A Petri Net Based Deadlock Prevention Policy for Flexible Manufacturing Systems

Joaquín Ezpeleta, José Manuel Colom, and Javier Martínez

***Abstract*—In this paper we illustrate a compositional method for modeling the concurrent execution of working processes in flexible manufacturing systems (FMS) through a special class of Petri Nets that we call S^3PR. In essence, this class is built from state machines sharing a set of places modeling the availability of system resources. The analysis of S^3PR leads us to characterize deadlock situations in terms of a zero marking for some structural objects called siphons. In order to prevent the system from deadlocks, we propose a policy for resource allocation based on the addition of new places to the net imposing restrictions that prevent the presence of unmarked siphons (direct cause of deadlocks). Finally, we present the application of this technique to a realistic FMS case.**

***Index Terms*—Petri net models. Sequential processes. Deadlock prevention strategies. Flexible Manufacturing Systems.**

I. Introduction

The present paper fits in the modeling and analysis of Flexible Manufacturing Systems (FMS). In general, an FMS is structured as a set of workstations, where products must be processed, and a flexible transport system, the goal of which is to load and unload the workstations. An FMS is built for the manufacturing of a set of different types of products. Every product follows a route through the set of system resources, according to a preestablished working plan. The sequence of operations performed in order to manufacture a product is what we call a *working process* (WP). In a WP we distinguish the execution states. Every state groups a set of operations using the same set of resources (in the present work, we restrict to one the number of resources used at each state). A state of a WP can be reached, from a previous one, when the resource used by the operations performed in it is available. On the other hand, alternative sequences are allowed in a WP. By a system resource we mean an element of the system that is able to hold a product (for transport, operation, storage, quality control). The working processes in a FMS are executed concurrently, and therefore, they have to compete for the set of common resources. These relations of competition can cause deadlocks. Roughly speaking, a deadlock is a system state so that some working processes can never be finished. In our context, a deadlock situation is due to a wrong resource allocation policy. In fact, behind a deadlock problem there is a circular wait situation for a set of resources.

When deadlock situations can arise in a system, it is important to characterize them in order to avoid the system to reach them (*deadlock prevention/avoidance problem*) or to recover the system from such situations (*deadlock recovery problem*).

We shall focus our attention on the deadlock prevention/avoidance problem. The goal of these approaches (prevention and avoidance) to the deadlock problem is to add to the system a control policy preserving the system from deadlock situations. But the way both approaches deal with the problem is different. The deadlock prevention approach establishes the control policy in a static way, so that, once established, we are sure that the system cannot reach undesirable deadlock situations. In [10], [17], [5], [6] different approaches of this kind may be found. The deadlock avoidance approach is different: at each system state, the control policy determines (on-line) which system evolutions, among the set of feasible ones, are the correct. In [17], [9], [2] solutions of this kind have been adopted.

In our approach we have adopted Petri nets as a tool for modeling the dynamic behavior of the system. This tool has also been adopted in several papers related to the study of deadlock problems in FMS environments [17], [2], [9], [6]. For a general class of Petri net models, in [17] both prevention and avoidance control policies are proposed. The first one is based on the net reachability graph, while the second one is based on a look-ahead procedure that searches for deadlock situations by simulating the system evolution for a preestablished number of steps. Due to the fact that the avoidance policy does not assure that deadlocks are not reachable, they propose to combine this policy with a deadlock recovery system. In [2] a deadlock avoidance algorithm is proposed for a class of Petri net models formed by a set of sequential processes (without alternatives in its execution) that use a resource in each state. The algorithm controls the input of new tokens in a model "zone", assuring that system evolutions are always possible. For the same class of models, Hsieh and Chang propose in [9] a different deadlock avoidance control policy based on the concept of Minimal Resource Requirement (minimal number of resources assuring the existence of a system evolution that allows to complete all the jobs in the system).

The Petri net models that we obtain from our systems belong to a particular class of nets that we call Systems of Simple Sequential Processes with Resources (S^3PR). This class of models is a generalization of the one used in [2],

Manuscript received October 30, 1992; revised July 27, 1993. This work was partially supported by the Spanish "Comisión Interministerial de Ciencia y Tecnología" (CICYT), project TIC-91/0354 and the Aragonese "Consejo Asesor de Investigación" (CONAI), project P IT-6/91.

The authors are with the Departamento de Ingeniería Eléctrica e Informática, Centro Politécnico Superior, Universidad de Zaragoza, María de Luna, 3, 50015 Zaragoza, Spain.

IEEE Log Number 9409081.

1042-296X/95$04.00 © 1995 IEEE

[9] since, considering that the use of resources is made in the same way, our working processes allow choices in their executions. In the present paper we study some properties of S^3PR and we give a characterization of the liveness in terms of structural Petri net items (siphons). The liveness of a system means that each system action can be made in the future, no matter what system state has been reached. This result about S^3PR model analysis is the starting point for the definition of a control policy whose goal is the (total and partial) deadlock prevention. This control policy can be implemented by adding some new net elements (places and related arcs) to the initial S^3PR model. The intensive use of information from the net structure is one of the main differences with previous works in the literature on the topic of deadlock prevention/avoidance.

From the system model designer point of view, the modeling methodology resulting from the approach proposed in this paper consists of three phases: 1) Modeling of the FMS in terms of Petri nets. 2) Off-line analysis of the resulting S^3PR in order to establish the control policy preventing deadlocks in the system. The proposed control policy is also implemented in terms of Petri net elements. 3) Automatic code generation for the controlled Petri net model in order to establish the on-line system control.

The rest of the paper is organized as follows. In Section II we present, in an intuitive way, how to model WP's sharing a set of resources in a FMS. The resulting Petri net models belong to the class of S^3PR. In Section III we recall the definitions of the main concepts related to Petri nets. The class of S^3PR is defined in a formal way in Section IV, where some interesting properties are shown. Some results on liveness analysis for this class of nets are presented in Section V. The definition and the correctness proof of a deadlock prevention control policy for S^3PR is shown in Section VI. Section VII introduces an example of a flexible manufacturing system and illustrates the application of the previous control policy. Finally, some conclusions are presented in Section VIII.

II. An Intuitive Approach to a Class of Petri Net Models For FMS

In this section we introduce, in an intuitive way, some of the main concepts that will be used later on.

The modeling of working processes: We have adopted Petri nets to model the dynamic behavior of the working processes. The use of the Petri net analysis theory will give us the techniques for checking interesting properties about the good behavior of the system and also some "hints" on how to avoid non desirable situations. Fig. 1(b) shows a Petri net model of a working process corresponding to the manufacturing of a product in the robotized cell shown in Fig. 1(a). The model has six different states $\{is, R.M1.M2, inM1, inM2, toOB, fs\}$ (a state is modeled by means of a place, represented by a circle) and six transitions modeling the changes between states (a transition is represented by means of a box). In the model, the description of the operations to be performed at each state has been omitted because this information is not relevant for the system control at the level of the resource allocation problem. States is and fs are considered as the "initial state" (the process has not started) and the "final state" (the process is finished).

In the previous model the resources used in the working plan execution are not represented. They can be modeled by means of places, the marking of which model the availability of the resource. In Fig. 1(c) the model of the working process in Fig. 1(b) is completed with the resource places used by the WP (places $M1$, $M2$ and $M3$). The marking of $M1, M2$ and R models availability of both machines and the robot, respectively (we assume that each resource can hold only one product at a time).

Let us now specify which class of models and working processes we have considered. The constraints for these models are the following:

1) A working process describes the set of possible sequences of operations the system has to perform in order to manufacture a product.
2) A working process has an initial and a final state.
3) Choices are allowed in a working process, but iterations are not. However, if the number of iterations is a previously known constant, we can construct an equivalent sequence, as depicted in Fig. 2.
4) Only one shared resource is allowed to be used at each state in a working process. The resource used in a state is released when the system moves to a next state. Two adjacent states cannot use the same resource.
5) Initial and final states do not use resources.

We can see that the model of a working process is a state machine plus a set of places modeling availability of resources. We call these places *resources*. For instance, in Fig. 1(c), places $M1, M2$ and R are resources. Taking into account the constraints imposed on the FMS under consideration, in Petri net terminology, a resource is a structural implicit place [3]. This means that if we have an arbitrarily large number of resources (i.e., the number of tokens in places representing resources is arbitrarily large), the marking of these places does not limit the concurrent processing of products, and then, these places can be removed (because they become implicit places).

At a given moment, in an FMS several identical processes can be executed concurrently. This fact can be modeled by means of a unique Petri net model for each type or family of identical processes, allowing this model to have as many tokens as instances of the identical processes being in execution. Each token models the execution of one process. For a working process, the number of processes (products) that can be concurrently executed (manufactured) depends on the capacity of the resources that they need to use. In order to model this feature, we can "collapse" the initial state and the final state places of the same working process model, so as to have "cyclic models". The new place generated will be called the "idle state" place. Therefore, we can interpret the initial marking of the idle place as the maximum number of products of the corresponding working plan that are allowed to be concurrently manufactured in the system (this number is determined by the system resource capacity). In an FMS several WP's can operate concurrently. In this case, the model

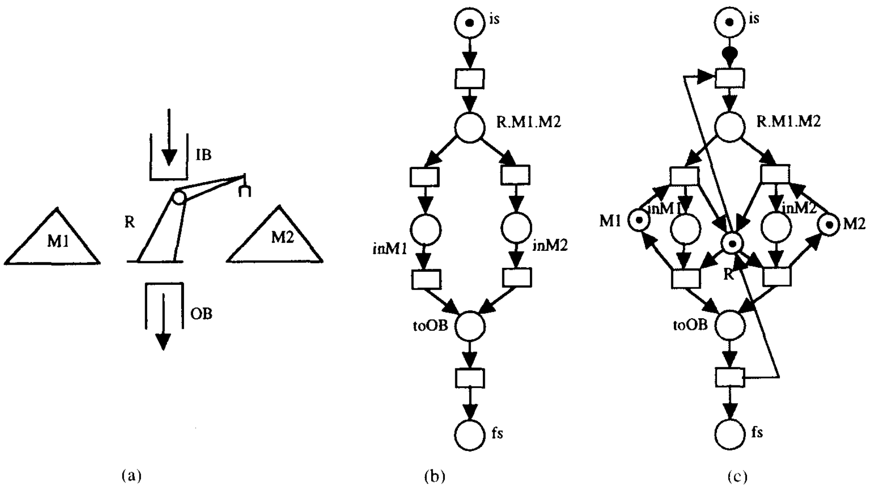


Fig. 1. (a) A robotized cell. (b) Petri net modeling a working process so that a product is manufactured either in machine $M1$ or in machine $M2$. (c) Final model with the resource capacity constraints.

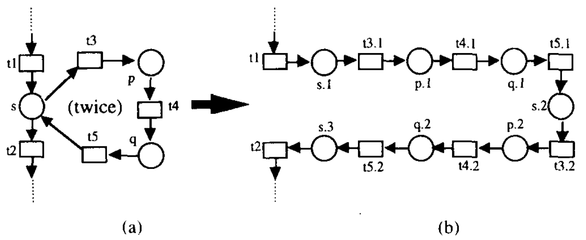


Fig. 2. Finite iterations can be modeled without cycles.

of the global system is obtained from the models of each WP by fusion of the places modeling the same resources. The final initial marking of each one of these common resources will be the maximal of the markings that they have in each WP model (we assume that each model is correct). The competition relations among several WP's are modeled by the interaction on the common places.

Deadlocks and liveness: In a production system, a set of processes are executed concurrently and they share a set of common resources. Fig. 3(c) shows a model of a system where two types of working processes are executed. Places $r1, r2, r3, r4, r5$ model availability of resources. The global model is obtained by fusion of the common places in models in figures 3-a and b. In order to have a correct system behavior, it is desirable that each production order can finish; i.e., we have to impose that each process can reach its final state (places p and p' in Fig. 3). However, an incorrect control in the execution of the working processes can lead to deadlock situations, in the sense that a set of processes, at a given state, can never reach the final state. Let us consider, for instance, a state of the system in Fig. 3(c) so that there are two tokens (products) in place b and one in place b'. It is clear that none of them can progress due to the fact that the resources they need to progress have been allocated and they are not available. A circular wait for resources $r2$ and $r3$ arises.

Let us now focus on the liveness in Petri net terminology. Liveness means that, for every reachable state, the model can evolve in such a way that every transition can always be fired in the future, or, in other words, every system activity (modeled by means of a transition) can ultimately be carried out. Translating these ideas to the FMS domain, the liveness property means that every production process can always be finished and that it is always possible to introduce new products in the system to be manufactured.

Deadlock control policy: Now, the question is as follows: What can we do when the model of our working processes is not live? In these cases, a control policy ensuring that each working process may finish will be added to the model. This control policy will constrain the system behavior to a set of states so that, whichever state the system reaches, there is always a system evolution so that the treatment of each product can reach its final state.

III. Basic Petri Net Definitions

In this section, the main definitions related to Petri net models are introduced in a very compact way. For a complete study of this subject, the reader is referred to [15], [13], [12].

Petri nets: A Petri net (or Place/Transition net) is a 3-tuple $\mathcal{N} = \langle P. T. F \rangle$where P and T are two nonempty disjoint sets, called *places* and *transitions*. The set $F \subseteq (P \times T) \cup (T \times P)$ is the *incidence (flow) relation*. Given a net $\mathcal{N} = \langle P. T, F \rangle$ and a node $x \in P \cup T$, $^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ is the *preset* of x, while $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is the *post-set* of x. This notation is extended to a set of nodes as follows: given $X \subseteq P \cup T$. $^\bullet X = \cup_{x \in X} {}^\bullet x$, $X^\bullet = \cup_{x \in X} x^\bullet$. A self-loop free Petri net $\mathcal{N} = \langle P. T. F \rangle$ can alternatively be represented as $\mathcal{N} = \langle P, T. C \rangle$ where C is the net *flow matrix*: a $P \times T$ integer matrix so that $C = C^+ - C^-$ where $C^+[p. t] =$ *If* $(t, p) \in F$ *then* 1 *else* 0; $C^-[p, t] =$ *If* $(p. t) \in F$ *then* 1 *else* 0

A *marking* is a mapping $m : P \longrightarrow \mathbb{N}$; in general, we will use the multi-set notation for markings: $m = \sum_{p \in P} m(p).p$. When talking about a set of places $S \subseteq P$, $m(S) = \sum_{p \in S} m(p)$. The pair $\langle \mathcal{N}. m_0 \rangle$, where $\mathcal{N}$ is a net and m_0

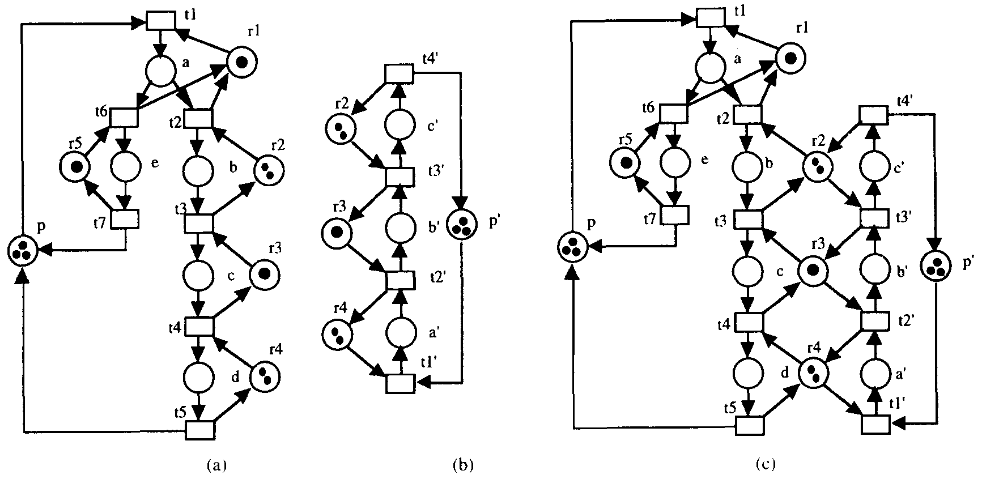


Fig. 3. (a) and (b) are two marked S^2PR. (c) The S^3PR corresponding to the concurrent execution of processes (a) and (b).

is an (initial) marking, is called a *marked Petri net* (or net system). A transition $t \in T$ is *enabled* at a marking m iff $\forall\, p \in {}^\bullet t,\ m(p) > 0$; this fact will be denoted as $m[t\rangle$; when fired (in the usual way), this gives a new marking m'; this will be denoted as $m[t\rangle m'$. A marking m' is *reachable* from another one m iff there exists a firing sequence $\sigma = t_1 t_2 ... t_n$ so that $m[t_1\rangle m_1[t_2\rangle m_2 ... m_{n-1}[t_n\rangle m'$. This fact is denoted as $m[\sigma\rangle m$; the set of markings reachable from m in $\mathcal{N}$ is denoted as $\mathcal{R}(\mathcal{N}, m)$.

Some structural objects and properties of a marked Petri net: A set of places $S \subseteq P$ is a *siphon* iff ${}^\bullet S \subseteq S^\bullet$, and it is a *trap* iff $S^\bullet \subseteq {}^\bullet S$. A net system $\langle \mathcal{N}, m_0 \rangle$ is *deadlock-free* iff $\forall\ m \in \mathcal{R}(\mathcal{N}, m_0),\ \{t \in T \mid m[t\rangle\} \neq \emptyset$. A transition t is *live* iff $\forall\ m \in \mathcal{R}(\mathcal{N}, m_0),\ \exists\ m' \in \mathcal{R}(\mathcal{N}, m)$ so that $m'[t\rangle$. A marked net is live iff every transition is live. We will say that a transition t is *dead for a reachable marking* m iff there is no reachable marking from m that enables t. Notice that a transition is live iff it is not dead for each reachable marking. A *P-semiflow* Y is a P-indexed vector so that $\forall\ p \in P,\ Y(p) \in \mathbb{N}$ and $Y^T.C = 0$, whereas a *T-semiflow* X is a T-indexed vector so that $\forall\ t \in T,\ X(t) \in \mathbb{N}$ and $C.X = 0$. We call the *support* of a vector Y (respectively, X) the set $\|Y\| = \{p \in P \mid Y[p] \neq 0\}$ $(\|X\| = \{t \in T \mid X[t] \neq 0\})$. If there exists a P-semiflow $Y > 0$ (T-semiflow $X > 0$) so that $P = \|Y\|$ $(T = \|X\|)$, the net is said to be *conservative* (*consistent*).

Subnets: Let $\mathcal{N} = \langle P. T. F \rangle$ be a net, and let $X \subseteq P \cup T$. Then, X *generates* the subnet $\mathcal{N}_X = \langle P_X . T_X , F_X \rangle$ where $P_X = P \cap X . T_X = T \cap X . F_X = F \cap (X \times X)$. A subnet $\mathcal{N}_X$ is called a P-subnet iff $X = P_1 \cup {}^\bullet P_1 \cup P_1{}^\bullet$, where $P_1 \subseteq P$. A strongly connected subnet $\mathcal{N}_X$ of a net $\mathcal{N}$ is called a *P-component* of $\mathcal{N}$ iff $\forall t \in T_X : \mid {}^\bullet t \cap P_X \mid = 1$ and $\mid t^\bullet \cap P_X \mid = 1$ (presets and postsets are taken with respect to $\mathcal{N}$). A string $x_1 . .... x_n$ is called a *path* of $\mathcal{N}$ iff $\forall i \in \{1. ...., n - 1\} : x_{i+1} \in x_i{}^\bullet$. A *simple path* is a path whose nodes are all different (except, perhaps, x_1 and x_n). A path $x_1, ..., x_n$ is called a *circuit* iff it is a simple path and $x_1 = x_n$.

IV. The Class of the S^3PR Models

The definition of the class of models resulting from the kind of FMS under consideration is made in a constructive way. First of all, we define the class of the Simple Sequential Processes (S^2P); then, we extend it to model the use of resources (the class of S^2PR) and, finally, we define the class of Systems of Simple Sequential Processes with Resources (S^3PR) by net composition of S^2PR via a set of common places.

Definition IV.1: A *Simple Sequential Process*(S^2P) is a Petri net $\mathcal{N} = \langle P \cup \{p^0\}. T. F \rangle$where: 1) $P \neq \emptyset, p^0 \notin P$ (p^0 is called the process idle place); 2) $\mathcal{N}$ is a strongly connected state machine and 3) every circuit of $\mathcal{N}$ contains the place p^0.

The third condition imposes a property of "termination" to the working processes that we are considering: if a process evolves then it will finish.

This definition of a (simple) sequential process is nearer to the definition of "free sequential system" in [10] or "job subnet" in [9] than to the definition of "sequential process" in [16] (that includes the set of buffers used by the process, and that is a generalization of the notion of "sequential machine" in [14]).

We define now a Simple Sequential Process with Resources (S^2PR), as an S^2P which needs the use of a unique resource at every state that is not the idle state. Because the interactions with the rest of the processes in the system will be made by means of the sharing of the set of resources, it is natural to assume that in the idle state there is no interaction with the rest of the system, and therefore, no resource is used in this state.

Definition IV.2: A *Simple Sequential Process with Resources* (S^2PR) is a Petri net $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R . T, F \rangle$ so that:

1) The subnet generated by $X = P \cup \{p^0\} \cup T$ is an S^2P
2) $P_R \neq \emptyset$ and $(P \cup \{p^0\}) \cap P_R = \emptyset$
3) $\forall\ p \in P,\ \forall\ t \in {}^\bullet p,\ \forall t' \in p^\bullet ,\ {}^\bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$
4) The two following statements are verified:
   a) $\forall\ r \in P_R .\ {}^{\bullet\bullet} r \cap P = r^{\bullet\bullet} \cap P \neq \emptyset$
   b) $\forall\ r \in P_R .\ {}^\bullet r \cap r^\bullet = \emptyset$
5) ${}^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R = \emptyset$

The marking of places in P_R models either the capacity of a resource to accept new parts or the number of non engaged copies of the considered resource. In the sequel we will call resource places to the elements of P_R (in short, resources). P is the set of *state places*. For a given state place $p \in P$, the place $r_p \in P_R$ given by condition 3 in the definition models the resource used at this state. For a given $r \in P_R$, we will denote as $H(r) = ({}^{\bullet\bullet}r) \cap P$ the set of *holders* of r (states that use r). Condition 4 in the previous definition imposes that two adjacent states of a WP (both of them different from the idle state) cannot use the same resource. This is not a constraint, since from the liveness perspective, two adjacent states using the same resource can be collapsed into a unique state, preserving the behavioral properties of the net (see [15], [12]).

The definition of an S^2PR is a generalization of the concept of "production sequence" in [2] or "production Petri net model" in [9]. This generalization is due to the fact that in the S^2PR models choices are allowed in the state machines modeling the flow of parts. The two special constraints imposed to the state machines in an S^2P and the way the S^2PR uses the set of resources is what gives the name "simple" to these processes.

Now, we are going to introduce a class of initial markings for the S^2PR class.

Definition IV.3: Let $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R, T, F\rangle$ be an S^2PR. An initial marking m_0 is called an *acceptable initial marking* for $\mathcal{N}$ iff: 1) $m_0(p^0) \geq 1$; 2) $m_0(p) = 0, \ \forall p \in P$ and 3) $m_0(r) \geq 1, \ \forall r \in P_R$.

The couple $\langle \mathcal{N}, m_0\rangle$ is called a (acceptably) marked S^2PR. Notice that an acceptable marking assigns at least one token in the idle place (then, we assume that, initially, each copy -token- of each process is idle) and at least one token in every resource, i.e., there is at least a copy of every resource in the system. It is clear that if there exists a resource for which there is no copy, the system is not well defined, because it can have some production sequence that cannot be carried out. Note also that this marking is "greater or equal" than the "minimal resource requirement" as defined in [9].

In the sequel, when we talk about a marked S^2PR, we will refer to an S^2PR with an acceptable initial marking. In Figs. 3(a) and 3(b) two marked S^2PR are shown. For instance, the different elements of the S^2PR in Fig. 3(b) are the following: $P^0 = \{p'\}$, $P_R = \{r2, r3, r4\}$, $P = \{a', b', c'\}$.

Notation: in the sequel, given an S^2PR, $\mathcal{N} = \langle P \cup \{p^0\} \cup P_R, T, F\rangle$, we denote $P^0 = \{p^0\}$.

We introduce now, recursively, the definition of a system of S^2PR, that we call S^3PR.

Definition IV.4: A System of S^2PR, S^3PR, is defined recursively as follows:

1) An S^2PR is an S^3PR
2) Let $\mathcal{N}_i = \langle P_i \cup P_i^0 \cup P_{R_i}, F_i\rangle$, $i \in \{1, 2\}$ be two S^3PR so that $(P_1 \cup P_1^0) \cap (P_2 \cup P_2^0) = \emptyset$, $P_{R_1} \cap P_{R_2} = P_C$ $(\neq \emptyset)$ and $T_1 \cap T_2 = \emptyset$ (in which case we will say that $\mathcal{N}_1$ and $\mathcal{N}_2$ are *two composable* S^3PR); then, the net $\mathcal{N} = \langle P \cup P^0 \cup P_R, T, F\rangle$ resulting of the composition of $\mathcal{N}_1$ and $\mathcal{N}_2$ via P_C (denoted as $\mathcal{N} = \mathcal{N}_1 \circ \mathcal{N}_2$) defined as follows: 1) $P = P_1 \cup P_2$, 2) $P^0 = P_1^0 \cup P_2^0$, 3) $P_R = P_{R_1} \cup P_{R_2}$, 4) $T = T_1 \cup T_2$ and 5) $F = F_1 \cup F_2$ is also an S^3PR.

The meaning of the previous definition is clear: two S^3PR are composable when they share a set of resources, and then, their composition is defined as the composition of the two nets via a set of common places. We assume that shared resources have the same labels in both S^2PR.

We introduce now the definition of an acceptable marking for an S^3PR.

Definition IV.5: Let $\mathcal{N}$ be an S^3PR. $\langle \mathcal{N}, m_0\rangle$ is an *acceptably marked* S^3PR iff one of the two following statements is true:

- $\langle \mathcal{N}, m_0\rangle$ is an acceptably marked S^2PR
- $\mathcal{N} = \mathcal{N}_1 \circ \mathcal{N}_2$, so that $\langle \mathcal{N}_i, m_{0_i}\rangle$ is an acceptably marked S^3PR and

    a) $\forall i \in \{1, 2\}, \ \forall p \in P_i \cup P_i^0, \ m_0(p) = m_{0_i}(p)$
    b) $\forall i \in \{1, 2\}, \ \forall r \in P_{R_i} \setminus P_C, \ m_0(r) = m_{0_i}(r)$
    c) $\forall r \in P_C, \ m_0(r) = max\{m_{0_1}(r), m_{0_2}(r)\}$

The last condition concerns the initial marking of the shared resources in the composed model. This condition is quite natural if we have a set of partial and "correct" models that have to be composed in order to obtain the global model. In effect, the submodel of the global model corresponding to each working process ought to have enough resources ensuring the correct behavior of the isolated process. For instance, if the initial marking in an S^2PR of a resource is k_1, while in other S^2PR is k_2 $(k_2 \geq k_1)$, and both have to be composed, assuming that both models are correct, the composed system will have k_2 copies of the resource. In the sequel, we denote by means of $\mathcal{N} = \bigcirc_{i=1}^{k} \mathcal{N}_i$ the net defined as follows: *if* $k = 1$ *then* $\mathcal{N} = \mathcal{N}_1$; *if* $k > 1$ *then* $\bigcirc_{i=1}^{k} \mathcal{N}_i = (\bigcirc_{i=1}^{k-1} \mathcal{N}_i) \circ \mathcal{N}_k$. Given $\mathcal{N}$ in this way, we denote $I_{\mathcal{N}} = \{1, \ldots, k\}$; on the other hand, $\overline{\mathcal{N}}_i$ represents the S^2P from which we form the S^2PR $\mathcal{N}_i$. Fig. 3(c) shows the S^3PR resulting from the composition of the S^2PR in figures 3-a and b.

In the sequel, when talking about a marked S^3PR we refer to an S^3PR with an acceptable initial marking. We present now some structural features of an S^3PR that will be used later on.

Notation: Given a set $X \subseteq P \cup P^0 \cup P_R$, by e_X we denote the $(P \cup P^0 \cup P_R)$-indexed vector so that $e_X(p) =$ *If* $p \in X$ *then* 1 *else* 0.

Proposition IV.1 [7]: Let $\mathcal{N} = \langle P \cup P^0 \cup P_R, T, F\rangle$ be an S^3PR. The family $\{e_{P_i \cup P_i^0} \mid i \in I_{\mathcal{N}}\} \cup \{e_{H(r) \cup \{r\}} \mid r \in P_R\}$ is the set of minimal p-semiflows of $\mathcal{N}$. Moreover, this family forms one basis of the left anuller space of the flow matrix C.

As an immediate corollary of the previous proposition we have that:

Corollary IV.1: Let $\langle \mathcal{N}, m_0\rangle$ be a marked S^3PR. Then:

1) $\mathcal{N}$ is conservative
2) For all $m \in \mathcal{R}(\mathcal{N}, m_0)$ we have that: **1)** for all $i \in I_{\mathcal{N}}$, $\sum_{p \in P_i \cup P_i^0} m(p) = m_0(p_i^0)$. **2)** for all $r \in P_R$, $\sum_{p \in H(r) \cup \{r\}} m(p) = m_0(r)$

The above corollary says that each place of $\mathcal{N}$ belongs, at least, to the support of a p-semiflow (part 1). Part 2 states that each p-semiflow induces a token conservation law holding for each reachable marking. Moreover, given the special form of the p-semiflows and taking into account that the initial marking is acceptable, we can conclude that all p-semiflows are always marked. The following proposition states that a siphon not containing the support of any p-semiflow has, at least, two places modeling system resources. This property will be used later on.

Proposition IV.2: Let $\mathcal{N} = \langle P \cup P^0 \cup P_R, T, F\rangle$ be an S^3PR, and let $S(\neq \emptyset)$ be a siphon such that it does not contain the support of any p-semiflow. Then, $\mid S \cap P_R \mid > 1$.

Proof: If $S \cap P_R = \emptyset$, since each $\overline{\mathcal{N}}_i$ is a strongly connected state machine and $T_i \cap T_j = \emptyset$, $\forall\ i \neq j$, we can conclude that there exists $i \in I_{\mathcal{N}}$ so that $(P_i \cup P_i^0) \subseteq S$, and then, S contains the support of a p-semiflow, which is not possible. Then, $\mid S \cap P_R \mid > 0$.

Let us consider now $r \in S \cap P_R$; since $H(r) \not\subseteq S$ (because, by Proposition 1, $H(r) \cup \{r\}$ is the support of one p-semiflow), let $p \in H(r) \setminus S$. Let $\{t\} = p^\bullet \cap {}^\bullet r$; since S is a siphon and $p \notin S$, necessarily ${}^\bullet t \cap P_R = \{r'\} \subset S$. Considering now statement 4.b of Definition 2, we can ensure that $r \neq r'$, and then, $\{r, r'\} \subseteq S$.

V. Liveness Analysis of S^3PR Models

In this section we study some behavioral properties of S^3PR's. We will see that the behavior of these systems is related to structural Petri net objects such as siphons. First of all, we will see that the special structure of our systems allows us to prove that an empty siphon, under a marking m, is a necessary and sufficient condition for the net to have a dead transition. This characterization will be used later on to give a method to synthesize live models.

Previously we prove some technical lemmata that allow us to have a better understanding of our systems.

Lemma V.1: Let $\langle\mathcal{N}, m_0\rangle$ be a marked S^3PR. Then, for all $t \in T$ there exists $\sigma_t$such that $m_0[\sigma_t\rangle m_t$ and $m_t[t\rangle$.

Proof: Let us assume that $t \in T_i$. Since $\overline{\mathcal{N}}_i$ is a strongly connected state machine, there exists a path from p_i^0 to t. Let $(p_i^0 = p_0, t_1, p_1, t_2, \ldots t_n, p_n, t_{n+1} = t)$ be such a path. Considering that m_0 is an acceptable initial marking, $m_0[t_1\rangle m_1$. Then, $m_1(p_1) > 0$ and $m_1({}^\bullet t_2 \cap P_R) > 0$ because the resource ${}^\bullet t_2 \cap P_R$ is not used in p_1 and the rest of the processes are in their idle state. Then, $m_1[t_2\rangle$. By iteration of this reasoning, we can reach a marking m_t as $m_0[t_1 \ldots t_n\rangle m_t[t\rangle$ and we are done.

The previous lemma means that a transition $t \in T_j$ can be fired from the initial state if we freeze the S^2PR's not containing t. This is due to the fact that there exist enough resources, in each idle S^2PR, to finish each working process.

Notation: Given a set $X \subseteq T$, by $\sigma_{|X}$ we denote the projection of the firing sequence σ with respect to the set X.

Lemma V.2: Let $\langle\mathcal{N}, m_0\rangle$ be a marked S^3PR, $m \in \mathcal{R}(\mathcal{N}, m_0)$ and $j \in I_{\mathcal{N}}$ the index of an S^2P such that $m(p_j^0) = m_0(p_j^0)$. If there exists a firing sequence σ such that $m[\sigma\rangle m'[t\rangle$ where $t \notin T_j$, then $\sigma_{|T \setminus T_j}$ verifies $m[\sigma_{|T\setminus T_j}\rangle m''[t\rangle$.

Proof: In the case in which $\sigma_{|T\setminus T_j} = \sigma$, the lemma follows trivially. Let us assume now that $\sigma_{|T\setminus T_j} \neq \sigma$. If we remove all transitions belonging to T_j from σ, we are removing constraints to the firing of the rest of the transitions. In effect, at marking m, $\overline{\mathcal{N}}_j$ is in its idle state, and then, it uses no resource. If we "freeze" $\overline{\mathcal{N}}_j$, we have more free resources to fire the transitions of σ not belonging to T_j. Therefore, $\sigma_{|T\setminus T_j}$ is also firable at marking m.

The previous lemma states that if we apply a firing sequence from the initial marking and the net reaches a marking that enables a transition t, then there exists a firing sequence where no transition of a process in an idle state arises and which leads to a marking that also enables t.

The next lemma states that, if a transition of an S^3PR is dead for a reachable marking, then m_0 is not reachable any more.

Lemma V.3: Let $\langle\mathcal{N}, m_0\rangle$ be a marked S^3PR, and let $m \in \mathcal{R}(\mathcal{N}, m_0)$ be a reachable marking so that $t \in T$ is a dead transition for m. Then, $m_0 \notin \mathcal{R}(\mathcal{N}, m)$

Proof: Straight forward from Lemma 1.

The following corollary proves that if there is a reachable marking such that a transition is dead, then there exists a set of processes where some tokens cannot evolve any more.

Corollary V.1: Let $\langle\mathcal{N}, m_0\rangle$ be a marked S^3PR; let $m \in \mathcal{R}(\mathcal{N}, m_0)$. If $t \in T$ is a dead transition for m, there exist $m' \in \mathcal{R}(\mathcal{N}, m)$ and two subsets $I \subseteq I_{\mathcal{N}}$ and $H \subset I_{\mathcal{N}}$ such that $I_{\mathcal{N}} = I \cup H$, $I \cap H = \emptyset$, $I \neq \emptyset$ and: 1) $\forall\ h \in H, m'(p_h^0) = m_0(p_h^0)$; 2) $\forall\ i \in I, m'(p_i^0) < m_0(p_i^0)$ and $\{p^\bullet \mid p \in P$ and $m'(p) > 0\}$ is a set of dead transitions.

Proof: Let us consider $I \subseteq I_{\mathcal{N}}$, the set of all indexes so that $\forall\ m'' \in \mathcal{R}(\mathcal{N}, m)$, $m''(p_i^0) < m_0(p_i^0)$ (the case $m''(p_i^0) > m_0(p_i^0)$ is not possible because of Corollary 1). Note that $I \neq \emptyset$, since, on the contrary, $m_0 \in \mathcal{R}(\mathcal{N}, m)$, which is not possible because t is a dead transition for m (Lemma 3). We denote $H = I_{\mathcal{N}} \setminus I$.

Then, we can ensure that there exists $m_H \in \mathcal{R}(\mathcal{N}, m)$ so that $\forall\ h \in H$, $m_H(p_h^0) = m_0(p_h^0)$. From this marking, and moving only processes corresponding to indexes in I, and considering condition 3 of Definition 1, the net reaches a marking $m' \in \mathcal{R}(\mathcal{N}, m)$ verifying also condition 2 in the hypothesis, because on the contrary $m_0 \in \mathcal{R}(\mathcal{N}, m)$, which is in contradiction with Lemma 3.

Now, using the special structure of an S^3PR, we prove one of its main behavioral properties: if a transition is dead for a reachable marking, then a marking is reachable from it so that a siphon is empty. The basic idea behind the proof is the building of an empty siphon. This siphon is composed of two different sets of places the marking of which is 0: the unmarked resources and the unmarked holders of these resources.

Theorem V.1: Let $\langle\mathcal{N}, m_0\rangle$ be a marked S^3PR, let $m \in \mathcal{R}(\mathcal{N}, m_0)$ and let $t \in T$ be a dead transition for m. Then, $\exists\ m' \in \mathcal{R}(\mathcal{N}, m), \exists\ S$ a siphon so that $m'(S) = 0$.

Proof: Let us consider the marking m' given in Corollary 2, and let $S_R = \{r \in P_R \mid m'(r) = 0\}$ and $S_P = \{p \in H(r) \mid r \in S_R, m'(p) = 0\}$. Then, defining $S = S_R \cup S_P$, we are going to prove that S is a nonempty unmarked siphon.

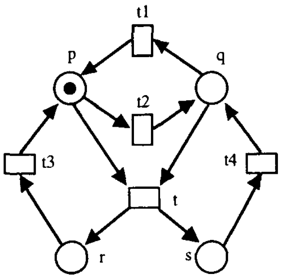


Fig. 4. Transition t is a dead transition for the shown marking, but the only siphon $\{p, q, r, s\}$ is always marked.

- $S \neq \emptyset$: By contradiction. If $S = \emptyset$, then $S_R = \emptyset$. This means that for all $p \in P$ such that $m'(p) > 0$ we have that for all $t' \in p^\bullet$, $m'[t'\rangle$. Therefore, the set of indexes I given in Corollary 2 is empty, which is not possible.
- $m'(S) = 0$: This is immediate considering the way in which the places of S_R and S_P have been taken.
- S is a siphon: Let $t' \in {}^\bullet S$. Then, we have to consider two different cases.

  a) $t' \in {}^\bullet r$ for $r \in S_R$: Let $\{q\} = {}^\bullet t' \cap P$. If $m'(q) = 0$, then $q \in S_P$, which implies that $t' \in q^\bullet \subseteq S_P{}^\bullet \subseteq S^\bullet$. If $m'(q) \neq 0$, then ${}^\bullet t' \cap P_R \neq \emptyset$, since, on the contrary, $m'[t'\rangle$, which is in contradiction with Corollary 2. Let $\{r'\} = {}^\bullet t' \cap P_R$. If $m'(r') > 0$ then $m'[t'\rangle$, which is not possible. Therefore, $r' \in S_R$, and then, $t' \in r'^\bullet \subseteq S_R{}^\bullet \subseteq S^\bullet$

  b) $t' \in {}^\bullet p$ for $p \in S_P$: Since $p \in S_P$, then $r_p \in S_R$, and considering statement 3 in Definition 2, $t' \in r_p{}^\bullet \subseteq S_R{}^\bullet \subseteq S^\bullet$

The last result is not true in general Petri nets, as shown in Fig. 4. Transition t is dead for the shown marking, but the only siphon in the net, $\{p, q, r, s\}$, is always marked.

Now, we can characterize the liveness in S^3PR models:

Corollary V.2: Let $\langle \mathcal{N}, m_0 \rangle$ be a marked S^3PR. Then, $\langle \mathcal{N}, m_0 \rangle$ is live if, and only if, $\forall$ $m \in \mathcal{R}(\mathcal{N}, m_0)$, $\forall$ (minimal) siphon S, $m(S) \neq 0$

Proof: $\Longrightarrow$) if there exists $m \in \mathcal{R}(\mathcal{N}, m_0)$ and a siphon S so that $m(S) = 0$, then $\forall m' \in \mathcal{R}(\mathcal{N}, m)$, $m'(S) = 0$ (this a basic property of siphons) and then, no output transition of S can be enabled any more. Therefore, the net is not live.

$\Longleftarrow$) if no reachable marking leads to an empty siphon, by theorem 1 no transition can be dead.

VI. A Control Policy for Deadlock Prevention in S^3PR Models

Let us suppose now that we have an S^3PR model where some deadlock can arise. Our aim is to introduce into the system a control policy assuring that in the system evolution no deadlock situation is reachable. We define a control policy as the addition of new constraints to the system so that its initial behavior is restricted to a set of states that we consider as "good states", considering a good state as the one allowing the system to evolve without reaching a deadlock state.

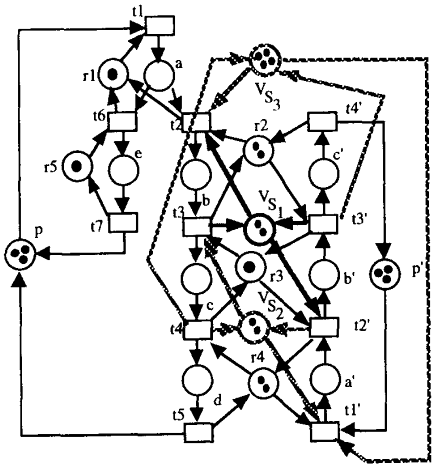


Fig. 5. The S^3PR in Fig. 3 with a wrong control policy.

Let us consider the S^3PR shown in Fig. 3(c). The marking $m_1 = 2b + b' + r1 + 2r4 + r5 + p + 2p'$ is reachable from m_0 after the firing sequence $t1t2t1t2t1't2'$ and it is a deadlock. Analogously, the reachable marking $m_2 = c + 2a' + r1 + 2r2 + r5 + 2p + p'$ is also a deadlock. In this case, we can see that for m_1 the siphon $S_1 = \{r2, r3, c, c'\}$ is empty, and that for m_2 the siphon $S_2 = \{r3, r4, d, b'\}$ is also empty. Taking into account Corollary 3 we can ensure that if, by the addition of some control policy, no siphon can be emptied, then the controlled system is live. But this control has to be carried out carefully because, even if we guarantee that no siphon can become empty, we must also ensure that no new problem is introduced.

A "naive" control policy will try to evaluate the total number of tokens flowing for every transition and every minimal siphon that is the support of no p-semiflow. For instance, in the previous example, let us consider the transition $t2$. Its firing withdraws a token from siphon S_1 and puts no token into it. Considering the system carefully, we can see that, for siphon S_1, every withdrawn token is put into places b or b' and for all reachable marking m, it is verified that $m(b) + m(b') + m(S_1) = m_0(S_1)$. So, if for every reachable marking $m(b) + m(b') < m_0(S_1)$, we have that $m(S_1) \geq 1$, and then, S_1 will never be empty. We can implement this policy by means of the addition of a new place V_{S_1} so that $m_0(V_{S_1}) = m_0(S_1) - 1$ and the following new arcs: $(V_{S_1}, t2), (V_{S_1}, t2'), (t3, V_{S_1}), (t3', V_{S_1})$ (transitions $t3$ and $t3'$ withdraw one token each from siphon S_1 and put two tokens into it). We make the same for siphons $S_2 = \{r3, r4, d, b'\}$ and $S_3 = \{d, c', r2, r3, r4\}$, obtaining the places V_{S_2} and V_{S_3}, respectively.

Figure 5 shows the resulting controlled model. However, even if no siphon in the initial net can be emptied, we have a deadlock in the new model because the marking $m_3 = 2b + 2a' + r1 + r3 + r5 + p + p'$ is a deadlock. This is due to the fact that, in the new model, added places generate a new siphon, $\{d, c', r2, r4, V_{S_1}, V_{S_2}\}$, that is empty for marking m_3.

As a conclusion, we can say that we need to avoid not only markings with some empty siphon, but also the markings leading ``inevitably" to other markings with empty siphons.

In the following, we present a control policy that added to an S^3PR will allow us to guarantee that the final system is live. First of all, we introduce some notations.

Notation: Let $\mathcal{N} = \langle P, T, F\rangle$ be an S^2P.

- Let $\mathcal{C}$ be a circuit of $\mathcal{N}$ and let x, y be two nodes of $\mathcal{C}$. We will say that x is *previous* to y in $\mathcal{C}$ if, and only if, there exists a path in $\mathcal{C}$ from x to y the length (given a circuit $\mathcal{C}$, we denote ||{\cal C}|| the set of nodes in it, and *length*($\mathcal{C}$) =| $||\mathcal{C}||$ |) of which is greater than 1 and does not pass over p^0. This fact will be denoted as $x <_{\mathcal{C}} y$. For instance, in the net in Fig. 3(c), $t2 <_{\mathcal{C}} d$, being $\mathcal{C}$ the cycle the support of which is $\{p, t1, a, t2, b, t3, c, t4, d, t5\}$.
- Let x, y be two nodes of $\mathcal{N}$. We will say that x is *previous* to y in $\mathcal{N}$ if, and only if, there exists a circuit $\mathcal{C}$ so that $x <_{\mathcal{C}} y$. This fact will be denoted as $x <_{\mathcal{N}} y$
- Let x and $A \subseteq (P \cup T)$ be a node and a set of nodes of $\mathcal{N}$, respectively. Then $x <_{\mathcal{N}} A$ if, and only if, there exists a node $y \in A$ so that $x <_{\mathcal{N}} y$.
- Let x and $A \subseteq (P \cup T)$ be a node and a set of nodes of $\mathcal{N}$, respectively. Then $A <_{\mathcal{N}} x$ if, and only if, there exists a node $y \in A$ so that $y <_{\mathcal{N}} x$.

The following notation will be used in the establishment of the control policy.

Notation: Let $\mathcal{N} = \bigcirc_{i=1}^{k}\mathcal{N}_i = \langle P \cup P_R \cup P^0, T, F\rangle$ be an S^3PR.

- $\mathcal{S}$ denotes the set of minimal siphons which does not contain the support of any p-semiflow (i.e., siphons that can be emptied). For the S^3PR in Fig. 3(c), we have $\mathcal{S} = \{S_1, S_2, S_3\}$, where $S_1 = \{c, c', r2, r3\}$, $S_2 = \{d, b', r3. r4\}$ and $S_3 = \{d, c', r2, r3, r4\}$.
- Given $S \in \mathcal{S}$ ($S = S_P \cup S_R$, $S_R = S \cap P_R, S_P = S \setminus S_R$) C_S denotes the following set of state places: $C_S = (\cup_{r \in S_R} H(r)) \setminus S$, while $C_S^i = C_S \cap P_i$. For a given siphon S, C_S is the set of holders, corresponding to resources in S, which do not belong to S. For the net in Fig. 3(c) we have $C_{S_1} = \{b, b'\}$, $C_{S_2} = \{c, a'\}$, $C_{S_3} = \{b, c, a', b'\}$.

  Notice that for a given siphon $S \in \mathcal{S}$, a token in a place belonging to C_S models a WP state that uses one of the resources involved in the siphon. Then, each time a token reaches C_S, the marking of siphon S has decreased in one token. Informally speaking, we will say that the token modeling the part has ``stolen" some token from the siphon. It is easy to see that $\forall\ S \in \mathcal{S}$, $C_S \cup S = S_R \cup (\cup_{r \in S_R} H(r))$, and then, $C_S \cup S$ is the support of a p-semiflow. C_S will be called the *complementary set of S*
- $\mathcal{S}^+ : T \longrightarrow \mathcal{P}(\mathcal{S})$ (Given a set A, $\mathcal{P}(A)$ denotes the powerset of A) is the mapping defined as follows: let us assume that $t \in T_i$; then $\mathcal{S}^+(t) = \{S \in \mathcal{S} \mid t <_{\overline{\mathcal{N}}_i} C_S^i\}$. If $S \in \mathcal{S}^+(t)$ then the set C_S is ``reachable" from t; i.e., there exists in $\overline{\mathcal{N}}_i$ a path leading to an state place which is not in S and which uses some resource of S. For the net in Fig. 3(c), we have, for instance, that $\mathcal{S}^+(t2) = \{S_1, S_2, S_3\}$, while $\mathcal{S}^+(t4) = \emptyset$.
- $\mathcal{S}^- : T \longrightarrow \mathcal{P}(\mathcal{S})$ is the mapping defined as follows: let us assume that $t \in T_i$; then $\mathcal{S}^-(t) = \{S \in \mathcal{S} \mid C_S^i <_{\overline{\mathcal{N}}_i} t\}$. For instance, $\mathcal{S}^-(t2) = \emptyset$, while $\mathcal{S}^-(t4) = \{S_1, S_2, S_3\}$.
- $\forall\ i \in I_{\mathcal{N}}$, $\forall\ S \in \mathcal{S}$, $P_S^i = C_S^i \cup \{p \in P_i \mid p <_{\overline{\mathcal{N}}_i} C_S^i\}$, and $P_S = \cup_{i \in I_{\mathcal{N}}} P_S^i$. (Notice that $p_i^0 \notin P_S^i$). For instance, $P_{S_1} = \{a, b, a', b'\}$.

Definition VI.1: Let $\langle \mathcal{N}, m_0\rangle$ be a marked S^3PR ($\mathcal{N} = \bigcirc_{i=1}^{k}\mathcal{N}_i = \langle P \cup P_R \cup P^0. T, F\rangle$). The net

$$\langle \mathcal{N}_A, m_{0_A}\rangle = \langle P \cup P_R \cup P^0 \cup P_A, T, F \cup F_A, m_{0_A}\rangle$$

is *the controlled system* of $\langle \mathcal{N}, m_0\rangle$ if, and only if,

- $P_A = \{V_S \mid S \in \mathcal{S}\}$ is a set of places so that there exists a bijective mapping from $\mathcal{S}$ into it.
- $F_A = F_A^1 \cup F_A^2 \cup F_A^3$ so that:

$$F_A^1 = \{(V_S, t) \mid t \in {P^0}^{\bullet}, S \in \mathcal{S}^+(t)\}$$
$$F_A^2 = \{(t, V_S) \mid t \in {C_S}^{\bullet}, S \notin \mathcal{S}^+(t)\}$$
$$F_A^3 = \bigcup_{i \in I_{\mathcal{N}}} \{(t, V_S) \mid t \in T_i \setminus {P^0}^{\bullet}, S \notin \mathcal{S}^-(t),$$
$${}^{\bullet}t \cap P_i \in P_S^i, t \not<_{\overline{\mathcal{N}}_i} C_S^i\}$$

- m_{0_A} is defined as follows:

  a) $\forall p \in P \cup P_R \cup P^0, m_{0_A}(p) = m_0(p)$
  b) $\forall V_S \in P_A, m_{0_A}(V_S) = m_0(S) - 1$

In the following, we denote by $\mathcal{M} = \mathcal{R}(\mathcal{N}, m_0)$ and $\mathcal{M}_A = \mathcal{R}(\mathcal{N}_A, m_{0_A})$ the set of reachable markings of the net and its controlled one, respectively. The intuitive meaning of the control policy is the following. Each token leaving the idle state (i.e., each process whose execution starts) takes a token from the added places related to ``dangerous" siphons from which this process can ``steal" some token during its execution. This is implemented by arcs in F_A^1. Tokens leaving the complementary set of a siphon S (i.e., tokens that can reach the idle state place without ``visiting" places in $C_S$) have to release the token corresponding to the added place related to $S$ (this token was taken when the process started). These cases are considered by the set of arcs $F_A^2$. Arcs in $F_A^3$ represent the situations of processes that when started could ``steal" some tokens of a siphon but that finally, because of the routing followed, cannot.

Then, the goal of an added place V_S is to avoid that the number of processes that can stay in its C_S places be greater or equal than the initial marking of S, without generating new deadlock situations. The resulting model of the S^3PR in Fig. 3(c) is shown in Fig. 6.

In this case, the considered elements are the following:

$$P_A = \{V_{S_1}, V_{S_2}, V_{S_3}\},$$
$$F_A^1 = \{(V_{S_1}, t1), (V_{S_1}, t1'), (V_{S_2}, t1'),$$
$$(V_{S_2}, t1), (V_{S_3}, t1), (V_{S_3}, t1')\},$$
$$F_A^2 = \{(t3, V_{S_1}), (t3', V_{S_1}), (t4. V_{S_2}).$$
$$(t2', V_{S_2}). (t4. V_{S_3}). (t3'. V_{S_3})\},$$
$$F_A^3 = \{(t6, V_{S_1}). (t6. V_{S_2}). (t6. V_{S_3})\}.$$
$$m_{0_A}(V_{S_1}) = m_{0_A}(V_{S_2}) = 2,$$
$$m_{0_A}(V_{S_3}) = 4$$

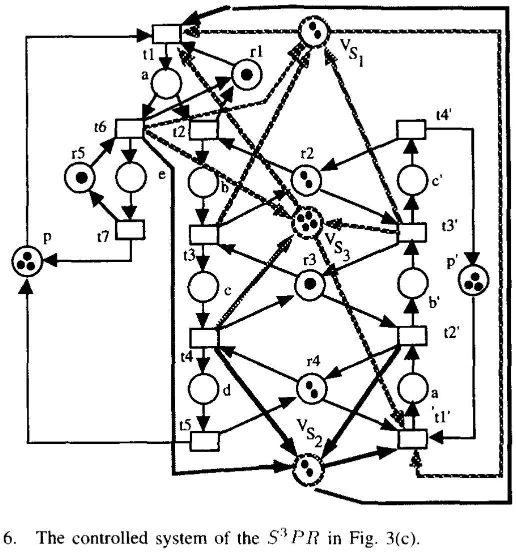


Fig. 6. The controlled system of the S^3PR in Fig. 3(c).

Now, we have to prove that the established control policy leads to live models.

Lemma VI.1: Let $\langle \mathcal{N}_A, m_{0_A} \rangle$ be the controlled system of the marked S^3PR, $\langle \mathcal{N}, m_0 \rangle$, and let $m_A \in \mathcal{M}_A$. Then, $\forall\ S \in \mathcal{S}$ the following invariant relation is verified

$$m_A(V_S) + \sum_{i \in I_{\mathcal{N}}} m_A(P_S^i) = m_{0_A}(V_S) \qquad (1)$$

Proof: Let us assume that $m_{0_A}[\sigma\rangle m_A$. The proof will be made by induction over the length of σ. If $\sigma = \epsilon$ (the empty firing sequence) then $\forall i \in I_{\mathcal{N}}$, $\forall\ S \in \mathcal{S}$, $m_A(P_S^i) = 0$, and therefore, (1) is equivalent to $m_{0_A}(V_S) = m_{0_A}(V_S)$, that is obviously true. Let us assume now that $m_{0_A}[t_1\rangle m_A^1 ... [t_{n-1}\rangle m_A^{n-1}[t_n\rangle m_A^n = m_A$, and that (1) is verified by m_A^{n-1}. Let us also assume that $t_n \in T_j$ and $p = {}^\bullet t_n \cap P_j$, $q = t_n{}^\bullet \cap P_j$. Let $S \in \mathcal{S}$. We have to consider three different cases:

1) Let $S \in \mathcal{S}\ \ni\ t_n \in V_S{}^\bullet$: in this case, $p = p_0^j$. Then

   a) $m_A^n(V_S) = m_A^{n-1}(V_S) - 1$
   b) $\forall\ i \neq j,\ m_A^n(P_S^i) = m_A^{n-1}(P_S^i)$
   c) $t_n \in V_S{}^\bullet \Longrightarrow p \notin P_S^j, q \in P_S^i \Longrightarrow m_A^n(P_S^j) = m_A^{n-1}(P_S^j) + 1$

2) Let $S \in \mathcal{S}\ \ni\ t_n \in {}^\bullet V_S$:

   a) $m_A^n(V_S) = m_A^{n-1}(V_S) + 1$
   b) $\forall\ i \neq j,\ m_A^n(P_S^i) = m_A^{n-1}(P_S^i)$
   c) If $(t_n, V_S) \in F_A^2 \Longrightarrow (t_n \in C_S{}^\bullet \wedge S \notin \mathcal{S}^+(t_n)) \Longrightarrow p \in C_S^j$; since $t_n \not<_{\overline{\mathcal{N}}_j} C_S^j$, then $q \not<_{\overline{\mathcal{N}}_j} C_S^j$ ($\Longrightarrow q \notin C_S^j$); therefore $m_A^n(P_S^j) = m_A^{n-1}(P_S^j) - 1$. If $(t_n, V_S) \in F_A^3 \Longrightarrow p <_{\overline{\mathcal{N}}_j} P_S^j$ and $t \not<_{\overline{\mathcal{N}}_j} C_S^j$, then $q \notin P_S^j$, and therefore, $m_A^n(P_S^j) = m_A^{n-1}(P_S^j) - 1$.

3) $t_n \notin \cup_{S \in \mathcal{S}}({}^\bullet V_S \cup V_S{}^\bullet)$:

   a) $m_A^n(V_S) = m_A^{n-1}(V_S)$
   b) $\forall\ i \neq j,\ m_A^n(P_S^i) = m_A^{n-1}(P_S^i)$
   c) If $p \in P_S^j$, then $q \in P_S^j$ since, on the contrary, $t_n \in {}^\bullet V_S$. If $p \notin P_S^j$, then $q \notin P_S^j$ since, on the contrary, $t_n \in V_S{}^\bullet$. In both cases, $m_A^n(P_S^j) = m_A^{n-1}(P_S^j)$.

We can see that in all cases (1) is true.

The following obvious lemma states that every firing sequence of the controlled net is also a firing sequence of the initial one.

Lemma VI.2: Let $\langle \mathcal{N}_A, m_{0_A} \rangle$ be the controlled system of the marked S^3PR $\langle \mathcal{N}, m_0 \rangle$, and let σ be a firing sequence of $\langle \mathcal{N}_A, m_{0_A} \rangle$. Then, σ is also a firing sequence of $\langle \mathcal{N}, m_0 \rangle$.

The following lemma shows that no siphon of the initial net can become empty:

Lemma VI.3: Let $\langle \mathcal{N}_A, m_{0_A} \rangle$ be the controlled system of the marked S^3PR $\langle \mathcal{N}, m_0 \rangle$, and let $m_A \in \mathcal{M}_A$ be a reachable marking. Then, $\forall\ S \in \mathcal{S},\ m_A(S) \neq 0$

Proof: Let $S \in \mathcal{S}$; since $S \cup C_S = S_R \cup \bigcup_{r \in S_R} H(r)$ is the support of a p-semiflow (weighted by 1) and $S \cap C_S = \emptyset$, we can conclude that $m_A(S) + m_A(C_S) = m_{0_A}(S) = m_0(S)$.

Considering (1) and this last result, if $m_A(S) = 0$, we have that $m_{0_A}(S) = m_A(C_S) \leq \sum_{i \in I_{\mathcal{N}}} m_A(P_S^i) = m_{0_A}(V_S) - m_A(V_S)$ and then, $m_{0_A}(S) \leq m_{0_A}(S) - 1 - m_A(V_S)$. Therefore, $1 \leq -m_A(V_S)$, which is not possible.

We use now these results in order to prove the liveness of the controlled system.

Lemma VI.4: Let $\langle \mathcal{N}_A, m_{0_A} \rangle$ be the controlled system of the marked S^3PR $\langle \mathcal{N}, m_0 \rangle$, let $m_A \in \mathcal{M}_A$ and let $t \in T$. Then, t is not dead for m_A in $\langle \mathcal{N}_A, m_{0_A} \rangle$.

Proof: Let us assume that $t \in T_i$. The proof is made by induction over the number of tokens in the system that are not in their idle state. Let K^{m_A} be such a number. If $K^{m_A} = 0$, then $m_A = m_{0_A}$, and taking into account that $m_{0_A}(V_S) \geq 1$, $\forall\ S \in \mathcal{S}$ and lemma 2, the thesis is proved (the proof of this lemma for a controlled system is the same than for the initial one). Let us assume now that $K^{m_A} > 0$; since no siphon in the initial net is empty for m_A, there exists $t' \in T \setminus P^{0\bullet}$ so that $m_A/_{\mathcal{N}}[t'\rangle$, and since ${}^\bullet t' \cap (\cup_{S \in \mathcal{S}} V_S) = \emptyset$, we have that $m_A[t'\rangle$. Iterating this reasoning, and taking into account once again that the idle state places are "inevitable" and lemma 2, we can conclude that there exists a firing sequence σ' so that $m_A[\sigma'\rangle m'_A$ with $K^{m'_A} = K^{m_A} - 1$, and we are done.

And finally, we prove the liveness of the controlled system:

Theorem VI.1: Let $\langle \mathcal{N}_A, m_{0_A} \rangle$ be the controlled system of the marked S^3PR $\langle \mathcal{N}, m_0 \rangle$. Therefore, $\langle \mathcal{N}_A, m_{0_A} \rangle$ is live.

Proof: Considering lemma 7, for every reachable marking no transition is dead, and, therefore, we can conclude that the net is live.

VII. An Illustrative Example

As an example of the generation of a live model from an initial one in which we have a non live S^3PR, let us consider the production cell shown in Fig. 7.

TABLE I
SET OF SIPHONS OF THE S^3PR IN FIG. 8 THAT ARE THE SUPPORT OF NO P-SEMIFLOW

i	S_i	m_0
1	{P3M4,P1R3,R3,M4}	3
2	{P2R2,P2R2',P1R2,P1R2',P3M3,R2,M3}	3
3	{P2R2,P2R2',P1R2,P1R2',P3R1,M1,M3,R1,R2}	6
4	{P2R2',P1M2,P1R2',P3R2,M2,R2}	3
5	{P2R2',P1M2,P1R2',P3M3,M2,R2,M3}	5
6	{P2R2',P1M2,P1R2',P3R1,M1,M2,M3,R1,R2}	8
7	{P2R2,P2R2',P1R2,P1M4,P3R2,R2,M4}	3
8	{P2R2,P2R2',P1R2,P1M4,P3M3,M3,M4,R2}	5
9	{P2R2,P2R2',P1R2,P1M4,P3R1,M1,M3,M4, R1,R2}	8
10	{P2R2',P1M2,P1M4,P3R2,M2,M4,R2}	5
11	{P2R2',P1M2,P1M4,P3M3,M2,M3,M4,R2}	7
12	{P2R2',P1M2,P1M4,P3R1,M1,M2,M3,M4, R1,R2}	10
13	{P2R2,P2R2',P1R2,P1R3,P3R2,M4,R2,R3}	4
14	{P2R2,P2R2',P1R2,P1R3,P3M3,R2,R3,M3,M4}	6
15	{P2R2,P2R2',P1R2,P1R3,P3R1,M1,M3, M4,R1,R2,R3}	9
16	{P2R2',P1R3,P3R2,M2,M4,R2,R3}	6
17	{P2R2',P1R3,P3M3,M2,M3,M4,R2,R3}	8
18	{P2R2',P1R3,P3R1,M1,M2,M3,M4,R1,R2,R3}	11

This cell is composed of three robots ($R1, R2$ and $R3$; each one can hold a product at a time) and four machines ($M1, M2, M3$ and $M4$; each one can process two products at a time). There are three loading buffers (named I1,I2,I3) and three unloading buffers (named O1,O2,O3) for loading and unloading the cell. The action area for robot R1 is I1,O3,M1,M3; for robot R2 is I2,O2,M1,M2,M3,M4; and for robot R3 is M2,M4,I3,O1.

Every raw product arriving to the cell belongs to one of the three following types: P1,P2,P3. The product type characterizes the process to be made in the cell as follows:

- A raw product of type P1 is taken from I1 and, once it has been manufactured, is moved to O1. The sequence of operations for this type is $M1; M2$ (i.e., treatment in M1 and then in M2) or $M3; M4$ (treatment in M3 followed by the treatment in M4).
- A raw product of type P2 is taken from I2, manufactured in M2 and routed to O2.

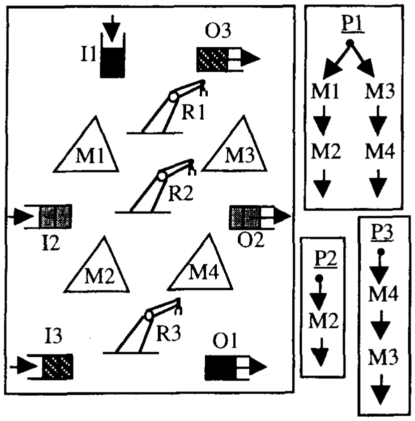


Fig. 7. An example of production cell.

- A raw product of type P3 is taken from I3, manufactured in M4 followed by its treatment in M3, and, finally, placed in O3.

In Fig. 8 the S^3PR resulting from the three working processes obtained for the products in Fig. 7 is shown.

The meaning of the states in Fig. 8 is the following:

- A token in place PiMj, where $i \in \{1,2,3\}, j \in \{1,2,3,4\}$, represents a product of type Pi that is being operated in machine Mj.
- A token in place PiRj, where $i, j \in \{1,2,3\}$, represents a product of type Pi that is now held by the robot Rj.
- Places Rj, where $j \in \{1,2,3\}$, model the available state of the robot Rj
- Places Mj, where $j \in \{1,2,3,4\}$, model the available state of the machine Mj
- Places PiRj' model that the robot Rj is holding a product of type Pi that has already been held by this robot; for instance, a part of type P2 is held by robot R2 twice: once to be moved from I2 to M2, and once again to be moved from M2 to O2; since this states are different, we name the second one with a "prime".

What about the net initial marking? Let us assume that, in the initial state, no product is in the cell and, therefore, the initial marking of every resource is equal to its capacity (one for each robot and two for each machine). And the idle state places (P10,P20,P30)? In order to have the maximum number of products being concurrently manufactured in the cell, the initial marking of each idle place is the total capacity of the net modeling each working process: $m_0(P10) = 11, m_0(P20) = 3, m_0(P30) = 7$ (the maximal capacity of every state place is given by the initial marking of the resource used in this state).

The computation of siphons that are the support of no p-semiflow in the S^3PR in Fig. 8 give us the results shown in Table I.

The new places and arcs added by the control policy are shown in Table II. We can conclude that the final model, once the control policy has been added, is live.

VIII. CONCLUSIONS

This work focused on the research of the liveness property of systems for the control of working processes concur-

TABLE II
THE SETS C_{S_i} AND THE CONTROL PETRI NET ELEMENTS ADDED BY THE CONTROL POLICY

i	C_{S_i}	$V_{S_i}{}^\bullet$	${}^\bullet V_{S_i}$	$m_{0_A}(V_{S_i})$
1	{P1M4,P3R3}	$\{t_1, t_{18}\}$	$\{t_{19}, t_{10}, t_2\}$	2
2	{P1M3,P3R2}	$\{t_1, t_{18}\}$	$\{t_2, t_8, t_{21}\}$	2
3	{P1R1,P1M1,P1M3,P3R2,P3M3}	$\{t_1, t_{18}\}$	$\{t_3, t_8, t_{22}\}$	5
4	{P2R2,P2M2,P1R2}	$\{t_1, t_{12}\}$	$\{t_7, t_4, t_{14}\}$	2
5	{P2R2,P2M2,P1R2,P1M3,P3R2}	$\{t_1, t_{12}, t_{18}\}$	$\{t_4, t_8, t_{14}, t_{21}\}$	4
6	{P2R2,P2M2,P1M2,P1R2,P1M3,P3R2,P3M3}	$\{t_1, t_{12}, t_{18}\}$	$\{t_4, t_8, t_{14}, t_{22}\}$	7
7	{P1R2',P3M4}	$\{t_1, t_{18}\}$	$\{t_2, t_9, t_{20}\}$	2
8	{P1M3,P1R2',P3M4,P3R2}	$\{t_1, t_{18}\}$	$\{t_2, t_9, t_{21}\}$	4
9	{P1R1,P1M3,P1R2',P3M4,P3R2}	$\{t_1, t_{18}\}$	$\{t_2, t_9, t_{22}\}$	7
10	{P2R2,P2M2,P1R2,P1R2',P3M4}	$\{t_1, t_{12}, t_{18}\}$	$\{t_4, t_9, t_{14}, t_{20}\}$	4
11	{P2R2,P2M2,P1R2,P1M3,P1R2',P3M4,P3R2}	$\{t_1, t_{12}, t_{18}\}$	$\{t_4, t_9, t_{14}, t_{21}\}$	6
12	{P2R2,P2M2,P1M1,P1R2,P1M3,P1R2',P3M4,P3R2,P3M3}	$\{t_1, t_{12}, t_{18}\}$	$\{t_4, t_9, t_{14}, t_{22}\}$	9
13	{P1R2',P1M4,P3R3,P3M4}	$\{t_1, t_{18}\}$	$\{t_2, t_{10}, t_{20}\}$	3
14	{P1M3,P1R2',P1M4,P3R3,P3M4,P3R2}	$\{t_1, t_{18}\}$	$\{t_2, t_{10}, t_{21}\}$	5
15	{P1R1,P1M1,P1M3,P1R2',P1M4,P3R3,P3M4, P3R2,P3M3}	$\{t_1, t_{18}\}$	$\{t_2, t_{10}, t_{22}\}$	8
16	{P2R2,P2M2,P1R2,P1M2,P1R2',P1M4,P3M4,P3R3}	$\{t_1, t_{12}, t_{18}\}$	$\{t_5, t_{10}, t_{14}, t_{20}\}$	5
17	{P2R2,P2M2,P1R2,P1M2,P1M3,P1R2',P1M4,P3R3,P3M4,P3R2}	$\{t_1, t_{12}, t_{18}\}$	$\{t_5, t_{10}, t_{14}, t_{21}\}$	7
18	{P2R2,P2M2,P1R1,P1M1,P1R2,P1M2,P1M3,P1R2',P1M4.P3R2 P3M4,P3R3,P3M3}	$\{t_1, t_{12}, t_{18}\}$	$\{t_5, t_{10}, t_{14}, t_{22}\}$	10

rently executed in FMS environments. We have identified a class of systems whose dynamic behavior can be described by means of an S^3PR model, which is a subclass of Petri nets. Constraints imposed on these systems are not too restrictive and give a significant practical interest to our approach.

We have proved a necessary and sufficient condition for liveness of S^3PR models. This characterization has been made in terms of siphons, which are structural Petri net elements. The computation of siphons is a solved problem in Petri net theory [1], [11], [4] and it is implemented in the main tools for analysis of Petri nets [8]. However, an ''ad hoc" implementation for the computation of siphons in S^3PR nets computed the siphons of the net in Section VII in less than 20 seconds.

In the approach followed for the study of deadlock problems, we try to make an extensive use of the structural elements with which the formal model provides us. The control policy establishment complexity is strongly conditioned by the complexity of computing the set of minimal siphons. Efficient algorithms to compute these elements can be found in [11], [4]. These algorithms have an exponential worst-case complexity because the number of minimal siphons can be exponential. However, in the practical cases on which we have worked, the number of elements is not exponential (w.r.t. the number of places in the net). In any case, the computational effort in order to obtain the set of minimal siphons is not critical because this computation is carried out once and off-line. Once the control policy has been established, the response time of the controlled system is shorter than the response time of deadlock avoidance algorithms since these have to make some computations on-line (in real time) each time the system ought to change its state. A second advantage of the proposed method with respect to other known avoidance algorithms is that the liveness characterization allows us to distinguish those systems which, because of their structure, have a live behavior (there is no siphon that can be emptied). We do not add any control policy to these systems and so, the system response time is not increased. As a conclusion, we can say that the proposed deadlock prevention control policy is very interesting when we are dealing with systems for which a deadlock situation is not acceptable at all and for which the on-line system response time is critical.

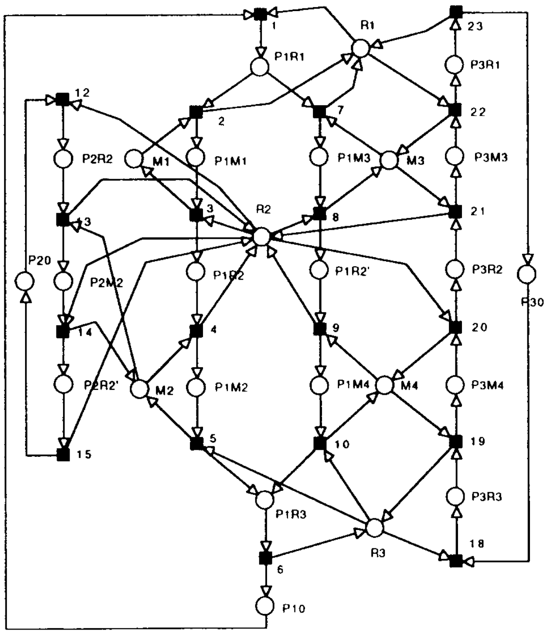


Fig. 8. The S^3PR modeling the system in Fig. 7.

Acknowledgment

The authors would like to thank the anonymous referees whose comments and suggestions helped us to improve this version of the paper. Also, Elena Castrillo, who revised the English version.

References

[1] H. Alaiwan, J. M. Toudic, "Recherche des semi-flots, des verrous et des trappes dans le réseaux de Petri," *T.S.I.*, , vol. 4, no. 1, pp. 104–112, 1985.

[2] Z. Banaszak and B. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Trans. Robot. and Automat.*, , vol. 6, no. 6, pp. 720–734, Dec. 1990.

[3] J. M. Colom and M. Silva, "Improving the linearly based characterization of P/T nets," *Advances in Petri Nets 90, LNCS 483.* New York: Springer-Verlag, 1991, pp. 113–145.

[4] J. Ezpeleta, J. M. Couvreur, M. Silva, "A new technique for finding a generating family of siphons, traps and st-components. Application to colored Petri Nets," *Advances in Petri Nets 1993, Lecture Notes on Computer Science, no. 674,* G. Rozenberg, Ed. New York: Springer-Verlag, 1993, pp. 126–147.

[5] J. Ezpeleta and J. Martínez, "Formal specification and validation in production plants," in *Proc. 3th. Int. Conf. Comput. Integrated Manufacturing*, 1992, pp. 64–73.

[6] ——, "Synthesis of live models for a class of FMS," in *Proc. 1993 IEEE Int. Conf. Robot. and Automat.*, 1993, pp. 557–563.

[7] J. Ezpeleta, "Analysis and synthesis of deadlock free models for concurrent systems," Ph.D. thesis, Dept. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, 1993 (in Spanish).

[8] F. Feldbrugge, "Petri net tool overview 1992," in *Advances in Petri Nets 1993, Lecture Notes on Computer Science, no. 674,* G. Rozenberg, Ed. New York: Springer-Verlag, 1993, pp. 169–209.

[9] F. S. Hsieh and S. C. Chang, Deadlock avoidance controller synthesis for flexible manufacturing systems, in *Proc. 3rd. Int. Conf. Comput. Integrated Manufacturing*, 1992, pp. 252–261.

[10] K. Lautenbach and P. S. Thiagarajan, "Analysis of a resource allocation problem using petri nets," in *1st. European Conf. on Parallel and Distributed Syst.*, J. C. Syre, Ed. 1979, pp. 1–17.

[11] K. Lautenbach, "Linear algebraic calculation of deadlocks and traps," in *Concurrency and Nets,* Voss, Genrich, and Rozenberg, Eds. New York: Springer-Verlag, 1987, pp. 315–336.

[12] T. Murata, "Petri nets: Properties, analysis and applications," in *Proc. IEEE,* vol. 77, no. 4, pp. 541–580, Apr. 1989.

[13] W. Reisig, *Petri Nets. An Introduction,* (EATCS Monographs on Theoretical Computer Science), W. Brauer, G. Rozenberg, and A. Salomaa, Eds. New York: Springer-Verlag, 1985.

[14] ——, "Deterministic buffer synchronization of sequential processes," *Acta Informatica,*, vol. 18, pp. 117–134, 1982.

[15] M. Silva, *Las redes de Petri en la Informática y la Automática.* Madrid: Editorial AC, 1985 (in Spanish).

[16] Y. Souissi, "Deterministic systems of sequential processes: A class of structured petri nets," in *Proc. 12th. Int. Conf. Applicat. and Theory of Petri Nets,*, 1991, pp. 62–81.

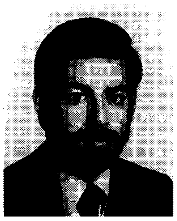
[17] N. Viswanadham, Y. Narahari, T. Johnson, "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models," *IEEE Trans. Robot. and Automat.*, vol. 6, no. 6, pp. 713–723, Dec. 1990.

Joaquín Ezpeleta received the degree on Mathematics in 1985 and the Ph.D. degree in System Engineering and Computer Sciences in 1993 from the University of Zaragoza, Spain.

For a year, he was working as a researcher at the MASI Laboratory, University of Paris 6. He is currently Associate Professor at the Department of Electrical Engineering and Computer Science of the University of Zaragoza, where he is in charge of courses on Computer Programming. His research is devoted to the modeling and analysis of concurrent systems using Petri nets. In a special way, he is currently working on the synthesis of "well behaved" models for systems (i.e., deadlock-free/live/impartial models) where a set of processes share a set of common resources.

José Manuel Colom received his Ph.D. degree in Industrial-Electrical Engineering from the University of Zaragoza, Zaragoza, Spain, in 1989.

He is currently Associate Professor at the Department of Electrical Engineering and Computer Science of the University of Zaragoza, where he is in charge of courses on Operating Systems and Computer Architecture. His research interests include modeling, qualitative and performance analysis, and implementation of parallel and distributed systems using Petri nets.



Javier Martínez received the electrical engineering degree in 1977, from the Universidad Politécnica de Madrid, and the Ph.D. degree in 1984, from the University of Zaragoza.

In 1980 he joined the Centro Politécnico Superior of the University of Zaragoza, where he is currently Professor in Computer Programming. His current research focuses the area of parallel and programming and theory and applications of Petri nets.