,

Dip. di Ingegneria Elettrica ed Elettronica, Università di Cagliari,

Piazza d'Armi, 09123 Cagliari, Italy

Phone: +39-070-675-5892 – Fax: +39-070-675-5900 – Email: giua@diee.unica.it

Frank DiCesare

Dept. of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute

Troy NY 12180-3590, USA – Email: dicesare@ecse.rpi.edu

Abstract

The primary motivation for this research is to show how Petri nets may be efficiently used within the framework of Supervisory Control. In particular, the paper discusses how Integer Programming techniques for Petri net models may be used to validate supervisors for the control of discrete event systems.

We consider a class of Place/Transition nets, called Elementary Composed State Machines. The reachability problem for this class can be solved by a modification of classical incidence matrix analysis. In fact it is possible to derive a set of linear inequalities that exactly defines the set of reachable markings. Finally, we show how important properties of discrete event systems, such as the absence of blocking states or controllability, may be analyzed by Integer Programming techniques.

Published as:
A. Giua, F. DiCesare, "“Petri Net Structural Analysis for Supervisory Control," *IEEE Trans. on Robotics and Automation*, Vol. 10, No. 2, pp. 185-195, April, 1994.

1 Introduction

This paper discusses how Integer Programming techniques for Petri net (PN) models may be used to validate supervisors for the control of discrete event systems. We will consider a class of Place/Transition (P/T) nets called *Elementary Composed State Machines* (ECSM). The most interesting property of this class of nets is given by the fact that the set of reachable markings is an (integer) convex set. The set of linear inequalities that define the reachability set may be computed, following our approach, from the analysis of the simple state machine modules that compose the net.

1.1 Motivation

The primary motivation for this research is to show how Petri nets may be efficiently used within the framework of Supervisory Control, a control theory for discrete event systems originated by the work of Ramadge and Wonham [21]. Although Supervisory Control theory is well established, we still lack effective models for supervisory controllers capable of coping with the state space explosion that is a characteristic of these complex systems.

Holloway and Krogh [13] have used *controlled Petri nets* (a model introduced in [14]) for the efficient solution of a class of control problems. The main characteristic of the controlled PN approach to Supervisory Control, is the fact that no transition structure for a controller is given. The control law is a function of the actual marking of the net, but needs to be computed at each step.

The approach followed in this paper is different and is based on the design procedure given by Wonham [22] where a transition structure is computed for a supervisor. Thus a closed-loop model is constructed and analyzed to validate the desired properties. However while in [22] the models used are finite state automata, in this paper Place/Transition (P/T) nets are used. In [9] is discussed how conservative P/T nets constructed by concurrent composition may be used to design supervisors. The advantages of Petri nets over state machines are:

- Since the states of a PN are represented by the possible markings and not by the places, they allow a compact description, i.e., the structure of the net may be maintained small in size even if the number of the markings grows.
- They allow modular synthesis, i.e., the net can be considered as composed of interrelated subnets, in the same way as a complex system can be regarded as composed of interacting subsystems.

This paper shows how some of the PN analysis techniques, namely those based on the structure of the net, may be used to validate the net for properties of interest without resorting to the construction of the corresponding reachability tree. The properties of interest in a Supervisory Control problem are the absence of *blocking states* and *controllability*.

In classic incidence matrix analysis, the set of reachable markings of a system $\langle N, M_0 \rangle$, denoted $R(N, M_0)$, is approximated by the solutions of the *state equation*, i.e., by the set $PR(N, M_0) = \{M \in \mathbb{N}^{|P|} \mid (\exists \vec{\sigma} \in \mathbb{N}^{|T|})[M = M_0 + C \cdot \vec{\sigma}]\}$, where C is the incidence matrix of the net. In another approach [5], a *basis of P-semiflows* B is used to approximate the reachability set, defining the set $PR^B(N, M_0) = \{M \in \mathbb{N}^{|P|} \mid B^T \cdot M = B^T \cdot M_0\}$. The first approximation is generally better, in the sense that $R(N, M_0) \subseteq PR(N, M_0) \subseteq PR^B(N, M_0)$. However, the computation of $PR^B(N, M_0)$ does not involve the firing count vector $\vec{\sigma}$, and this feature has additional advantages that we will discuss in the following.

We propose an approach similar to the computation of $PR^B(N, M_0)$ where we use, along with the equations derived from the P-semiflows, inequalities derived from the basic traps of the net and we define $PR^A(N, M_0) = \{M \in \mathbb{N}^{|P|} \mid A \cdot M \geq \vec{b}\}$ [10] where $\vec{b}$ is a vector that depends on M_0. We show how A and $\vec{b}$ may be computed for ECSM nets and we prove that for this class of nets $PR^A(N, M_0) = R(N, M_0)$.

There are significant differences between our approach and the analysis based on the state equation.

Firstly, propositions such as $(\forall M \in PR(N, M_0))\ [M_f \in PR(N, M)]$ (i.e., M_f is a home-state) cannot be verified with a linear algebraic formalism because the set $PR(N, M_0)$ is defined in terms of linear equations containing the variables M and $\vec{\sigma}$. This proposition expresses the nonblocking property [21] of a discrete event system modeled by P/T nets. If we define the set of reachable markings as the solution of a set of inequalities that do not contain the firing count vector $\vec{\sigma}$, we will be able to write simple Integer Programming problems to study nonblocking properties of systems [8].

Secondly, for the class of ECSM nets, the state equation gives only necessary but not sufficient conditions for reachability, since it may contain spurious solutions (solutions which do not correspond to reachable markings), i.e., in general $PR(N, M_0) \supset R(N, M_0)$. However, for the same class of nets there exists a matrix A such that $PR^A(N, M_0) = R(N, M_0)$, i.e., the reachability set of an ECSM may be exactly described by a set of linear inequalities, without spurious solutions.

Thirdly, note that since the domain $PR^A(N, M_0)$ represents the integer solutions of a set of linear inequalities, it is a convex set of integers. Thus, there exists a matrix A such that $PR^A(N, M_0) = R(N, M_0)$ only if the reachability set of system $\langle N, M_0 \rangle$ is a convex

set. If a net is such that the state equation does not contain spurious solutions, this does not imply that its reachability set is a convex set. As an example, it has been proved that the state equation of acyclic nets does not contain spurious solutions [15]. In [10] we gave an example of a net that is acyclic but does not have a convex reachability set.

In our approach determining if a marking is reachable requires the use of Integer Programming. Integer Programming problems may not be solved, in general, in polynomial time. However, it is possible to relax the constraint that the solution be integer to obtain a sufficient condition for the validation of the net. Thus we may use Linear Programming techniques to prove that a given undesirable marking is not reachable.

1.2 Related Work

Incidence matrix and related analysis techniques have been used by several authors to validate properties of Place/Transition nets.

Colom [5] has developed a methodology for the verification of assertions on P/T nets in terms of markings and firing count vectors. This approach is extremely general, i.e., can be applied to any P/T net, but unfortunately can only guarantee necessary or sufficient conditions. There exist assertions, such as determining if a marking is a home state, for which neither a necessary nor a sufficient condition may be given.

In the work of Colom [5] other approaches to describe the set of reachable markings are also discussed. Berthomieu [3] uses the set of linear equations given by the P-semiflows to represent the space of reachable markings. In this case, only properties that depend on the markings may be proved. However, the author also shows that it is possible to prove some properties that depend on the firing count vector by adding new places in the net, whose marking indicates the number of times a given transition has fired. (The computation of the firing bounds, introduced in Section 4.2 of this paper, has the same purpose.) Johnen [17] uses an hybrid approach, based partly on incidence matrix analysis and partly on the analysis of the state space, to verify that a given marking is a home state. (Note that the problem of determining a home state is essentially equivalent to that of determining if a supervisor is blocking, that is discussed in Section 5.1 of this paper.)

Ichikawa and Hiraishi have studied under which conditions a firing count vector $\vec{\sigma}$, satisfying the state equation of a Petri net, yields a firing sequence. In [15], as reported by Murata [20], they proved that in acyclic nets, $\vec{\sigma}$ always yields a firing sequence, i.e., for this class of nets $R(N, M_0) = PR(N, M_0)$. In [14, 20] other classes of nets, such as trap-circuit nets, trap-containing-circuit nets, etc., are considered. For these classes, there exists necessary and sufficient conditions, based on the analysis of the firing subnet, to

determine if $\vec{\sigma}$ gives a firing sequence.

The composition of nets by common transitions has also been extensively investigated and is discussed by Berthelot [2]. Best and Fernández [4] define *S-net* a net in which each transition has at most one input place and one output place. An *S-decomposition* is a partition of a net into S-net components. Hack [12] has defined *state machine decomposable net* a net constructed by composition of strongly connected state machines; the liveness properties for this class of nets are discussed by Jantzen and Valk [16]. The class of nets obtained by composition of state machine modules has been named *Superposed Automata Nets* by De Cindio, *et al.* [6].

Avrunin *et al.* [1] have used a formalism similar to Petri net incidence matrix analysis for verifying properties of concurrent systems described as finite state automata.

Li and Wonham [18, 19] have discussed the use of vector addition systems, that are equivalent to Petri nets, as discrete event models. There several differences between their approach and ours. Firstly, in Li and Wonham's approach Integer Programming is used to *compute* the optimal control law and only the open loop system is modeled as a vector addition system. In this paper, instead, we first construct a candidate monolithic supervisor by concurrent composition and then use Integer Programming to *validate* it. Secondly, while in [18] the set $PR(N, M_0)$ is used to represent the reachability set, we use the set $PR^A(N, M_0)$. This was done to study nonblocking properties that cannot be studied with the state equation. Finally, there are different restrictions in the two approaches. In [18] the uncontrollable transitions of the system must not form cycles (so that the state equation analysis gives necessary and sufficient conditions for reachability as shown in [15]); furthermore, the class of control specifications is restricted to *mutual exclusion constraints*[1], that limit the weighted sum of tokens contained in the places of a Petri net. In this paper the restriction is that the monolithic supervisor be an ECSM.

1.3 The Model

The model considered in this paper is based on state machine Place/Transition nets with multiple tokens. State machines may model choice, since a place may have more that one outgoing arc, but strongly limit the possibility of modeling concurrency, since the only concurrent behavior is given by the presence of multiple tokens in the net. To describe concurrent systems, the model is extended by composing the state machine modules through concurrent composition, an operator that requires the merging of common transitions.

[1]This class of specifications for Petri nets has also been considered by [11, 13] in the framework of Supervisory Control.

In this approach it is necessary to restrict the type of compositions considered, in order to guarantee some important properties. The final model, called Elementary Composed State Machines (ECSM nets) [10], can model both choice and concurrent behavior, and at the same time has the property that the space of reachable markings is a linear integer convex set, i.e., it is given by the integer solutions of a set of linear inequalities.

In particular there is an algorithm to determine this set of inequalities that defines the set of reachable markings. The inequalities are derived from the computation of the basic traps of the state machine modules and from the computation of the firing bounds of the merged transitions.

The paper is structured in six sections. In Section 2 is given the basic notation on Petri nets. Section 3 deals with state machines and shows how it is possible to derive the set of linear inequalities that defines the space of reachable markings. In Section 4 is defined the class of Elementary Composed State Machine nets and the results derived in Section 3 are extended to this class. Section 5 shows how this approach may be applied to the validation of supervisors for the control of discrete event systems. A discussion of the results is presented in Section 6.

2 Background

The basic notation on Petri nets is introduced in this section, following [4, 20].

2.1 Basic Terminology

A *Place/Transition net* (P/T net) is a structure $N = (P, T, I, O)$ where: P is a set of *places* represented by circles, $\| P \| = n$; T is a set of *transitions* represented by bars, $\| T \| = m$; $I : P \times T \to I\!N$ is the *input function* that specifies the arcs directed from places to transitions; $O : P \times T \to I\!N$ is the *output function* that specifies the arcs directed from transitions to places.

The *preset* and *postset* of a transition t are respectively: ${}^\bullet t = \{p \in P \mid I(p,t) > 0\}$ and $t^\bullet = \{p \in P \mid O(p,t) > 0\}$. The *preset* and *postset* of a place p are respectively: ${}^\bullet p = \{t \in T \mid O(p,t) > 0\}$ and $p^\bullet = \{t \in T \mid I(p,t) > 0\}$.

A *marking* is a vector $M : P \to I\!N$ that assigns to each place of a P/T net a non negative integer number of tokens, represented by black dots. $M(p)$ indicates the number of tokens assigned by marking M to place p. A *marked net* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition $t \in T$ is *enabled* by a marking M if $(\forall p \in P)\ [M(p) \geq I(p,t)]$. The firing of transition t generates a new marking M' with: $M'(p) = M(p) + O(p,t) - I(p,t)$. When a marking M' can be reached from marking M by executing a (possibly empty) *firing sequence* of transitions $\sigma = t_1 \ldots t_k$ we write $M\ [\sigma\rangle\ M'$. The set of markings reachable on a net N from a marking M is called *reachability set* of N and M and is denoted as $R(N,M)$.

The *incidence matrix* of a net $N = (P,T,I,O)$ is a $(n \times m)$ matrix of integers defined as: $C = \{c_{ij} \mid c_{ij} = O(p_i,t_j) - I(p_i,t_j)\}$. If marking M' is reachable from marking M by firing a sequence of transitions σ, the following *state equation* is satisfied: $M' = M + C \cdot \vec{\sigma}$, where $\vec{\sigma} : T \to I\!N$ is a vector of non-negative integers, called the *firing count vector*. $\vec{\sigma}(t)$ represents the number of times transition t has fired during the execution of σ.

A *trap* is a set of places $\mathcal{T} \subseteq P$ such that: $\bigcup_{p \in \mathcal{T}} p^\bullet \subseteq \bigcup_{p \in \mathcal{T}} {}^\bullet p$. A trap is *minimal* if it is not the superset of any other trap. A trap is *basic* if it is not the disjoint union of other traps.

A *P-semiflow* is a vector $Y : P \to I\!N$ such that $Y \geq \vec{0}$ and $Y^T \cdot C = \vec{0}$ (^{T} is the transpose operator). The *support* of Y is: $\| Y \| = \{p \in P \mid Y(p) > 0\}$. The support of a P-semiflow is both a trap and a siphon.

The *reversal* of a net $N = (P,T,I,O)$ is the net $N^R = (P,T,O,I)$, i.e., a new net where the direction of all arcs of N is reversed.

2.2 State Machines and Simple Paths

A *state machine* is a P/T net such that each transition has exactly one input arc and one output arc. A state machine is *connected* if the underlying graph (i.e., the graph having as nodes places and transitions, and as edges the arcs) is connected; *strongly connected* if for any two nodes there exists a directed path from each one to the other; *acyclic* if no directed path forms a cycle.

A *simple path* of a net N is a sequence of transitions and places $\theta = t_0 p_1 t_1 \ldots p_r t_r$ containing no place or transition more than once and such that: $(\forall i = 1, \ldots, r)\ [I(p_i,t_i) = O(p_i,t_{i-1}) = 1 \wedge I(p_i,t) = 0$ **if** $t \neq t_i \wedge O(p_i,t) = 0$ **if** $t \neq t_{i-1} \wedge I(p,t_i) = 0$ **if** $p \neq p_i \wedge O(p,t_{i-1}) = 0$ **if** $p \neq p_i]$. Note that a single transition may be considered as a simple path with no places.

Given a net N, and k simple paths $\theta_i = t_{i,0} \ldots t_{i,r_i}\ \ (i = 1, \ldots, k)$, the k paths are looped in N if: $(\exists p \in P)\ (\forall i = 1, \ldots, k)\ [I(p,t_{i,0}) = O(p,t_{i,r_i}) = 1 \wedge (\forall p' \neq p)\ I(p',t_{i,0}) = O(p',t_{i,r_i}) = 0]$. In simple words, the k paths are looped if the input (and output)

transitions of all paths input from (and output to) the same place p with a single arc.

2.3 Composition, Projection and Modularity

Definition 2.1. *Given two nets* $N_1 = (P_1, T_1, I_1, O_1)$ *and* $N_2 = (P_2, T_2, I_2, O_2)$*, with initial markings* $M_{0,1}$ *and* $M_{0,2}$*, assume that* $\Theta = \{\theta_1, \ldots, \theta_k\}$ *is a set of simple paths present in both nets. We assume that:* $P_1 \cap P_2 \setminus \{p \mid (\exists \theta \in \Theta)[p \in \theta]\} = \emptyset$*, and* $T_1 \cap T_2 \setminus \{t \mid (\exists \theta \in \Theta)[t \in \theta]\} = \emptyset$*. We also assume that* $(\forall p \in P_1 \cap P_2)\ [M_{0,1}(p) = M_{0,2}(p)]$*. The* concurrent composition *of* N_1 *and* N_2 *is the net* $N = (P, T, I, O)$ *with initial marking* M_0 *where:* $P = P_1 \cup P_2$*,* $T = T_1 \cup T_2$*,* $I(p,t) = I_i(p,t)$ **if** $(\exists i \in \{1,2\})\ [p \in P_i \wedge t \in T_i]$ **else** $I(p,t) = 0$*;* $O(p,t) = O_i(p,t)$ **if** $(\exists i \in \{1,2\})\ [p \in P_i \wedge t \in T_i]$ **else** $O(p,t) = 0$*;* $M_0(p) = M_{0,1}(p)$ **if** $p \in P_1$ **else** $M_0(p) = M_{0,2}(p)$*. The composed net* N *is denoted* $N = N_1 \parallel N_2$*.*

Definition 2.2. *Let* N *be a composed net* $N = N_1 \parallel \ldots \parallel N_n$*, and let* M *be a marking,* $\vec{\sigma}$ *be a firing count vector, and* σ *a firing sequence defined on it. The* projection of M over N_i*, denoted* $\mathcal{P}_i(M)$*, is the vector obtained from* M *by removing all the components associated to places not present in* N_i*. The* projection of $\vec{\sigma}$ over N_i*, denoted* $\mathcal{P}_i(\vec{\sigma})$*, is the vector obtained by* $\vec{\sigma}$ *removing all the components associated to transitions not present in* N_i*. The* projection of σ over N_i*, denoted* $\mathcal{P}_i(\sigma)$*, is the firing sequence obtained by* σ *removing all the transitions not present in* N_i*.*

From this definition, it follows that while N generates the string σ sequencing from M_0 to M, N_i generates $\mathcal{P}_i(\sigma)$ sequencing from $\mathcal{P}_i(M_0)$ to $\mathcal{P}_i(M)$.

3 Defining the Set of Reachable Markings on State Machines

In this section we discuss how the reachability set of a state machine may be described by a set of equations that do not involve the firing count vector.

State machines represent a very simple PN model that has been extensively investigated. For instance, Murata [20] reports several results on the liveness and reachability characterization of this class of nets. However, the focus is generally on live state machines (i.e., strongly connected state machines). In the framework of Supervisory Control, the requirement that the model be live is not important, while it is required that the system be *non-blocking*, i.e., that from any reachable marking it may be possible to reach a final marking. This motivates the attention given in this paper to state machines not

necessarily live.

Theorem 3.1. *Let $\langle N, M_0 \rangle$ be a marked state machine. The set of reachable markings $R(N, M_0)$ is an integer linear convex set, i.e., can be defined as the integer solutions of a set of linear inequalities.*

Proof: Follows from the constructive Algorithm 3.1 presented in the following. $\diamond$

To determine the set of linear inequalities that the set of reachable markings of a state machine must satisfy, the following algorithm may be used. Note first that for a connected (not necessarily strongly connected) state machine a basis of P-semiflows contains only one vector whose support contains all the places.

Algorithm 3.1. Let $\langle N, M_0 \rangle$ be a state machine, and $\mathcal{T}^i$ a trap of N. Let $\vec{a}_i : P \times \{0, 1\}$ be such that $\vec{a}_i(p) = 1$ **if** $p \in \mathcal{T}^i$ **else** $\vec{a}_i(p) = 0$.

1. Consider all basic traps of the net along with the support of the P-semiflow $\mathcal{T}^0$.
2. For $\mathcal{T}^0$ write the equality: $\vec{1}^T \cdot M = \vec{1}^T \cdot M_0$.
3. For each trap $\mathcal{T}^i$ write the inequality: $\vec{a}_i^T \cdot M \geq \vec{a}_i^T \cdot M_0$.
4. If $\vec{a}_i^T \cdot M_0 = 0$, the inequality for $\mathcal{T}^i$ $(i \neq 0)$ can be removed.
5. If $\mathcal{T}^i \subset \mathcal{T}^j$ $(j \neq 0)$ and $\vec{a}_i^T \cdot M_0 = \vec{a}_j^T \cdot M_0$, the inequality for $\mathcal{T}^j$ can be removed, since it is implied by the inequality for $\mathcal{T}^i$.
6. The remaining set of inequalities plus the inequalities: $M \geq 0$, gives the set of markings reachable from the initial marking. These inequalities will be indicated as $\mathcal{A}(N)$.

The semiflow equation may be rewritten as two inequalities: $\vec{1}^T \cdot M \geq \vec{1}^T \cdot M_0$ and $-\vec{1}^T \cdot M \geq -\vec{1}^T \cdot M_0$. Hence, according to this algorithm the reachability set of a state machine $\langle N, M_0 \rangle$ can be represent by a set $PR^A(N, M_0) = \{M \in \mathbb{N}^{|P|} \mid A \cdot M \geq \vec{b}\}$, where $\vec{b} = A \cdot M_0$.

Here is an example of application for the algorithm.

Example 3.1. *Consider the net in Figure 1. Here the support of the P-semiflow and the*

basic traps are:

$$\mathcal{T}^0 = \{p_1, p_2, p_3, p_4, p_5, p_6\};$$
$$\mathcal{T}^1 = \{p_2\};$$
$$\mathcal{T}^2 = \{p_5\};$$
$$\mathcal{T}^3 = \{p_6\};$$
$$\mathcal{T}^4 = \{p_1, p_2, p_5\};$$
$$\mathcal{T}^5 = \{p_3, p_5, p_6\};$$
$$\mathcal{T}^6 = \{p_3, p_4, p_5, p_6\};$$
$$\mathcal{T}^7 = \{p_1, p_2, p_3, p_5, p_6\}.$$

Note, e.g., that the trap $\{p_2, p_5\}$ is not considered, since it is given by the union of the disjoint sets $\mathcal{T}^1$ and $\mathcal{T}^2$. The corresponding set of linear inequalities is:

$$M(p_1) + M(p_2) + M(p_3) + M(p_4) + M(p_5) + M(p_6) = 4;$$
$$M(p_2) \geq 0;$$
$$M(p_5) \geq 0;$$
$$M(p_6) \geq 1;$$
$$M(p_1) + M(p_2) + M(p_5) \geq 2;$$
$$M(p_3) + M(p_5) + M(p_6) \geq 2;$$
$$M(p_3) + M(p_4) + M(p_5) + M(p_6) \geq 2;$$
$$M(p_1) + M(p_2) + M(p_3) + M(p_5) + M(p_6) \geq 4.$$

The second and third inequality will be removed in step 4 of the algorithm; the the seventh inequality will be removed in step 5 as it is implied by the sixth one. Finally the set of markings reachable from the initial marking is given by:

$$M(p_1) + M(p_2) + M(p_3) + M(p_4) + M(p_5) + M(p_6) = 4;$$
$$M(p_6) \geq 1;$$
$$M(p_1) + M(p_2) + M(p_5) \geq 2;$$
$$M(p_3) + M(p_5) + M(p_6) \geq 2;$$
$$M(p_1) + M(p_2) + M(p_3) + M(p_5) + M(p_6) \geq 4;$$
$$M \geq \vec{0}.$$

Finally, it is necessary to discuss the correctness of the algorithm, i.e., the fact that the set of markings that satisfy the linear inequalities required by Algorithm 3.1 is exactly the reachability set. The idea is to show that a description of a state machine in term of traps captures the behavior of the net. This will be proven through the following propositions.

Proposition 3.1. *Let $\mathcal{T}$ be a trap of a state machine. The number of tokens in $\mathcal{T}$ is non decreasing.*

Proof: Trivially follows from the definition of state machine. ◇

Proposition 3.2. *Let $\mathcal{T}$ be a minimal trap of a state machine, i.e., it is not a superset of any other trap. Then a token in $\mathcal{T}$ can move to any place in $\mathcal{T}$.*

Proof: For state machines, it is sufficient to prove that all places in $\mathcal{T}$ are strongly connected, i.e., there is a path from any place to all others. This is trivially true if $\mathcal{T}$ consist of a single place. Assume $\mathcal{T}$ consists of k places. Consider a place $p_1 \in \mathcal{T}$. There must be a transition from p_1 to some place p_2, otherwise $\{p_1\}$ is a trap; also $p_2 \in \mathcal{T}$. Consider $\{p_1, p_2\}$. With the same reasoning it is possible to infer that there must be a transition leading from $\{p_1, p_2\}$ to a place $p_3 \in \mathcal{T}$. This reasoning can be carried out until $\{p_1, p_2, ...p_k\} = \mathcal{T}$ is reached. Hence p_1 is connected to all places in $\mathcal{T}$. Since this reasoning can be applied to any place $p_i \in \mathcal{T}$, the proof is complete. ◇

Proposition 3.3. *Let $\mathcal{T}$ be a trap, and let $\mathcal{T}' = \{\cup\mathcal{T}^i \mid \mathcal{T}^i \subset \mathcal{T}, \mathcal{T}^i$ is a trap $\}$. Then a token in $\mathcal{T} \setminus \mathcal{T}'$ can move to any place in $\mathcal{T}$.*

Proof: Similar to the proof of Proposition 3.2. ◇

The soundness of the algorithm is then proved by the following reasoning. The marking of the trap is strictly non decreasing (Proposition 3.1). The tokens initially present in a minimal trap may freely move to any place of the trap (Proposition 3.2). The token initially present in a trap that is not minimal are free to move to any place of the trap provided they also satisfy the constraints enforced by the traps contained in the non minimal trap (Proposition 3.3). These are the only constraints that the set of reachable markings must satisfy and these constraints are captured by Algorithm 3.1.

We point out that for a general Petri net the number of basic traps may be exponential in the number of places. However, the next proposition holds for state machines.

Proposition 3.4. *The number of basic traps in a state machine is at most equal to the number of places.*

Proof: Consider a state machine with n places and no transitions; clearly there are n basic traps each one containing a single place. Now assume we have a state machine with its set of basic traps, and assume we add a new transition from place p to place p'. Then all the basic traps that do not contain p or that contain p' are still basic traps. All the basic traps that contain p and do not contain p' will not be traps in the new net and we need to add to each of them the minimal trap containing p' (there is only one such minimal trap) to obtain new basic traps. After the new traps are computed, it may well be the case that two of them are identical, that is the number of basic traps may only decrease when we add a transition. Since any state machine may be obtained by this construction, the result follows. ◇

The construction we used in the proof of Proposition 3.4 may be used to compute the set of basic traps.

4 Composition of State Machines Modules

The section discusses how the properties studied in Section 3 for state machines are preserved by concurrent composition.

4.1 Elementary Composed State Machines

Let $N = N_1 \parallel \ldots \parallel N_n$ be a composed net and let $\vec{\sigma}$ be the firing count vector solution of: $M = M_0 + C \cdot \vec{\sigma}$, where C is the incidence matrix of N. Assume that on each module N_i $(\forall i = 1, \ldots, n)$, the firing count vector $\mathcal{P}_i(\vec{\sigma})$ yields a firable sequence σ_i, i.e., $\mathcal{P}_i(M_0)\ [\sigma_i\rangle\ \mathcal{P}_i(M),\ \ (i = 1, \ldots, n)$. This does not imply, however, that for the composed net $(\exists\sigma)\ [M_0\ [\sigma\rangle\ M]$.

There exist particular compositions, however, such that the reachability of a marking of the overall net may be determined simply by the analysis of the modules that compose it. These compositions, called elementary, are used to define the following class of P/T nets.
Definition 4.1. Elementary Composed State Machine *(ECSM) nets are the minimal class of P/T nets that is a superset of the class of state machines and is closed under the following compositions:*

1. *Composition of two nets sharing a single transition (or a simple path).*
2. *Composition of two nets through a set T_s of k transitions (or a set Θ of k simple paths) when the transitions (or simple paths) are looped in one of the nets.*

Although the two compositions we used to define ECSM may appear exceedingly simple, they permit the modular synthesis of realistic systems. As an example, the first kind of compositions may be used to construct the model of several systems *acyclically* connected through buffers or channels. The second kind of composition may be used to represent shared resources.

An important property of ECSM is given by the following theorem, proven in [10].
Theorem 4.1. *Let $\langle N, M_0\rangle$ be a marked ECSM net where $N = N_1 \parallel \cdots \parallel N_n$ and N_i, $(i = 1, \ldots n)$, is a state machine. A firing count vector for N yields a firable sequence if and only if on each module N_i the projection of the firing count vector yields a firable sequence.*

4.2 Defining the Set of Reachable Markings on ECSM

This subsection will show how it is possible to derive the set of linear inequalities that defines the set of reachable markings in ECSM nets. First, the case of two state machines, composed through a single transition or a set of looped transitions, is discussed. Then the results will be extended to the composition of ECSM through simple paths.

Definition 4.2. *Let $\langle N, M_0\rangle$ be a marked state machine where M_0 assigns all the tokens to a place p_0, and let M_p be a marking that assigns a single token to place p and no token elsewhere. Given a transition t, it is possible to define the following two sets of places on the net:* $P_t = \{p \in P \mid (\exists\sigma)\ [\vec{\sigma}(t) > 0,\ M_{p_0}\ [\sigma\rangle\ M_p]\}$; $P_{\overline{t}} = \{p \in P \mid (\exists\sigma)\ [\vec{\sigma}(t) = 0,\ M_{p_0}\ [\sigma\rangle\ M_p]\}$.

Similarly, given a set of transitions T_s, it is possible to define the following two sets of places on the net: $P_{T_s} = \{p \in P \mid (\exists\sigma)\ (\exists t \in T_s)\ [\vec{\sigma}(t) > 0,\ M_{p_0}\ [\sigma\rangle\ M_p]\}$; $P_{\overline{T_s}} = \{p \in P \mid (\exists\sigma)\ (\forall t \in T_s)\ [\vec{\sigma}(t) = 0,\ M_{p_0}\ [\sigma\rangle\ M_p]\}$.

In simple words, the set P_t contains the places that *may* be marked *firing* t at least once by a token contained in the initial place p_0, while the set $P_{\overline{t}}$ contains the places than *may* be marked *without firing* t by a token contained in the initial place p_0. Note that same places may belong to $P_t \cap P_{\overline{t}}$ and that the places in $P_t \setminus P_{\overline{t}}$ *must* be marked *firing* t at least once by a token contained in the initial place p_0.

Proposition 4.1. (Firing Bounds) *Let $\langle N, M_0\rangle$ be a marked state machine where M_0 assigns all the tokens to a place p_0. Now given a marking $M \in R(N, M_0)$, the number of times t has fired to reach M may vary and can assume any integer value in the range $[\rho_{\min}(M,t), \rho_{\max}(M,t)]$, where:*

$$\rho_{\min}(M,t) = \sum_{p \in P_t \setminus P_{\overline{t}}} M(p)$$

$$\rho_{\max}(M,t) = \begin{cases} \sum_{p \in P_t} M(p) & \text{if there is no cycle containing } t \\ 0 & \text{if there is a cycle containing } t \\ & \text{and } \sum_{p \in P_t} M(p) = 0 \\ \infty & \text{if there is a cycle containing } t \\ & \text{and } \sum_{p \in P_t} M(p) > 0 \end{cases}$$

We call $\rho_{\min}(M,t)$ and $\rho_{\max}(M,t)$ the firing bounds *of t for marking M.*

Proof: For each token in a place of $P_t \setminus P_{\overline{t}}$ transition t must have fired at least once, while for each token in a place of P_t transition t may have fired once if there is no cycle containing t or an arbitrary large number of times if there is a cycle containing t. ◇

Note that $\rho_{\min}(M,t) \leq \sum_{p \in P} M_0(p)$. Also, when there is a cycle containing t a good approximated bound — that will be used in the following — for $\rho_{\max}(M,t)$ is $\rho'_{\max}(M,t) = h \sum_{p \in P_t} M(p) \leq \rho_{\max}(M,t)$, where h is a sufficiently large integer.

Proposition 4.1 is restricted to the case of an initial marking that assigns all tokens to a single place. This requirement is fair, in the sense that in the application cases of interest here, such as manufacturing systems, etc., the presence of multiple tokens in a state machine module indicates a multiplicity of identical resources, such as buffer spaces or identical machines. Hence the multiple tokens should initially be assigned to the same place. A more detailed discussion on this point is presented in [8].

When two nets N_i $(i = 1, 2)$ are composed through a single transition t, the marking of the overall net will be a subset of the cartesian product of the markings of the two modules. Given a reachable marking M_i on the net N_i, the marking $M = [M_1^T M_2^T]^T$ will be reachable on the composed net only if (on ECSM nets "if and only if" because of Theorem 4.1) there is a sequence σ_i reaching M_i on the net N_i and: $\vec{\sigma}_1(t) = \vec{\sigma}_2(t)$. This requires that:

$$[\rho^1_{\min}(M_1,t), \rho^1_{\max}(M_1,t)] \bigcap [\rho^2_{\min}(M_2,t), \rho^2_{\max}(M_2,t)] \neq \emptyset$$

where the exponent i in $\rho^i_{\min}(M_i,t)$ and $\rho^i_{\max}(M_i,t)$ denotes that the firing bound is computed on the net N_i. Clearly the two intervals will have a nonempty intersection if and only if: $\rho^1_{\min}(M_1,t) \leq \rho^2_{\max}(M_2,t)$, and $\rho^2_{\min}(M_2,t) \leq \rho^1_{\max}(M_1,t)$. Thus we have the following theorem whose proof is given in [10].

Theorem 4.2. *When two state machines N_1 and N_2 are composed through a single transition t the set of markings reachable from the initial marking M_0 for the composed net $N = N_1 \parallel N_2$ is given by the following set of linear inequalities $\mathcal{A}(N)$:*

$$\mathcal{A}(N_1)$$

$$\mathcal{A}(N_2)$$

$$\sum_{p \in P^1_t \setminus P^1_{\overline{t}}} M(p) \leq h_2 \sum_{p \in P^2_t} M(p)$$

$$\sum_{p \in P^2_t \setminus P^2_{\overline{t}}} M(p) \leq h_1 \sum_{p \in P^1_t} M(p)$$

where: $\mathcal{A}(N_i)$ $(\forall i = 1, 2)$, is the set of inequalities for the net N_i (as derived with Algorithm 3.1); the set of places P^i_t and $P^i_{\overline{t}}$ belongs to N_i $(\forall i = 1, 2)$, and are determined as in Definition 4.2; $h_1 = 1$ $(h_2 = 1)$, if there is no cycle containing t in N_1 (N_2), else h_1 (h_2) is equal to the number of tokens contained in the net N_2 (N_1). As noted before, we are using an approximated linear bound for $\rho^i_{\max}(M_i,t)$.

Example 4.1. *Consider the composed system of Figure 2. The set of places of interest are:* $P_t^1 = \{p_2\}$; $P_{\overline{t}}^1 = \{p_1\}$; $P_t^2 = \{p_4, p_5, p_6, p_7$; $P_{\overline{t}}^2 = \{p_3, p_4, p_7\}$. *Also there is no cycle containing* t *in* N_1, *hence* $h_1 = 1$, *while, since there is a cycle containing* t *in* N_2, $h_2 = 3$. *The linear inequalities that define the space of reachable markings on* N_1 *is:*

$$M(p_1) + M(p_2) = 3$$

$$M(p_1), M(p_2) \geq 0$$

and on N_2*:*

$$M(p_3) + M(p_4) + M(p_5) + M(p_6) + M(p_7) = 2$$

$$M(p_3), M(p_4), M(p_5), M(p_6), M(p_7) \geq 0$$

Hence the set of reachable markings on the composed system is defined by:

$$M(p_1) + M(p_2) = 3$$

$$M(p_3) + M(p_4) + M(p_5) + M(p_6) + M(p_7) = 2$$

$$M(p_2) \leq 3(M(p_4) + M(p_5) + M(p_6) + M(p_7))$$

$$M(p_5) + M(p_6) \leq M(p_2)$$

$$M \geq \vec{0}$$

Note 4.1. *The inequalities derived in Theorem 4.2 may not always be necessary. Suppose that on one of the modules, say* N_1, *one of the following conditions holds:*

1. $P_t^1 \setminus P_{\overline{t}}^1 = \emptyset$. *Hence:*
$$0 = \sum_{p \in P_t^1 \setminus P_{\overline{t}}^1} M(p) \leq h_2 \sum_{p \in P_t^2} M(p)$$
is always verified.

2. $p_0^1 \in P_t^1$. *Here* p_0^1 *is the place initially marked in* N_1. *In this case there is a cycle containing* t *and* p_0^1, *and* t *may fire infinitely often in* N_1 *for each reachable marking. Hence:*
$$\sum_{p \in P_t^2 \setminus P_{\overline{t}}^2} M(p) \leq h_1 \sum_{p \in P_t^1} M(p)$$
is always verified.

Once the redundant inequalities in Theorem 4.2 are removed, as suggested by the previous note, the remaining inequalities may be rewritten in the form $A \cdot M \geq \vec{b}$. The same applies to the following Theorems 4.3, 4.4 and 4.5.

Let us consider the composition of two state machines N_1 and N_2 through a set T_s of transitions that are all looped in one of the nets, say N_2. Given the special structure of N_2, any reachable marking M_2 of N_2 may be reached without firing any transition in T_s. Hence it is never possible, as suggested by the previous note, that a firing sequence on N_2 may require the firing of more transitions in T_s than a firing sequence on N_1. Also, if a marking M_2' of N_2 may be reached by firing a transition in T_s, then any sequence of transitions in T_s may also be fired. Hence the only constraint imposed by the composition of the two modules is that for any marking $M = [M_1^T M_2^T]^T$, if reaching M_1 requires the firing of one or more transition in T_s, M_2 may be also be reached by firing a transition in T_s. Thus the next theorem holds (see also [10]).

Theorem 4.3. *When two state machines N_1 and N_2 are composed through a set T_s of transitions and these transitions are looped in one of the nets, say N_2, the set of markings reachable from the initial marking M_0 for the composed net $N = N_1 \parallel N_2$ is given by the following set of linear inequalities $\mathcal{A}(N)$:*

$$\mathcal{A}(N_1)$$

$$\mathcal{A}(N_2)$$

$$\sum_{p \in P^1_{T_s} \setminus P^1_{\overline{T}_s}} M(p) \leq h \sum_{p \in P^2_{T_s}} M(p)$$

where: $\mathcal{A}(N_i)$ $(\forall i = 1, 2)$, is the set of inequalities for the net N_i (as derived with Algorithm 3.1); the sets of places $P^i_{T_s}$ and $P^i_{\overline{T}_s}$ $(\forall i = 1, 2)$, belongs to N_i, and are determined as in Definition 4.2; h is equal to the number of tokens contained in N_1.

The following two theorems generalize Theorem 4.2 and Theorem 4.3 to the composition of two ECSM (not only state machines) along simple paths (not only single transitions). These theorems will be given without proof.

When two ECSM nets are composed along a simple path it is necessary to consider the possibility that the path may belong to more than one state machine module on each net. Also the places determined in Definition 4.2 are computed with respect to the first transition of the path.

Theorem 4.4. *Let N_1 and N_2 be two ECSM nets, i.e., $N_i = N_{i,1} \parallel \ldots \parallel N_{i,n_i}$ $(i = 1, 2)$, where $N_{i,j}$ is a state machine. Assume N_1 and N_2 are to be composed through a simple path of transitions $\theta = t_0 p_1 t_1 \ldots p_r t_r$ that belongs to modules $N_{1,q}$ $(q \in J_1)$ and to modules $N_{2,s}$ $(s \leq J_2)$. The set of markings reachable from the initial marking M_0 for the composed net $N = N_1 \parallel N_2$ is given by the following set of linear inequalities $\mathcal{A}(N)$:*

$$\mathcal{A}(N_1)$$
$$\mathcal{A}(N_2)$$

$$\sum_{p \in P_{t_0}^{1,q} \setminus P_{\overline{t}_0}^{1,q}} M(p) \leq h_2^{q,s} \sum_{p \in P_{t_0}^{2,s}} M(p) \qquad (q \in J_1,\ s \in J_2)$$

$$\sum_{p \in P_{t_0}^{2,s} \setminus P_{\overline{t}_0}^{2,s}} M(p) \leq h_1^{q,s} \sum_{p \in P_{t_0}^{1,q}} M(p) \qquad (q \in J_1,\ s \in J_2)$$

where: $\mathcal{A}(N_i)$ *is the set of inequalities for the net* N_i*; the sets of places* $P_t^{i,j}$ *and* $P_{\overline{t}}^{i,j}$ $(\forall i = 1, 2;\ \forall j \in J_i)$*, belongs to* $N_{i,j}$*, and are determined as in Definition 4.2;* $h_1^{q,s} = 1$ $(h_2^{q,s} = 1)$ *if there not exists a cycle containing the path* θ *in the net* $N_{1,q}$ $(N_{2,s})$*, else* $h_1^{q,s}$ $(h_2^{q,s})$ *is equal to the number of tokens contained in the net* $N_{2,s}$ $(N_{1,q})$*.*

When two ECSM nets are composed along k simple paths, the paths are looped in one of the net, hence they belong to only one state machine module of the looped net. However, it is necessary to consider the possibility that each path may belong to more than one state machine module on the net that is not looped.

Theorem 4.5. *Let* N_1 *and* N_2 *be two ECSM nets, i.e.,* $N_i = N_{i,1} \parallel \ldots \parallel N_{i,n_i}$ $(i = 1, 2)$*, where* $N_{i,j}$ *is a state machine. Assume* N_1 *and* N_2 *are to be composed through a* k *simple path of transitions* $\theta_j = t_0^j p_1^j t_1^j \ldots t_{r_j}^j$ $(j = 1, \ldots, k)$*, and let* $T_s = \{t_0^1, \ldots, t_0^k\}$*. Assume furthermore that path* θ_j *belongs to modules* $N_{1,q}$ $(q \in J_j)$ *and that all paths are looped in the module* $N_{2,1}$*. The set of markings reachable from the initial marking* M_0 *for the composed net* $N = N_1 \parallel N_2$ *is given by the following set of linear inequalities* $\mathcal{A}(N)$*:*

$$\mathcal{A}(N_1)$$
$$\mathcal{A}(N_2)$$

$$\sum_{p \in P_1'} M(p) \leq h \sum_{p \in P_2'} M(p)$$

where: $\mathcal{A}(N_i)$ *is the set of inequalities for the net* N_i*;*

$$P_1' = \bigcup_{j=1}^{k} \bigcup_{q \in J_j} \left(P_{t_0^j}^{1,q} \setminus P_{\overline{t}_0^j}^{1,q} \right)$$

is the set of places in N_1 *that may be marked only by firing a transition* $t \in T_s$*;*

$$P_2' = P_{T_s}^{2,1}$$

is the set of places in N_2 *that may be marked firing a transition* $t \in T_s$*;* h *is equal to the sum of the tokens contained in the nets* $N_{1,q}$ $(q \in J_j)$ $(j = 1, \ldots, k)$*.*

We conclude this section pointing out that when state machines modules are composed to form an ECSM, the number of inequalities of that defines the reachability set grows, in the worst case, more than linearly. In the case of Theorem 4.4 we have to add $2 \times |J_1| \times |J_2|$ inequalities.

5 Supervisor Validation

In this section the results developed so far are applied to the validation of supervisors for the control of discrete event systems (DES). We briefly review the basic notions of Supervisory Control theory that have some interest in the present exposition. For more details see [21].

In the Supervisory Control theory, originated by the work of Ramadge and Wonham [21], a discrete event system is simply a generator of a formal language, defined on an alphabet Σ. Two languages are associated with a DES: the *closed behavior* $L(= \overline{L} \subseteq \Sigma^*)$ is a prefix-closed language that represents the possible evolutions of the system[2]; the *marked behavior* $L_m(\subseteq L)$, that represents the evolutions corresponding to the completion of certain tasks.

Petri nets may be used as language generators in this framework. Given a marked net $\langle N, M_0 \rangle$, the alphabet Σ is represented by the set of transitions T. The closed behavior is given by the language $L(N, M_0) = \{\sigma \in T^* \mid (\exists M)\ [M_0\ [\sigma\rangle\ M]\}$. Given a set of final markings $\mathcal{M}_f$, the marked behavior is defined as $L_m(N, M_0) = \{\sigma \mid (\exists M \in \mathcal{M}_f)\ [M_0\ [\sigma\rangle\ M]\}$.

A DES is said to be *non-blocking* when $\overline{L_m} = L$, i.e., any string $\sigma \in L$ can be completed into a string $\sigma\tau \in L_m$.

The transitions in T are partitioned into two disjoint subsets: the set T_c of *controllable transitions* (that can be disabled if desired), and the set T_u of *uncontrollable transitions* (that cannot be disabled by an external agent). Let us assume, now, that the behavior of the system is to be restricted within the limits of a *specification language* by choosing controllable transitions to disable, i.e., to prevent from firing.

The agent which specifies which events are to be enabled and disabled when the system is in a given state, is called a *supervisor*. Standard techniques may be used to design a supervisor for a given control problem. Consider m discrete event systems, represented by the state machines $N_1, \ldots, N_m$, working concurrently. It is generally assumed that the set of transitions of all these systems are disjoint. The specifications to be enforced on the concurrent behavior of these systems are represented by n different state machines $H_1, \ldots, H_n$, whose transitions are a subset of all the transitions of the N's. The procedure for determining a *monolithic supervisor* [9, 22] requires the construction, by concurrent composition, of the net: $E = N_1 \parallel \ldots \parallel N_m \parallel H_1 \parallel \ldots \parallel H_n$. Note that there exists a set of final markings associated to the N's and H's. The set of final markings for E will be given by the cartesian product of the final markings of the modules, in the same way in which the initial marking for a composed net is computed in Definition 2.1.

[2]Here the bar represents the prefix-closure operator, i.e., $\overline{L}$ is the set of all prefixes of strings in L.

E and the nets N_i $(i = 1, \ldots, n)$ will run in parallel, i.e., whenever a transition fires on one of the nets N_i, it is also fired in E. Furthermore, the transitions enabled in E at a given marking represent the transitions that are allowed to fire in the the nets N_i, while all other transitions are disabled. The net E will represent a proper supervisor, if the following two properties are ensured. *Trimness:* the net E does not admit blocking markings, i.e., reachable markings from which a final marking cannot not be reached. *Controllability:* it is not possible to reach a marking from which an uncontrollable transition, belonging to the net N_i, is enabled in N_i but is not enabled in E. If E enjoys these properties it is called a monolithic supervisor, being at the same time a proper supervisor and a closed-loop model of the system under control.

In the following subsection it is discussed how these properties may be verified by Integer Programming techniques in the case that E is an ECSM net. Clearly these restrictions are heavily limiting the class of control problems that can be solved by our approach. However, it is possible to check for these properties in more efficient ways than by brute force state space search. Additionally, as will be also shown, it may be the case that the model may be validated by simple Linear Programming.

5.1 Blocking

Let $\langle N, M_0 \rangle$ be a marked net, and M_f be a final marking associated to it. For the sake of simplicity assume that M_f is the only final marking of the net. The net N will be blocking iff

$$(\exists M)\ [M \in R(N, M_0), M_f \notin R(N, M)]$$

Now the set $R(N, M_0)$ of a ECSM net can be given as a convex linear set, as shown in Section 4. Similarly the *coreachability set* of M_f, i.e., the set $\{M \mid M_f \in R(N, M)\}$, can be given in this form. In fact, given a marking M_f of $N = (P, T, I, O)$, the coreachability set of M_f is identical to the reachability set from M_f in the reversed net $N^R = (P, T, O, I)$, that is $\{M \mid M_f \in R(N, M)\} = R(N^R, M_f)$.Thus, it is possible to check for the existence of blocking markings as follows.

Proposition 5.1. *Let $\langle N, M_0 \rangle$ be a marked ECSM net and let M_f be the final marking associated to it. Assume the reachability set of M_0 is given by the set of inequalities: $A_0 \cdot M \geq \vec{b}_0$, and the coreachability set of M_f is given by the inequalities: $A_f \cdot M \geq \vec{b}_f$, where:*

$$A_f = \begin{bmatrix} \vec{a}_{f,1}^T \\ \ldots \\ \vec{a}_{f,k}^T \end{bmatrix}$$

Then there exist a blocking marking M if and only if one or more of the following Con-

straint Sets admit an integer feasible solution $(\forall i = 1, \ldots, k)$*:*

$$\begin{array}{rr} A_0 \cdot M \geq \vec{b}_0 & \text{CS1}_\text{i} \\ \vec{a}^T_{f,i} \cdot M \leq \vec{b}_f(i) - 1 & \end{array}$$

Proof: M is a blocking marking $\iff M \in R(N, M_0) \wedge M_f \notin R(N, M) \iff [A_0 \cdot M \geq \vec{b}_0] \wedge \neg[A_f \cdot M \geq \vec{b}_f] \iff [A_0 \cdot M \geq \vec{b}_0] \wedge [\bigvee_{i=1}^{k} \vec{a}^T_{f,i} \cdot M < \vec{b}_f(i)] \iff (\exists i)[(A_0 \cdot M \geq \vec{b}_0) \wedge (\vec{a}^T_{f,i} \cdot M \leq \vec{b}_f(i) - 1)]$. $\diamond$

Note that each constraint $\vec{a}^T_{f,i} \cdot M < \vec{b}_f(i)$ has been rewritten in the equivalent form $\vec{a}^T_{f,i} \cdot M \leq \vec{b}_f(i) - 1$.

It is possible to relax the constraint that the vector M be integer and obtain a sufficient condition for the validation of the net. In fact, if no real vector M satisfies the previous systems of inequalities no blocking marking may be reached.

5.2 Controllability

Consider the net E constructed by concurrent composition of all the systems and specification modules. Let $t_q \in T_u$ be an uncontrollable transition and assume that t_q belongs to the system net N_i. (By the hypothesis that all the systems have disjoint transitions, a transition may belong to only one net N, although it may belong to more than one specification net H.) Let the preset of t_q be: ${}^\bullet t_q = \{p^0_q, \ldots, p^{k_q}_q\}$, where p^0_q is a place of N_i and p^j_q $(j > 0)$ is a place of some specification net H.

E is not controllable if and only if it is possible to reach a marking M such that an uncontrollable transition t_q is enabled by $\mathcal{P}_i(M)$ in N_i, but it is not enabled by M in E. In other words, E is not controllable iff

$$(\exists t_q \in T_u)\ (\exists M)\ [M \in R(E, M_0) \wedge M(p^0_q) \geq 1 \wedge (\bigvee_{j=1}^{k_q} M(p^j_q) \leq 0)]$$

Proposition 5.2. *Assume the reachability set of the marked ECSM net* $\langle E, M_0 \rangle$ *is given by the set of inequalities:* $A_0 \cdot M \geq \vec{b}_0$*. Then* E *will not be controllable if and only if one or more of following Constraint Sets admit an integer feasible solution* $\forall t_q \in T_u$ *and* $\forall p^j_q \in {}^\bullet t_q$ $(j > 0)$

$$\begin{array}{rr} A_0 \cdot M \geq \vec{b}_0 & \text{CS2}_\text{q,j} \\ M(p^0_q) \geq 1 & \\ M(p^j_q) \leq 0 & \end{array}$$

Proof: E is not controllable $\iff (\exists t_q \in T_u)\ (\exists M)\ [M \in R(E, M_0) \wedge M(p_q^0) \geq 1 \wedge (\bigvee_{j=1}^{k_q} M(p_q^j) \leq 0)] \iff (\exists t_q \in T_u)\ (\exists M)\ (\exists j)\ [A_0 \cdot M \geq \vec{b}_0 \wedge M(p_q^0) \geq 1 \wedge M(p_q^j) \leq 0]$. $\diamond$

If the constraint that M be integer is relaxed, we may use Linear Programming to derive a sufficient condition for E to be controllable.

Note also that a *semidecision procedure* (only sufficiency) for controllability may be given for a larger class of nets than ECSM in terms of the set $PR(N, M_0)$. We will state this in the next proposition.

Proposition 5.3. *Let $\langle E, M_0 \rangle$ be a net constructed as concurrent composition of state machines (E needs not be an ECSM). Then E will be controllable if no one of the following Constraint Sets admits an integer feasible solution $\forall t_q \in T_u$ and $\forall p_q^j \in {}^\bullet t_q$ $(j > 0)$*

$$
\begin{array}{cc}
M = M_0 + C \cdot \vec{\sigma} & \mathrm{CS2'_{q,j}} \\
M(p_q^0) \geq 1 & \\
M(p_q^j) \leq 0 & \\
M \geq \vec{0} & \\
\vec{\sigma} \geq \vec{0} &
\end{array}
$$

where C is the incidence matrix of E.

6 Discussion

6.1 Complexity Issues

The complexity of the decision procedure for supervisor validation has been discussed in [21]. It was shown that if both plant and specification are regular, then controllability is *polynomially decidable in the number of states*. However, it was also observed that in general the *number of states grows exponentially in the number of constituent systems*. This means that state space search is computationally unfeasible.

Petri net models give a compact description of concurrent systems, in the sense that the structure of a net grows linearly with the number of constituent systems. We have also seen that in our approach the size of the constraint set describing the set of reachable markings of a ECSM net grows in the worst case quadratically with the number of constituent systems.

The complexity of Integer Programming techniques, however, is an open problem. It is doubtful that these problems have polynomial complexity in the size of the constraint set. However, as suggested by other authors [18], their tractability and efficiency have been

demonstrated in practice. Thus this approach is to be preferred to state space search methods.

There also are two potential advantages in our approach.

- Linear Programming may be sufficient to validate the model. In this case, we may simply use the simplex method whose complexity is *almost linear* in the size of the constraint set.

  In fact, we suggest that at first a possibly non-integer solution should be computed. Subsequently, should the constraint set admit a non-integer solution, an integer one should be searched, using the cutting plane method, for instance.

- The method can be extended to systems with infinite state space, as we will discuss in the next section. These systems are obviously not tractable with brute state space search.

6.2 Model Extension

Unfortunately we have no results yet on the "number" of supervisors that can be modeled with our approach. Clearly not all possible supervisors of interest fall into the class of ECSM. We can only say that the two types of compositions used to define ECSM nets are the primitives to model one point rendezvous and shared resources.

Although we realize that our approach can be used with other nets than ECSM, we have not yet been able to formally define a superset of ECSM for which there exists an algorithm to compute the constraints defining the reachability set.

We will discuss here, with two examples, possible ways of generalization. Firstly, it should be possible to consider the composition of nets other than state machines. Secondly, it should be possible to consider types of compositions other than those used to define ECSM.

Example 6.1. *The net in Figure 3 has an infinite reachability set, that can be described by the equations:*

$$M(p_1) + M(p_3) = 1$$

$$M \geq \vec{0}$$

Additionally, the firing bounds are the following: $\rho_{\min}(M, t_1) = M(p_2)$; $\rho_{\max}(M, t_1) = M(p_2)$ **if** $M(p_3) = 0$ **else** $\rho_{\max}(M, t_1) = \infty$; $\rho_{\min}(M, t_2) = M(p_3)$; $\rho_{\max}(M, t_2) = M(p_3)$; $\rho_{\min}(M, t_3) = 0$; $\rho_{\max}(M, t_3) = 0$ **if** $M(p_3) = 0$ **else** $\rho_{\max}(M, t_3) = \infty$.

The results of Theorem 4.2 and Theorem 4.3 apply to any net, such as the one of the previous example, that can be completely characterized in terms of reachability set and firing bounds. Thus, the example shows that even unbounded nets, i.e., nets with an infinite state space, may be composed following the approach presented in this paper.

Example 6.2. *The net shown in Figure 4.(a) has been composed by concurrent composition of three modules:* $N_1 = (\{p_3, p_6\}, \{t_2, t_3\}, I_1, O_1)$; $N_2 = (\{p_1, p_2, p_3, p_4\}, \{t_1, t_2, t_3, t_4\}, I_2, O_2)$; $N_2 = (\{p_2, p_4, p_5\}, \{t_1, t_2, t_3, t_4\}, I_3, O_3)$. *This net is not an ECSM. In fact, the composition of the first and third net is not one of the two types of compositions used in Definition 4.1. However, the state space of the composed net may still be represents by a set linear inequalities. We just have to add to the inequalities of each component net:*

$$M(p_3) + M(p_6) = 1$$

$$M(p_1) + M(p_2) + M(p_3) + M(p_4) = 2$$

$$M(p_2) + M(p_4) + M(p_5) = 1$$

$$M \geq \vec{0}$$

a new inequality:

$$M(p_2) + M(p_5) + M(p_6) \geq 1$$

Note that $\{p_2, p_5, p_6\}$ *is a new trap created by the composition of the three nets. The reachability graph of the composed net is shown in Figure 4.(b).*

This example highlights the possibility of generalizing this approach to some compositions of nets sharing any *two transitions*.

Other types of compositions are also possible. As an example, one may refine a transition or a place along a shared simple path, substituting it with a more complex structure such as a *macrotransition* or *macroplace* net, as defined in [7]. This is equivalent to consider the sharing of complex subsystems between nets, rather than simple paths.

7 Conclusions

A class of P/T net, called Elementary Composed State Machine nets has been defined. The reachability problem for this class can be solved by deriving a set of linear inequalities that exactly define the set of reachable markings. Important properties of the net, such as the absence of blocking states or controllability, may be studied by Integer Programming techniques. This approach may be used to the validation of supervisors for the control of discrete event systems.

The main drawbacks of the approach presented in this paper may be summarized as follow:

- Integer Programming problems, although more tractable than methods based on brute state space search, may not always be solved in polynomial time. However, the paper also shows that it is possible to use Linear Programming techniques to derive sufficient conditions for supervisory validation.
- The model is limited, in the sense that there exist monolithic supervisors that cannot be modeled as ECSM nets. We have discussed possible ways of generalization.
- Although a procedure to validate a supervisor is given in this paper, the control problem is not directly solved, in the sense that if the model does not have the desired properties the approach does not directly lead to the construction of a proper supervisor.

References

[1] G.S. Avrunin, U.A. Buy, J.C. Corbett, L.K. Dillon, J.C. Wileden, "Automated Analysis of Concurrent Systems with the Constrained Expression Toolset," *IEEE Trans. Software Engineering*, Vol. 17, No. 11, pp. 1204–1222, November, 1991.

[2] G. Berthelot, "Transformations and Decompositions of Nets," *Petri Nets: Central Models and Their Properties, Advances in Petri Nets 1986*, W. Brauer, W. Reisig and G. Rosenberg (eds.), Lecture Notes in Computer Sciences, Vol. 254-I, pp. 359–376, Springer-Verlag, 1987.

[3] B. Berthomieu, "Methods for Carrying Proofs on Petri Nets Using their Structural Properties," *Technical Report*, Laboratoire d'Automatique et d'Analyse des Systèmes du CNR (Toulouse, France), January, 1987.

[4] E. Best, C. Fernández, "Notation and Terminology on Petri Net Theory," *Arbeitspapiere der Gesellschaft für Math. und Datenverarbeitung*, No. 195, 1986.

[5] J.M. Colom, "Análisis Estructural de Redes de Petri, Programación Lineal y Geometria Convexa," *Tesis Doctoral*, Universidad de Zaragoza (Zaragoza, Spain), 1989.

[6] F. De Cindio, G. De Michelis, L. Pomello, C. Simone, "Superposed Automata Nets," *Application and Theory of Petri Nets*, C. Girault and W. Reisig (eds.), Informatik-Fachberichte, Vol. 52, pp. 269–279, Springer-Verlag, 1982.

[7] A. Desrochers, H. Jungnitz, M. Silva, "An Approximation Method for the Performance Analysis of Manufacturing Systems Based on GSPNs," *Proc. 3rd Int. Conf. on Computer Integrated Manufacturing* (Troy, New York), pp. 46–54, May, 1992.

[8] A. Giua, "Petri Nets as Discrete Event Models for Supervisory Control," *Doctoral Thesis*, Rensselaer Polytechnic Institute (Troy, New York), 1992.

[9] A. Giua, F. DiCesare, "Supervisory Design Using Petri Nets," *Proc. 30th IEEE Int. Conf. Decision and Control* (Brighton, England), pp. 385–390, December, 1991.

[10] A. Giua, F. DiCesare, "A Class of Petri Nets with a Convex Reachability Set," *Proc. 1993 IEEE Int. Conf. on Robotics and Automation* (Atlanta, Georgia), pp. 578–583, May, 1993.

[11] A. Giua, F. DiCesare, M. Silva, "Petri Nets Supervisors for Generalized Mutual Exclusion Constraints," *Proc. 1993 IFAC World Congress* (Sidney, Australia), Vol. 1, pp. 267–270, July, 1993.

[12] M. Hack, "Extended State Machine Allocatable Nets, an Extension of Free Choice Petri Nets Results," *Computation Structures Group Memo 78-1*, Project MAC, Massachusetts Institute of Technology (Cambridge, Massachusetts), 1974.

[13] L.E. Holloway, B.H. Krogh, "Synthesis of Feedback Control Logic for a Class of Controlled Petri Nets," *IEEE Trans. on Automatic Control*, Vol. AC-35, No. 5, pp. 514–523, May, 1990.

[14] A. Ichikawa, K. Hiraishi, "Analysis and Control of Discrete Event Systems Represented by Petri Nets," *Discrete Events Systems: Models and Applications*, P. Varaiya and A.B. Kurzhanski (eds.), Lecture Notes in Control and Information Sciences, Vol. 103, pp. 115–134, Springer-Verlag, 1988.

[15] A. Ichikawa, K. Hiraishi, "A Class of Petri Nets That a Necessary and Sufficient Condition for Reachability is Obtainable," *Trans. Society of Instrument and Control Engineers, SICE* (in Japanese), Vol. 24, No. 6, 1988.

[16] M. Jantzen, R. Valk, "Formal Properties of Place/Transition Nets," *Net Theory and Applications*, W. Brauer (ed.), Lecture Notes in Computer Sciences, Vol. 84, pp. 165–212, Springer-Verlag, 1980.

[17] C. Johnen, "Analyse Algorithmique des Réseaux de Petri: Verification d'Espace d'Accueil, Systemès de Réécriture," *Thèse Doctoral*, Université Paris–Sud (Paris, France), 1987.

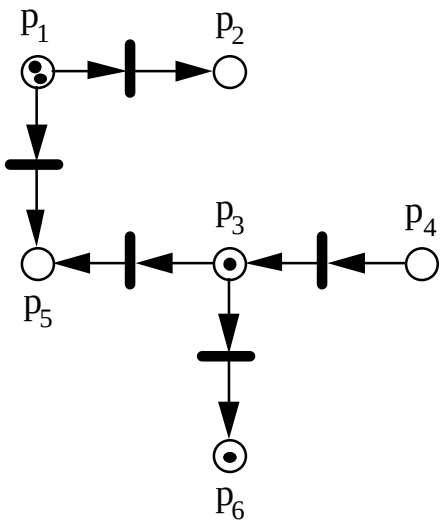

Figure 1: State machine in Example 3.1.

[18] Y. Li, W.M. Wonham, "Linear Integer Programming Techniques in the Control of Vector Discrete-Event Systems," *Proc. 27th Annual Allerton Conf. on Communication, Control and Computing*, University of Illinois, pp. 528–537, September, 1989.

[19] Y. Li, W.M. Wonham, "Control of Vector Discrete-Event Systems I — The Base Model," *IEEE Trans. on Automatic Control*, Vol. AC-38, No. 8, pp. 1214–1227, August, 1993.

[20] T. Murata, " Petri Nets: Properties, Analysis and Applications," *Proceedings IEEE*, Vol. PROC-77, No. 4, pp. 541–580, April, 1989.

[21] P.J. Ramadge, W.M. Wonham, "The Control of Discrete Event Systems," *Proceedings IEEE*, Vol. PROC-77, No. 1, pp. 81–98, January, 1989.

[22] W.M. Wonham, "A Control Theory for Discrete-Event Systems," *Advanced Computing Concepts and Techniques in Control Engineering*, M.J. Denham and A.J. Laub (eds.), Springer-Verlag, pp. 129–169, 1988.

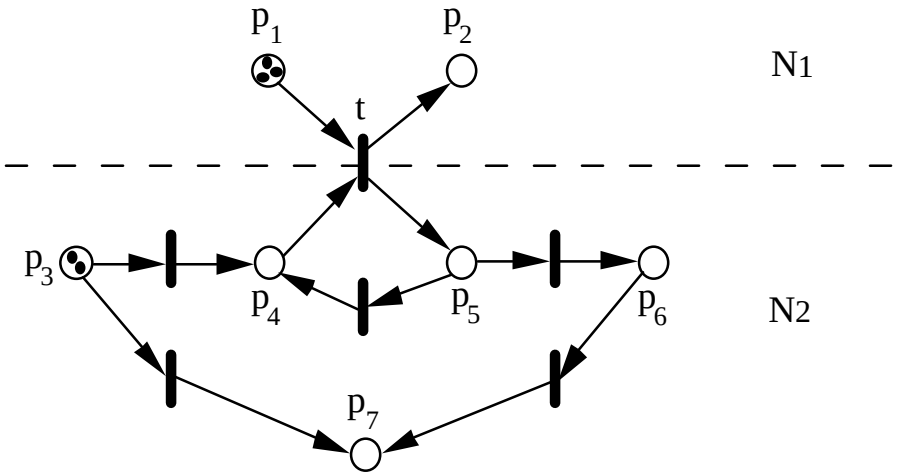

Figure 2: ECSM net in Example 4.1.

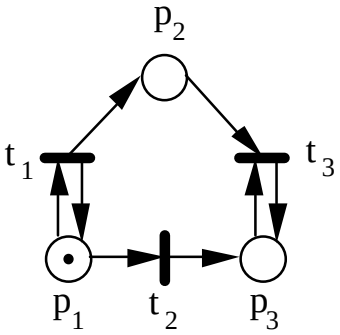

Figure 3: Unbounded net in Example 6.1.

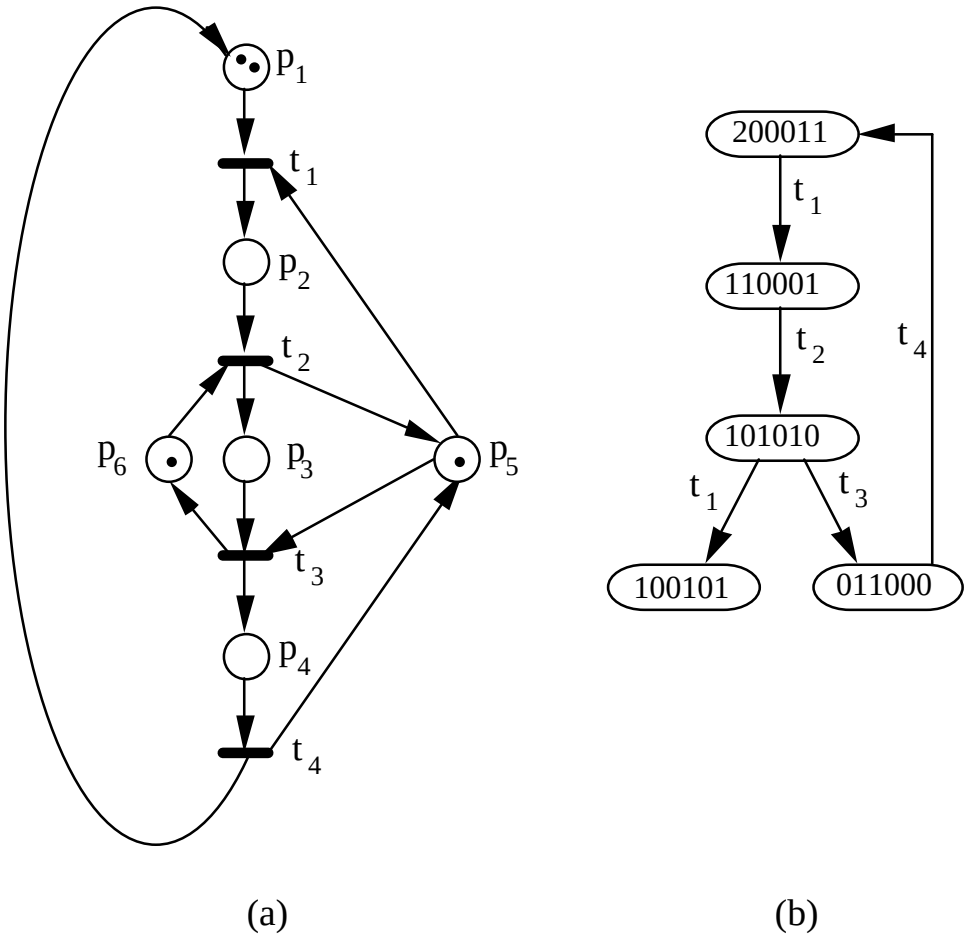

Figure 4: Composed net and its reachability graph in Example 6.2.