# Enforcing Liveness in $S^3PR$ Nets by Specialization of Resources

Asaftei Timotei[1] and José-Manuel Colom[2]
[1]*Department of Automatic Control and Industrial Informatics, Technical University "Gh. Asachi", Iasi, Romania*
[2]*Aragón Insitute of Research Engineering (I3A), University of Zaragoza, Zaragoza, Spain*

Keywords: $S^3PR$, Liveness, Pruning Graph, Minimal Siphons.

Abstract: Siphon-based control methods are often used in deadlock prevention in Petri nets models for resource allocation systems caused by the shared resources. In this article we used the properties of the minimal siphons of the $S^3PR$ and Pruning Graph to develop a new method to prevent the deadlock. This method consists in the increasing of the number of copies of a given resource type, and in the splitting of the total copies of resources in two new types of resources that will be used in a private way for each one of the two disjoint groups in which the old holder places of the original resource type are divided. The algorithm uses only structural information of the net.



## 1 INTRODUCTION

A Flexible Manufacturing System (FMS) contains multiple concurrent flows of job processes that make different products in the same time. To do this the FMS use shared resources, and these shared resources can cause the deadlock. In order to deal with this deadlock problem, many article was written trying to give a solution to this deadlock prevention problem (Chao, 2010),(Chao and Liu, 2012).

The use of Petri Net models has been proven very convenient for the formal modeling and further analysis and correction of the model of the Resource Allocation Systems. In particular, two important subclasses of Petri Nets have been defined for modeling a wide class of Resource Allocation Systems: the S3PR nets (Ezpeleta et al., 1995) and the S4PR nets (Tricas, 2003). For these classes of nets characterizations of nonliveness have been obtained that are based in the existence of certain structural objects named bad siphons that are the structural cause of the appearing of the deadlock states. A bad siphon is a set of places such that the set of input transitions to these places is a subset of the output transitions of these places. Therefore, if this set of places becomes empty of tokens, it remains empty forever and all output transitions are dead.

In this article a new algorithm which prevent deadlocks in $S^3PR$ will be developed. For this algorithm we have to construct the Pruning Graph of the net, and after that to split some siphons such that will does not exist any more strongly connected components. For this article some of the nodes of the Pruning Graph will have special properties because we consider the case with un-replicable resources. In (Asaftei and Colom, 2013) we considered that all resources in the system are replicable; i.e. a replicable resources has the property that it can be replicate (will appear a exact copy of this resource). For particular situations, because of the lack of space, money or even time, nor all resources in a systems are replicable. Nevertheless, in the logistic of an hospital, an un-replicable resource is common: the hospital do not has enough doctors or surgery rooms or medical equipment.

The article is organized as follows. Section 2 are introduced the $S^3PR$ nets. Section 3 presents the new RAS approach and in Section 4 the main results of the paper are presented: the algorithm for splitting the siphons and the liveness enforcing algorithm. Finally some conclusions are given in section 4.

## 2 DEFINITIONS AND PREVIOUS RESULTS ON $S^3PR$

In this section we recall the basic definitions, notations and previous results about the Petri Net models that we use to study the Resource Allocation Systems. These nets belongs to the $S^3PR$ class. We assume that the reader is familiar with the basic concepts and notations of Petri Nets. Next, the definition of the $S^3PR$


Timotei A. and Colom J..
Enforcing Liveness in S3PR Nets by Specialization of Resources.
DOI: 10.5220/0004275603180321
In *Proceedings of the 2nd International Conference on Operations Research and Enterprise Systems* (ICORES-2013), pages 318-321
ISBN: 978-989-8565-40-2

nets will be presented.

**Definition 1.** *((Ezpeleta et al., 1995)) An $S^3PR$ is defined recursively as follows:*

1. *An $S^2PR$ is an $S^3PR$*
2. *Let $N_i = \langle P_i \cup P_i^0 \cup P_{R_i}, F_i \rangle, i \in \{1,2\}$ 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 $N_1$ and $N_2$ are two composable $S^3PR$); then, the net $N = \langle P \cup \{P^0\} \cup P_R, T, F \rangle$ resulting of the composition of $N_1$ and $N_2$ via $P_C$ (denoted as $N = N_1 \circ N_2$) defined as follows:*
   *a. $P = P_1 \cap P_2$,*
   *b. $P^0 = P_1^0 \cap P_2^0$,*
   *c. $P_R = P_{R_1} \cap P_{R_2}$,*
   *d. $T = T_1 \cap T_2$,*
   *e. $F = F_1 \cap F_2$,*
   *is also an $S^3PR$.*

We will consider that for an acceptable marking, the nets have at least one token in each idle place and at least one token in every resource (there is at least one copy of every resource in the system). In $S^3PR$ there is no direct characterization of the liveness property but some known results present characterizations of the non-liveness of a net. In (Tricas, 2003) there are presented two theorems characterizing non-liveness of $S^4PR$ nets. These theorems are almost the same for a $S^3PR$ only with some minor modifications.

A *siphon* in a Petri net is a structural object $D$ where $D \subseteq P$ with the property:$^\bullet D \subseteq D^\bullet$. A siphon is a non-empty set of places which are connected via transitions. One property of the siphons is that once there is no token inside the places, it is not possible to gain tokens coming from the rest of the net. We say that a siphon $S$ is *minimal* if it does not contain another siphon as a proper subset. In a minimal siphon it must exists at least two places; otherwise the structure remained can not be considered a siphon.

For the algorithms that will be presented in this article, the concept of *pruning relation* between siphons will be used. The siphons we are working in this article are the minimal siphons. In (Cano et al., 2012) it is presented the pruning relation and the way the Pruning Graph is built. In (Asaftei and Colom, 2013) it is proven that the concepts of pruning and Pruning Graph can be applied to $S^3PR$ nets too.

# 3 THE NEW APPROACH FOR LIVENESS ENFORCING

The approach presented in this paper share with the previously presented liveness enforcing techniques that is based in the increasing of the number of available resources at the initial marking (Wang et al., 2010). This family of approaches is consistent with the fact that the deadlock states appearing in this kind of RAS systems are produced by problems in the allocation of resources. Taking into account that the resources are used in a conservative way and the number of customers is bounded by the initial marking of the idle places of the net, it is possible to increase the number of resources in a quantity such that the resources are not a constraint in the execution sequences of the system. In other words, the resource places become implicit places that can be removed from the net.

Nevertheless, this strategy based in the indiscriminate increase of resources, fails when the number of customers is increased. That is, the reached solution about the number of resources to make live the net for a given number of customers, must be revised if this number of customers change. The existence of this latent problem that manifests itself with the increase of customers is that the structural problems of the net remain. These structural problems are the existence of bad siphons that can be emptied after the execution of some sequences.

Deadlock prevention (Hou et al., 2010) techniques based in the addition of monitor places avoid this problem. In effect, monitor places are places, that can be interpreted as virtual resources of the system (in fact a new type of resources), that are designed with the goal of constraint the possible firing sequences that can lead to empty the siphon. In this sense, these techniques improves the techniques based in an indiscriminate increasing of resources in two different aspects. The first one is that the computation of the monitor places is done from the structural defect of the Petri Net giving rise to the appearing of deadlocks: the bad siphons of the net. The second aspect, is that the solution is based in the addition of constraints avoiding the emptiness of the siphon, i.e. increasing the number of customers, the constraints work well because the siphon cannot be emptied. But in this case there are problems with the concurrency that can be obtained in the system. In fact the monitors sequentialize the processes sharing the resources implied in the deadlock states, i.e. the concurrency is reduced. Moreover, you can increase the number of original resources but you cannot improve, in general, the concurrency of the system and its corresponding performance figures, because the constraints imposed by the monitors.

The approach proposed in this paper save the two previous drawbacks obtaining advantages of the two approaches. That is, we increase the number of re-

sources in order to enforce the liveness without the reduction of the concurrency of the system. Nevertheless, in order to propose a solution independent of the number of customers, a new type of resource is introduced (as in the case of the monitors) is such a way that the original siphons are broken and then disappear solving the structural defect of the original Petri Net. In essence, the steps to be done are:

1. Identification of a resource belonging to a bad siphon that can give rise to the appearing of a deadlock state.
2. Classification of the holder places of the selected resources in two disjoint groups. That is, the method proceeds increasing the number of resources but these new resources will be used in a private way by only one of the classified groups. This is the idea of specialization of the new resources.
3. Definition of two new resource places and removing the old one. Each one of these new resource places has one the previous sets of holder places and they are connected to the input and output transitions of the holder places in the same way than the original resource place. Observe that now there exist two different types of resources where previously existed only one type of resources. The copies of each type of resource are used in a private way by the corresponding set of holder places.
4. Introduction of the same number of tokens in each new resource place than the number of copies of resources in the original resource place.

The method can be considered as structural because the splitting of the original resource place into two new places is done in such a way that the siphon is broken or at least, in what concerns to the considered resource place the siphon has been reduced. Therefore, proceeding with all bad siphons in the same way we can obtain a net without bad siphons and then by the non-liveness theorem of $S^3PR$ nets, the admissibly marked net must be live: the method enforces liveness.

Observe, that the method increases the number of resources although they are used in a private way with respect to the way they are used in the original net. This means that the method does no cut bad markings, the method introduces new states allowing to go out from the old deadlock states. That is, the old deadlock states are reached but now there are enough resources (that they are used in a private way) to go out to a new state saving the bad scenario. In classical control theory, probably, this strategy is unsatisfactory because the strategy there is to forbid all bad states and all states that inevitably lead to a bad state. Nevertheless, in other application domains as in the hospital logistics the important goal is to reach a complete treatment for the patient, adding all needed resources to reach this goal.

For this new approach of liveness enforcing we need an algorithm which describes how the splitting is done in the $S^3PR$. The splitting is needed in the $S^3PR$ so there will appear new resources. With these resources the deadlock state is avoided. The idea is that after the construction of the initial Pruning Graph of the $S^3PR$, we need to find and eliminate all circuits from the net. An elimination in the Pruning Graph will be equivalent with a new resource in the $S^3PR$ (resource added by splitting an existing resources).

Next, an algorithm of splitting the siphon will be presented. For this algorithm we consider $r \in P_R$ is the resource we are working on.

**Algorithm 1:** The splitting of a resource.

**Input:** resource **r** which belongs at least to one circuit, $N$, the Pruning Graph of $N$;
**Output:** $N'$, **r** is substituted with $r^1$ and $r^2$;

1. Delete $r$ and all arcs between $r$ and the transitions $t_i$, $t_i \in \{r^\bullet \cup {}^\bullet r\}$;
2. Add places $r^1$ and $r^2$ in $S^3PR$;
3. Connect place $r^1$ with the net such that all the inputs of $v_i$ are the inputs of $v_i^1$: $\forall (s,r) \in E$ and $\forall p \in P_S$ such that $(t,p) \in L((s,r))$ add an arc from $r^1$ to all $t \in {}^\bullet p$ and an arc from each $t \in p^\bullet$ to $r^1$;
4. Connect place $r^2$ with the net such that all the outputs of $v_i$ are the outputs of $v_i^2$: $\forall (r,s) \in E$ and $\forall p \in P_S$ such that $(t,p) \in L((r,s))$ add an arc from $r^2$ to all $t \in {}^\bullet p$ and an arc from each $t \in p^\bullet$ to $r^2$;

For the **Algorithm 2** we use the fact that the multiplicity of the arcs from the Pruning Graph will be the same. Using this algorithm, all strongly connected components will be eliminated. One way for a graph to became acyclic is to use reversed edges. The problem of finding a set of smallest number of feedback edges is a problem called *minimum feedback arc set problem*. This is used in our algorithm when we compute the minimal set of arcs.

From the the two algorithms above, it can be seen that if the pruning graph is acyclic then the $S^3PR$ net is live. But in the most cases the pruning graph has cycles(Strongly Connected Components) which have to disappear. This can be done by removing(or change the direction) arcs from the pruning graph. Once the pruning graph is modified, some changes have to be done in the $S^3PR$ net according to *Algorithm 1*.

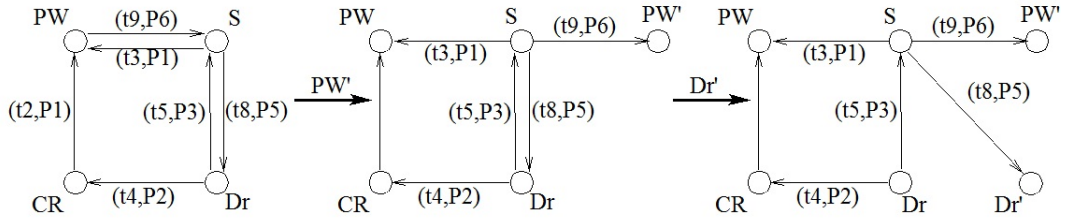


Figure 1: The evolution of the Pruning Graph according to Algorithm 1 and Algorithm 2.

**Algorithm 2:** The liveness enforcing approach.

**Input:** the Pruning Graph - PG, N, the nodes V, the edges E;

**Output:** the new Pruning Graph - $PG^{'}$, $N^{'}$, $V^{'}$;

1. Compute the pruning graph (PG) $G(V,E)$ of the $N$;
2. **While** G contains some cycles **do**
3. Compute a set of arcs $WS = e_j$ using a minimum feedback arc set algorithm;
4. **While** $WS \neq 0$ **do**
5. Take o node $v_i$ from $V$ which is connected to $e_j$ in PG;
6. Modify $N$ according to the $Algorithm1$;
7. Remove all arcs between $v_i$ and the nodes next to it in PG;
8. Remove $v_i$ and put $v_i^1$ and $v_i^2$;
9. $WS = WS \setminus \{e_j\}$
10. **Endwhile**
11. **Endwhile**

The Pruning Graph of the the $S^3PR$ net from Figure 1 from (Asaftei and Colom, 2013) is depicted in Figure 1. The $S^3PR$ net has three siphons. The goal of the approach is to split the siphons. The order we split the siphons is random (until now it does not exist an algorithm to describe the order the siphons have to be split). In this example we consider that first we split siphon $S_1$ and after that we split siphon $S_3$. After these two splits, the pruning graph become acyclic, so the algorithm stops. This is not the only solution of this problem. We can split siphon $S_3$ and after that siphon $S_1$ or siphon $S_2$ and $S_3$.

## 4 CONCLUSIONS

Exploiting the results of a previous work, where the minimal siphons has to be diminished, a new deadlock prevention method was presented. This is done by a set of process places that can be computed from a pruning relation on the minimal siphons with only one resource. This pruning relations are represented by a pruning graph. The method consists in two phases: one which work with the $S^3PR$ and one which work with its pruning graph. The main idea is to modify (by splitting resource places) the petri net so in the pruning graph will not left any strongly connected components. The main advantage for this method is that we work with high level objects (siphons). Using these objects, we improve the time and space needed to solve the deadlock problem.



## REFERENCES

Asaftei, T. and Colom, J. (2013). Enforcing liveness in s3pr nets by specialization of resources. In *2nd Int. Conf. on Operation Research and Enterprise Syst*.

Cano, E., Rovetto, C., and Colom, J. (2012). An algorithm to compute the minimal siphons in $s^4pr$ nets. In *Discrete Event Dynamic Systems. Springer Verlag. Published online*.

Chao, D. Y. (2010). A simple modification of deadlock prevention policy of s3pr based on elementary siphons. In *Transactions of the Institute of Measurement and Control*, volume 33, pages 93–115.

Chao, D. Y. and Liu, G. J. (2012). A simple suboptimal siphon-based control model of a well-known s3pr. In *Asian Journal of Control*, volume 14, pages 163–172.

Ezpeleta, J., Colom, J.-M., and Martinez, J. (1995). A petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Transactions on Robotics and Automation*, 11:173–184.

Hou, Y., Liu, D., Li, Z., and Zhao, M. (2010). Deadlock prevention using divide-and-conquer strategy for ws3pr. In *International Conference on Mechatronics and Automation (ICMA)*, pages 1635–1640.

Tricas, F. (2003). *Deadlock analysis, prevention and avoidance in sequential resource allocation systems*. PhD thesis, University of Zaragoza, Zaragoza.

Wang, S., Wang, C., and Yu, Y. (2010). An algorithm to find the condition on initial markings of resource places and job places for liveness of s3pmr. In *8th IEEE International Conference on Control and Automation*, pages 1445–1449.