Transition-Based Deadlock Detection and Recovery Policy for FMSs Using Graph Technique

YI-SHENG HUANG, National Defense University, Taiwan and National Ilan University, Taiwan
YEN-LIANG PAN, National Defense University, Taiwan and Air Force Academy, Taiwan
PIN-JUNE SU, Chung Shan Institute of Science and Technology, Taiwan

A transition-controlled deadlock detection and recovery prevention policy is presented for a subclass of Petri nets used to model flexible manufacturing systems. The subclass is called systems of simple sequential processes with resources (S^3PR). The proposed policy is different from the standard deadlock prevention policies. Instead of adding control places, this policy adds a controlled transition to solve a group of deadlocked markings that have the same graph-based property. Finally, the results of our study indicate that the proposed policy appears to be more permissive than those existing ones that add control places.

Categories and Subject Descriptors: I.6.8 [**Simulation and Modeling**]: Types of Simulation—*Discrete event*; C.4 [**Performance of Systems**]: Modeling Techniques; I.3.6 [**Computer Graphics**]: Methodology and Techniques—*Graphics data structures and data types; Languages*

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Petri net, deadlock prevention, flexible manufacturing system

ACM Reference Format:
Huang, Y.-S., Pan, Y.-L., and Su, P.-J. 2013. Transition-based deadlock detection and recovery policy for FMSs using graph technique. *ACM Trans. Embed. Comput. Syst.* 12, 1, Article 11 (January 2013), 13 pages.
DOI:http://dx.doi.org/10.1145/2406336.2406347

1. INTRODUCTION

A flexible manufacturing system (FMS) is proposed to produce a set of different types of products with varying batch size. It contains a set of computer-controlled machines and transportation systems. Various types of raw components enter it and are processed concurrently. Its limited resources among various competing jobs have to be carefully controlled and coordinated. One powerful tool for modeling it is called Petri nets (PNs). PNs are a modeling tool applicable to many FMSs. It is a promising tool for describing and studying information processing systems that are concurrent, asynchronous, distributed, and/or stochastic. When used in modeling a real-world system, PN checks whether the net model has the desired qualitative properties such as *liveness*, *boundedness*, and *reversibility*.

In FMSs, *liveness* ensures that deadlocks do not occur. *Boundedness* guarantees that the number of raw components, buffer spaces and resources is bounded. *Reversibility* enables the system to return to its initial state, thereby guaranteeing repetitive

This research was partially supported by the National Science Council of Taiwan, R.O.C. under Grant NSC 98-2918-I-606-002.
Authors' addresses: Y.-S. Huang, Department of Electronic Engineering, National Ilan University, Ilan, Taiwan, 260, R.O.C.; email: yshuang@niu.edu.t; Y.-L. Pan, Department of Avionic Engineering, Air Force Academy, Kaohsiung, Taiwan, 820, R.O.C.; email: peterpan960326@gmail.com.

DOI:http://dx.doi.org/10.1145/2406336.2406347

production. The competition for resources in an FMS may cause it to be deadlocked. In general, a deadlock occurs when the raw components are blocked while waiting for shared resources held by other production processes. Deadlock prevention [Barkaoui and Abdallah 1995; Chao 2006, 2008; Ezpeleta et al. 1995; Hu and Li 2008, 2009a, 2009b; Hu et al. 2009, 2010a, 2010b, 2010c; Huang et al. 2001, 2006; Huang 2007a, 2007b; Iordache and Mantsaklis 2003; Li and Zhou 2004, 2006, 2008a, 2008b, 2008c, 2009; Li and Zhao 2008; Li et al. 2007, 2008a, 2008b; Piroddi et al. 2008, 2009a, 2009b; Uzam 2002, 2004; Uzam and Zhou 2007; Uzam et al. 2007] in a sequential resource allocation system is a well-defined problem in FMSs. This article focuses on deadlock problems in such a system, called systems of simple sequential processes with resources (S^3PR) [Ezpeleta et al. 1995], and proposes a static deadlock recovery policy.

The proposed policy first constructs the reachability graph of the S^3PR net and finds all dead markings. Then it groups the dead markings using a graph-based technique. Finally, the policy adds a controlled transition to the net and converts the net into a live S^3PR net. The advantage of this proposed policy is that it gives the maximum permissible number of live markings in the S^3PR net.

The rest of this article is organized as follows: Section 2 presents basic definitions of PNs. Section 3 presents the proposed nets and their dead marking analysis based on the graph technique. Section 4 presents a new deadlock prevention policy and one example to show how it works. Section 5 compares it with other approaches in literature. Section 6 concludes some contributions of the article.

2. BASIC DEFINITIONS OF PETRI NETS

In this section, we provide a basic definition of the PN model. For more information, refer to Murata [1989], Marsan et al. [1995], and Peterson [1981]. A PN model is a direct bipart graph that has two types of nodes: *places* and *transitions*. Places are drawn as circles, and transitions as bars. These two types of nodes are connected by arcs. Arcs indicate which objects are changed by a certain activity and can be classified as *input* arcs, *output* arcs, and *inhibitor* arcs. Tokens are drawn as black dots within places and represent the specific value of the condition. The specific value is called the *marking* of a PN model. The PN approach is briefly summarized as follows.

Definition 1. $PN = (P, T, I, O, H, M_0)$ Marsan et al. [1995], where

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places,
$T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions, $T \cap P = \oslash$, $T \cup P \neq \oslash$,
$I : P \times T \to IN$ is an input function that defines directed arcs from places to transitions where IN is the set of non-negative integers, i.e., $IN = \{0, 1, 2, \ldots\}$,
$O : P \times T \to IN$ is an output function that defines directed arcs from transitions to places,
$H : P \times T \to IN$ is the inhibitor functions, and
$M_0 : P \to IN$ is the initial marking.

Functions I, O, and H describe the input, output and inhibitor arcs from the view point of transitions, respectively. M_0 is a function that associates a natural number to each place.

Definition 2 (*Enabling rule*). A transition t in a given Petri net is called fireable or enabled by a marking M if:

(a) for each pre-place of t, its marking is equal or greater than the weight of the arc from it to t, or
(b) *t has no pre-place.*

Definition 3 (*Firing rule*). An enabled transition t may fire. If the transition t fires, it destroys one token on each of its input places and creates one token on each of its output places.

Definition 4 (*Reachability set*). The reachability set of a PN system with initial marking M_0 is denoted $R(M_0)$, and it is defined as the smallest set of markings such that

(1) $M_0 \in R(M_0)$ and
(2) $M_1 \in R(M_0) \wedge \exists t \in T : M_1[t > M_2 \Rightarrow M_2 \in R(M_0)$,

where M_0 is the initial marking, $R(M_0)$ is the reachability set, T is the transition set, and t is the *enabled* transition in the transition set.

Definition 5 (*Reachability Graph (RG)*). A RG of PN is the label-directed multi-graph whose set of nodes is reachability sets (RS) of a PN system and whose set of arcs A is defined as follows:

(1) $A \subseteq RS \times RG \times T$ and
(2) $\langle M_i, M_j, t\rangle \in A \Leftrightarrow M_i[t{>}M_j$ where $M_i, M_j \in R(M_0)$,

where M_0 is the initial marking, and the notation $\langle M_i, M_j, t\rangle$ is used to indicate that connects a marking M_i connects to M_j via an arc t.

Definition 6 (*state equation*). The jth entry of $M_k$denotes the number of tokens in place j in M_k and M_k is the marking reached after the kth firing of a transition sequence. The kth firing vector u_k is an $n \times 1$ column vector of $n-1$ zeros and one nonzero entry, an entry 1 in the ith position indicating that transition i fires at the kth firing. Because the ith row of the incidence matrix A denotes the change in the marking as a result of the firing transition i, one can write the following state equation:

$$M_k = M_{k-1} + A^{\mathrm{T}} u_k, k = 1, 2 \ldots$$

Definition 7 (*Necessary reachability condition*). Suppose that a destination marking M_d is reachable from M_0through a firing sequence $u_1 u_2 \ldots u_d$. Writing the state equation for $i = 1, 2 \ldots d$ and summing them, one obtains

$$M_d = M_0 + A^T \sum_{k=1}^{d} u_k,$$

which can be rewritten as

$$A^T x = \Delta M,$$

where $\Delta M = M_d - M_0$ and $x = \sum_{k=1}^{d} u_k$. Here, x is an $n \times 1$ column vector of nonnegative integers and is called the firing-count vector. The ith entry of x denotes the number of times that transition i fires to transform M_0 to M_d.

3. DEADLOCK ANALYSIS OF S^3PR

In this section an S^3PR model is first described. It is composed of a set of state machines holding and releasing shared resources. We follow the original definition from Ezpeleta et al. [1995].

Definition 8 (*[Ezpeleta et al. 1995]*). S^3PR is defined as the union of a set of nets sharing common places, called resource places in this study, P_{Ri}, such that $N_i = (P_i \cup$

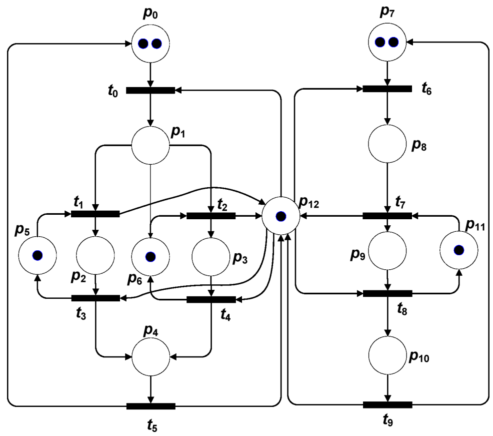

Fig. 1. An example of an S^3PR model.

$P_i^0 \cup P_R$, T_i, F_i), where $P_i \neq \oslash$; $P_i^0 \notin P_i$; $P_{Ri} \neq \oslash$; $\forall p \in P_i$, $({}^{\bullet\bullet}p \cap P_R = p^{\bullet\bullet} \cap P_R) \wedge (|p^{\bullet\bullet} \cap P_R|1)$. $N_i^{'}$ is a strongly connected state machine where $N_i^{'} = (P_i \cup P_i^0, T_i, F_i)$ is the result after P_{Ri} is removed from N_i. Every circuit of $N_i^{'}$ contains the place P_i^0 and any two $N_i^{'}$ can be composed when they share a set of common resource places. The place in P_i^0 is the *process-idle* place of N_i. The places in P_i and P_R are called *operation* and *resource* places, respectively, and transitions in $(P_i^0)^{\bullet}$ are called *source* transitions of an S^3PR net.

An S^3PR model, depicted in Figure 1 with the initial marking M_0 $(2p_0 + p_5 + p_6 + 2p_7 + p_{11} + p_{12})$, has the important characteristic that only one shared resource is allowed at each operation state. The resource used by the state is released when the system moves to the next one.

According to Definition 8, p_0 and p_7 are process-idle places. Places p_5, p_6, p_{11}, and p_{12} are resource ones, and the other places are operation ones.

In this study, the authors use the reachability-graph-based definition to define the deadlock condition of the S^3PR net. For further details, see references Murata [1989] and Marsan et al. [1995]. A deadlock situation involving an S^3PR net has following definition.

Definition 9 (*dead marking*). Given a marked PN (N, M_0), $M \in R(M_0)$ is called a deadlock marking if no transition is enabled in M, where N is the PN system structure, M_0 is the initial marking and $R(M_0)$ is the reachable marking from M_0.

From Figure 1 and according to Definition 9, there are four dead markings in the reachability graph of this S^3PR model. The reachability graph of Figure 1 is shown in Figure 2, which has 32 markings in it. From Figure 2, we can see that four dead states, listed in Table I, are markings M_{20}, M_{22}, M_{24}, and M_{26}.

In the reachability graph, these four states, i.e., M_{20}, M_{22}, M_{24}, and M_{26} are the end of reachability graph. Here, the four states are also called dead states. However, the four dead markings belong to two different siphons. We can establish a new method to deal with the dead states in S^3PR models, which relies on a graph-based technique. Examining dead states using Definition 8 as a guideline, we can find the connections between these four dead markings. Based on the S^3PR definition, we can divide places of Figure 1 into three groups: process-idle place, operation place, and resource place.

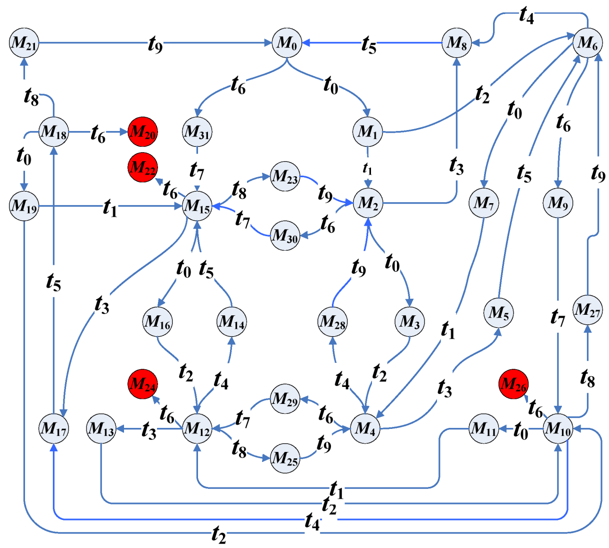

Fig. 2. The reachability graph of Figure 1.

Table I. The Dead States in the Reachability Graph of Figure 1

Dead states	Marking
M_{20}	(2, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0)
M_{22}	(0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0)
M_{24}	(1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0)
M_{26}	(1, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0)

In Figure 1, the process-idle places are p_0 and p_7, and operation ones are p_1, p_2, p_3, p_4, p_8, p_9, and p_{10}. Moreover, these places of process-idle and operating places can further be divided into two processes. Places p_7, p_8, p_9, and p_{10} belong to the first one, and p_0, p_1, p_2, p_3, and p_4 belong to the second one. Hence, we need the following definition.

Definition 10. A process place set of an S^{3}PR net is defined as $Q = P_i \cup \{P_i^0\}$, where 1) P_i^0 is the process-idle place; 2) P_i is the operation place, and 3) $q \subseteq Q$,$q_i \cap q_j = \emptyset$.

Definition 11. The subplace set $S_{qi} \subseteq q_i$, $S_{qj} \subseteq q_i$ and $S_{qi} = S_{qj}$ in dead markings of a S^{3}PR net, and also it exists a lived controlled transition t which $I(t) = S_{qi} = S_{qj}$. Then it is said that these dead markings are caused by the same process q_i.

Definitions 10 and 11 can help us to find the relationship between those dead markings. This article indicates that there is always having same marked place in those dead markings. And those dead markings which are always having same marked place can be connected to one manufacturing process.

For example, the connections among M_{20}, M_{22}, M_{24}, and M_{26} are found. They are all caused by the same process q_1. They have the same subplace set $S_{q1} = p_8 + p_9$ because they are caused by the same process. In the next section, we show how to develop a deadlock prevention policy based on Definitions 10 and 11.

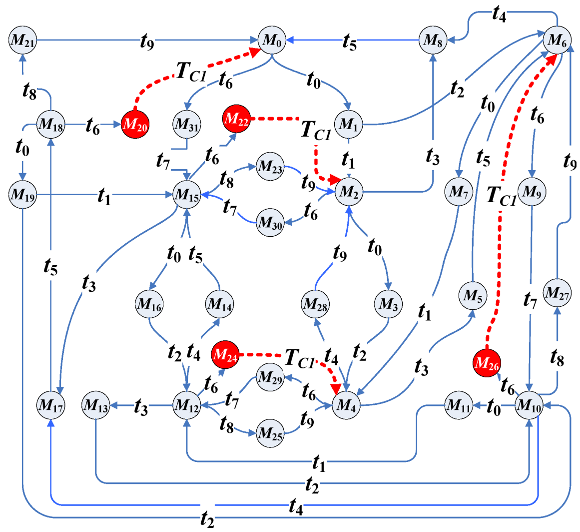

Fig. 3. A live reachability graph of Figure 1.

4. TRANSITION-BASED DEADLOCK PREVENTION POLICY FOR S^{3}PR

Based on the discussions in Section 3, this section proposes a static deadlock prevention policy based on Definitions 10 and 11. This method uses a transition-control technique. The policy begins by finding the dead states in the reachability graph of the S^{3}PR system with the initial marking M_0. Then, the proposed policy adds a control transition to the original net, such that a dead marking becomes a live marking. The policy presented in this study is very different from standard approach in Barkaoui and Abdallah [1995], Chao [2006, 2008], Ezpeleta et al. [1995], Huang et al. [2001, 2006], Huang [2007a, 2007b], Iordache and Mantsaklis [2003], Li and Zhao [2008], Li and Zhou [2004, 2006, 2008a, 2008b, 2008c], and Li et al. [2007, 2008a, 2008b], which added control places to prevent siphons from becoming unmarked. Instead, our proposed policy uses control transitions to convert dead markings into live ones. Using this policy raises an important question: How do we find the control transition? In our study, we use a graph-based technique to find control transitions to convert dead markings, which are caused by the same process, into live markings.

Taking Figure 1 as an example, this net has four dead markings (i.e. M_{20}, M_{22}, M_{24}, and M_{26}). Because they are caused by the same process, we can resolve them with one control transition T_{c1}, as shown in Figure 3. Firing T_{c1} leads them to four different live markings. It must be noted that controlled transition T_{c1} is not unique. Next it is shown how another control transition can be found for this case.

For more detailed information, we obtain $M' = M_{dead} + [O(T_c) - I(T_c)] = p_5 + p_6 + 2p_0 + O(T_c)$ where $M', M_{dead}, O(T_c)$, and $I(T_c)$ represents a live marking, dead marking, the output place of control transition T_c, and the input place of control transition T_c, respectively. Here, $S_{q1} = p_8 + p_9$, $S_{q1} = p_8 + p_9 = I(T_{c1})$ and dead marking $M_{20} = 2p_0 + p_5 + p_6 + p_8 + p_9$. In this case, we can find two output transitions $O(T_{ci})$ that are valid control transitions, which is shown in Table II.

In this study, we chose T_{c1} as the control transition. Notice that T_{c1} or T'_{c1} can obtain the same control result. Hence, a live S^{3}PR system of Figure 1 can be found in Figure 4.

Adding this control transition T_{c1} allows this deadlock S^{3}PR system to become a live system. We summarize our deadlock prevention policy to show the detailed deadlock

Table II. Two Valid Control Transitions

Control transition	$O(T_{ci})$	$I(T_{ci})$
T_{c1}	$2p_7 + p_{11} + p_{12}$	$p_8 + p_9$
T'_{c1}	$p_7 + p_{10} + p_{11}$	$p_8 + p_9$

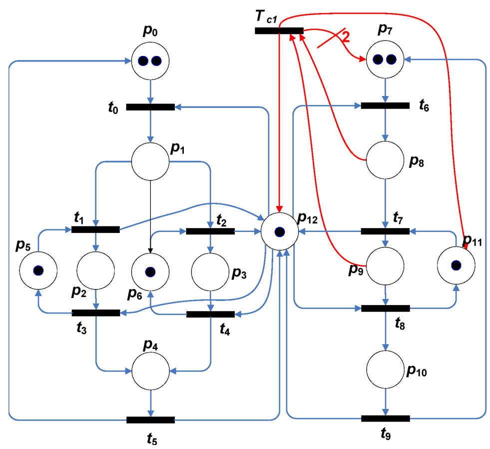

Fig. 4. A live S^{3}PR system of Figure 1.

Table III. Our Deadlock Prevention Policy

Algorithm: Deadlock Prevention Policy Input: a deadlock S^{3}PR model. Output: a live S^{3}PR model.
Step 1: Construct the reachability graph of a S^{3}PR system.
Step 2: Locate all dead markings in the reachability graph.
Step 3: Classify all dead markings.
Step 4: Decide how many control transitions are needed.
Step 5: Find *input* place of control transition $I(T_{ci})$.
Step 6: Find *output* place of control transition $O(T_{ci})$.
Step 7: Add control transitions T_{ci} on the original net.
Step 8: If livelock existing Then go to Step 4
Step 9: Obtain a live S^{3}PR model.

prevention policy in Table III. Moreover, a more complex flexible manufacturing system's PN model is examined below.

To demonstrate the new deadlock prevention policy, we employ an example which is taken Li and Zhou [2008b] and shown in Figure 5. The example is an S^{3}PR system with the initial marking $M_0 = 6p_0 + p_{13} + p_{14} + p_{15} + p_{16} + p_{17} + p_{18} + 6p_7$.

Step 1. Construct the reachability graph of the S^{3}PR system.

In this study, we used TINA [Berthomieu and Vernadat 2006] to build the S^{3}PR system. In this system, 282 markings can be found in the reachability graph.

Step 2. Locate all dead markings in the reachability graph.

According to Definition 9, the net of Figure 5 has 16 dead markings in the reachability graph. The total number of dead markings is listed in Table IV.

Step 3. Classify all dead markings based on Definitions 10 and 11.

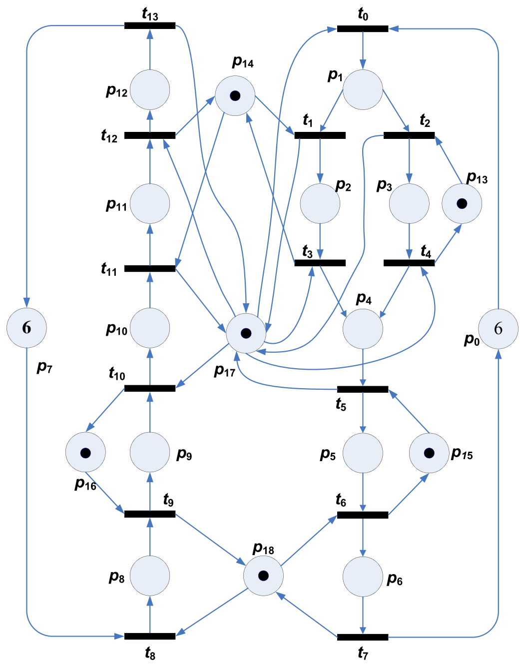

Fig. 5. A S³PR model taken from Li and Zhou [2008b].

Table IV. Dead Marking of Figure 5

Dead states	Marking
M_8	(3, 1, 1, 1, 0, 0, 0, 4, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0)
M_{18}	(2, 1, 1, 1, 0, 1, 0, 4, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0)
M_{49}	(4, 0, 1, 1, 0, 0, 0, 3, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0)
M_{53}	(3, 0, 1, 1, 0, 1, 0, 3, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0)
M_{75}	(3, 0, 0, 1, 1, 1, 0, 4, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0)
M_{76}	(3, 0, 1, 0, 1, 1, 0, 4, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0)
M_{83}	(4, 1, 0, 1, 0, 0, 0, 3, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0)
M_{84}	(3, 1, 0, 1, 0, 1, 0, 3, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0)
M_{100}	(5, 0, 1, 0, 0, 0, 0, 3, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0)
M_{104}	(4, 0, 1, 0, 0, 1, 0, 3, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0)
M_{134}	(4, 0, 0, 0, 1, 1, 0, 4, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0)
M_{171}	(5, 0, 0, 1, 0, 0, 0, 2, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0)
M_{175}	(4, 0, 0, 1, 0, 1, 0, 2, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0)
M_{205}	(6, 0, 0, 0, 0, 0, 0, 2, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0)
M_{209}	(5, 0, 0, 0, 0, 1, 0, 2, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0)
M_{234}	(4, 0, 0, 0, 1, 1, 0, 3, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0)

According to Definitions 10 and 11, these 16 dead markings can be divided into four groups. The first group has M_8, M_{18}, M_{75}, M_{76}, and M_{134} due to tokens in the places p_9-p_{12} of above markings are all the same. Similarly, the second one has M_{49}, M_{53}, M_{100}, and M_{104}. The third one has M_{83}, M_{84}, and M_{234}. The final group has M_{171}, M_{175}, M_{205}, and M_{209}.

Step 4. Decide how many control transitions are needed.

From these four groups, we can see that four control transitions are needed. Hence, we call them as T_{c1}–T_{c4} for groups 1–4, respectively.

Step 5. Find input place of control transition $I(T_{ci})$.

Table V. $O(T_{ci})$ *output* Place of Control Transitions

Output place	associated places
$O(T_{c1})$	$6p_7 + p_{16} + p_{18}$
$O(T_{c2})$	$6p_7 + p_{16} + p_{17} + p_{18}$
$O(T_{c3})$	$6p_7 + p_{14} + p_{16} + p_{18}$
$O(T_{c4})$	$6p_7 + p_{14} + p_{16} + p_{17} + p_{18}$

Table VI. Four Controlled Transitions of Figure 5

Controlled transition	$O(T_{ci})$	$I(T_{ci})$
T_{c1}	$2p_7 + p_{16} + p_{18}$	$p_8 + p_9$
T_{c2}	$3p_7 + p_{16} + p_{17} + p_{18}$	$p_8 + p_9 + p_{10}$
T_{c3}	$3p_7 + p_{14} + p_{16} + p_{18}$	$p_8 + p_9 + p_{11}$
T_{c4}	$4p_7 + p_{14} + p_{16} + p_{17} + p_{18}$	$p_8 + p_9 + p_{10} + p_{11}$

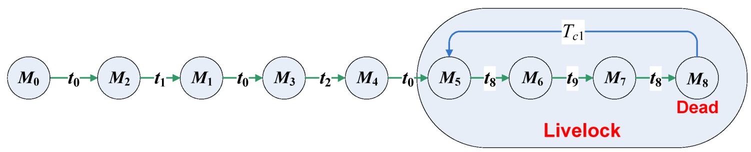

Fig. 6. Part of reachability graph Figure 5.

According to Definitions 11, we can find $I(T_{c1}) = 4p_7 + p_8 + p_9$, $I(T_{c2}) = 3p_7 + p_8 + p_9 + p_{10}$, $I(T_{c3}) = 3p_7 + p_8 + p_9 + p_{11}$ and $I(T_{c4}) = 2p_7 + p_8 + p_9 + p_{10} + p_{11}$.

Step 6. Find output place of control transition $O(T_{ci})$.

It must be noted that the output place of the control transition $O(T_{ci})$ should not contain the operation place, which already exists in $I(T_{ci})$.

By applying Definition 7, $M' = M_{dead} + [O(T_{ci}) - I(T_{ci})]$, we can obtain $O(T_{ci})$ in the reachability graph, as shown in Table V.

Step 7. Add control transitions T_{ci} on the original net.

The detailed description of the four control transitions is illustrated in Table VI.

Definition 12. $x \in Z^m$, $x \neq 0$, is a T-invariant if $A^T x = 0$, where x is called the firing-count vector, matrix A denotes the change of the marking, and Z^m is the firing set of x.

Based on Definition 12, one can infer that a livelock is formed if a reachability graph is not a T-*invariant*.

The livelock means S^3PR is dead in a zone. If a control transition is a valid one, then it will not form a livelock in the S^3PR system. For instance, in Table VI, we find that T_{c1} is not a valid transition and thus leads to a livelock after this control transition is added on the net. Part of the reachability graph of Figure 5 is shown in Figure 6. This figure describes the consequence of using the invalid control transition.

In Figure 6, we can see initial marking M_0 through firing sequence $t_0t_1t_0t_2t_0t_8t_9t_8$ to dead marking M_8. If the wrong controlled transition T_{c1} is used, then markings M_5, M_6, M_7, and M_8 become a livelock. Because T_{c1} is not valid, we need to go back to step 4.

The seven control transitions can solve the 16 dead markings of Figure 5. A live system of Figure 5 with seven controlled transitions will have the 282 maximally permissive markings. The live S^3PR system of Figure 5 can be seen in Figure 7.

Table VII. Valid Control Transitions of Figure 5

T_{ci}	$O(T_{ci})$	$I(T_{ci})$
T_{c1}	$3p_0 + p_{13} + p_{15} + p_{16} + p_{17} + p_{18} + 2p_7$	$p_3 + p_4 + p_5 + p_8 + p_9$
T_{c2}	$3p_7 + p_{16} + p_{17} + p_{18}$	$p_8 + p_9 + p_{10}$
T_{c3}	$3p_7 + p_{14} + p_{16} + p_{18}$	$p_8 + p_9 + p_{11}$
T_{c4}	$4p_7 + p_{14} + p_{16} + p_{17} + p_{18}$	$p_8 + p_9 + p_{10} + p_{11}$
T_{c5}	$3p_0 + p_{13} + p_{14} + p_{16} + p_{17} + p_{18} + 2p_7$	$p_1 + p_2 + p_3 + p_8 + p_9$
T_{c6}	$3p_0 + p_{14} + p_{15} + p_{16} + p_{17} + p_{18} + 2p_7$	$p_2 + p_4 + p_5 + p_8 + p_9$
T_{c7}	$2p_0 + p_{15} + p_{16} + p_{17} + p_{18} + 2p_7$	$p_4 + p_5 + p_8 + p_9$

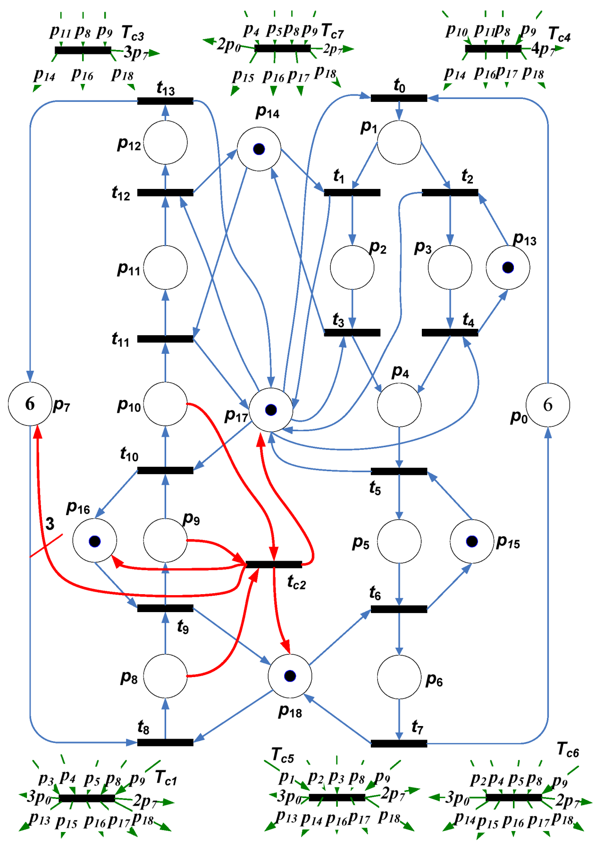

Fig. 7. A live S^3PR system of Figure 5.

5. COMPARISON WITH PREVIOUS STUDIES

This section presents a comparison between the proposed deadlock prevention policy and other approaches in previous studies (Ezpeleta et al. [1995], Huang [2007b], Li et al. [2007], Uzam and Zhou [2007]). In particular, the policies of the approaches (Ezpeleta et al. [1995], Huang [2007b], Li et al. [2007], Uzam and Zhou [2007]), all adds control places to control siphons. However, this proposed policy uses control transitions to resolve deadlocks. Here, an example taken from Ezpeleta et al. [1995] is used to compare our proposed policy with their work. The example is depicted in Figure 8.

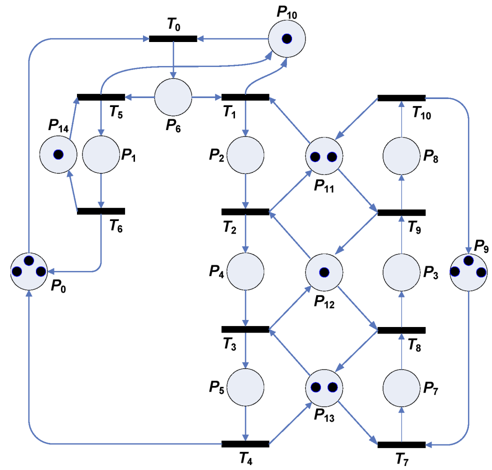

Fig. 8. Example S^3PR net from Ezpeleta et al. [1995].

Table VIII. Dead Marking of Figure 8

Dead states	marking
M_{60}	(0, 0, 2, 1, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 1)
M_{61}	(0, 1, 2, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 0)
M_{62}	(1, 0, 2, 1, 0, 0, 0, 2, 0, 0, 1, 0, 0, 0, 1)
M_{81}	(0, 0, 2, 0, 1, 0, 0, 2, 0, 1, 1, 0, 0, 0, 1)

Table IX. 7 Valid Controlled Transitions of Figure 8

T_{ci}	$O(T_{ci})$	$I(T_{ci})$
T_{c1}	$2p_{13} + p_{12}+3p_9$	$p_3+ 2p_7$
T_{c2}	$p_0 + p_{11} + p_{12}+ 2p_5 + p_8 + p_9$	$2p_2 + p_4+ 2p_7$

The initial marking of Figure 8 is $M_0 = 3p_0 + p_{10} + 2p_{11} + p_{12} + 2p_{13} + p_{14} + 3p_9$. This S^3PR net has 257 live markings and four dead markings. The four dead markings are M_{60}, M_{61}, M_{62}, and M_{81} and listed in Table VIII.

According to our prevention policy, these four dead markings can be divided into two groups. Group 1 is M_{60}, M_{61}, and M_{62}, and Group 2 is M_{81}. It suggests that we only need two control transitions to solve the four dead markings. After applying our prevention policy, we locate two control transitions and they are shown in Table IX.

The live S^3PR net of Figure 8 is depicted in Figure 9.

The comparison of our proposed policy with previous policies can be found in Table X.

Table X shows that our policy leads to more permissible compared to previous policies. A total of 261 markings can become live markings by using the transition-controlled deadlock prevention policy presented in this article.

6. CONCLUSIONS

This study tries to present a deadlock detection and recovery policy for a typical subclass of FMSs called S^3PR. The net is initially proposed by Ezpeleta et al. [1995]. The major strategy of this proposed article is to add control transitions on the net, which converts the net into a live one. This transition-controlled method is novel and much different from the place-controlled deadlock prevention policy in existing literature. The proposed policy seems to have more permissible live markings than other

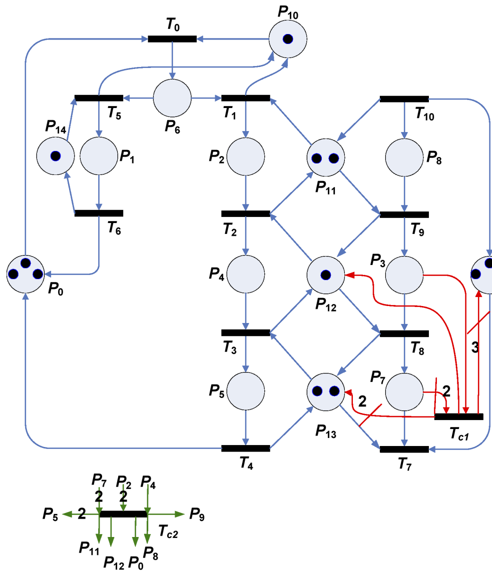

Fig. 9. Live S^{3}PR net of Figure 8.

Table X. Comparison Results with Past Control Policies

Control policy	Additional places	Token number	Additional transitions	Tangible states
Ezpeleta et al. [1995]	3	2, 2, 4	0	155
Li et al. [2007]	3	1, 1, 3	0	106
Uzam and Zhou [2007]	3	2, 3, 2	0	232
Huang [2007b]	3	2, 2, 7	0	212
The proposed one	0	0	2	261

place-controlled policies although our proposed algorithm is still NP complete since we need to check the reachability graph.

REFERENCES

Barkaoui, K. and Abdallah, B. 1995. Deadlock avoidance in FMS based on structural theory of Petri nets. In *Proceedings of the INRIA/IEEE Symp. Emerging Technologies for Factory Automation.* 499–510.

Berthomieu, B. and Vernadat, F. 2006. Time Petri nets analysis with TINA. In *Proceedings of the 3rd International Conference on the Quantitative Evaluation of Systems.* IEEE, 123–124.

Chao, D. Y. 2006. Computation of elementary siphons in Petri nets for deadlock control. *Comput. J. 49*, 4, 470–479.

Chao, D. Y. 2008. Incremental approach to computation of elementary siphons for arbitrary S3PR. *IET Control Theory Appl. 2*, 2, 168–179.

Ezpeleta, J., Colom, J. M., and Martinez, J. 1995. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Rob. Autom. 11*, 173–184.

Hu, H. S. and Li, Z. W. 2008. An optimal-elementary-siphons based iterative dead-lock prevention policy for flexible manufacturing systems. *Int. J. Adv. Manuf. Technol. 38*, 3–4, 309–320.

Hu, H. S. and Li, Z. W. 2009. Efficient deadlock prevention policy in automated manufacturing systems using shared resources. *Int. J. Adv. Manuf. Technol. 40*, 5–6, 566–571.

Hu, H. S. and Li, Z. W. 2009. Modeling and scheduling for manufacturing grid workflows using timed Petri nets. *Int. J. Adv. Manuf. Technol. 42*, 5–6, 553–568.

Hu, H. S., Zhou, M. C., and Li, Z. W. 2009. Liveness enforcing supervision of video streaming systems using non-sequential Petri nets. *IEEE Trans. Multimedia 11*, 8, 1457–1465.

Hu, H. S., Zhou, M. C., and Li, Z. W. 2010a. Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets. *IEEE Trans. Autom. Sci. Eng.*

Hu, H. S., Zhou, M. C., and Li, Z. W. 2010b. Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets. *IEEE Trans. Autom. Sci. Eng.*

Hu, H. S., Zhou, M. C., and Li, Z. W. 2010c. Ratio and supervisory control for automated manufacturing systems using Petri nets. *IEEE Trans. Syst. Man Cybern. A Syst. Human.*

Huang, Y. S. 2007a. Deadlock prevention for flexible manufacturing systems in sequence resource allocation systems. *J. Inf. Sci. Eng. 23*, 1, 215–231.

Huang, Y. S. 2007b. Design of deadlock prevention supervisors using Petri nets. *Int. J. Adv. Manuf. Technol. 35*, 3–4, 349–362.

Huang, Y. S., Jeng, M. D., Xie, X. L., and Chung, S. L. 2001. Deadlock prevention policy based on Petri nets and siphons. *Int. J. Prod. Res. 39*, 3, 283–305.

Huang, Y. S., Jeng, M. D., Xie, X. L., and Chung D. S. 2006. Siphon-based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. A Syst. Human 36*, 6, 2152–2160.

Iordache, M. A. and Mantsaklis, P. J. 2003. Design of T-liveness enforcing supervisors in Petri nets. *IEEE Trans. Automat. Contr. 48*, 11, 1962–1974.

Li, Z. W. and Zhao, M. 2008. On controllability of dependent siphons for deadlock prevention in generalized Petri nets. *IEEE Trans. Syst. Man Cybern. A Syst. Human 38*, 2, 369–384.

Li, Z. W. and Zhou, M. C. 2004. Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. A Syst. Human 34*, 1, 38–51.

Li, Z. W. and Zhou, M. C. 2006. Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets. *IEEE Trans. Ind. Inform. 2*, 4, 313–325.

Li, Z. W. and Zhou, M. C. 2008a. A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. A Syst. Human 38*, 2, 173–188.

Li, Z. W. and Zhou, M. C. 2008b. On siphon computation for deadlock control in a class of Petri nets. *IEEE Trans. Syst. Man Cybern. A Syst. Human 38*, 3, 667–679.

Li, Z. W. and Zhou, M. C. 2008c. Control of elementary and dependent siphons in Petri nets and their application. *IEEE Trans. Syst. Man Cybern. A Syst. Human 38*, 1, 133–148.

Li, Z. W. and Zhou, M. C. 2009. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach.* Springer-Verlag, Berlin.

Li, Z. W., Zhou, M. C., and Uzam, M. 2007. Deadlock control policy for a class of Petri nets without complete siphon enumeration. *IET Control Theory Appl.* 1594–1605.

Li, Z. W., Zhou, M. C., and Jeng, M. D. 2008a. A maximally permissive deadlock prevention policy for fms based on Petri net siphon control and the theory of regions. *IEEE Trans. Autom. Sci. Eng. 5*, 1, 182–188.

Li, Z. W., Uzam, M., Zhou, M. C. 2008b. Deadlock control of concurrent manufacturing processes sharing finite resources. *Int. J. Adv. Manuf. Technol. 38*, 787–800.

Marsan, M. A., Balbo, G. Conte, G., Donatelli, S., and Franceschinis, G. 1995. *Modeling with Generalized Stochastic Petri Nets.* Wiley Series in Parallel Computing. John Wiley and Sons.

Murata, T. 1989. Petri nets: Properties, analysis, and applications. *Proc. IEEE 71*, 541–580.

Peterson, J. L., 1981. *Petri Net Theory and the Modeling of Systems.* Prentice Hall, Englewood Cliffs, NJ.

Piroddi, L., Cordone, R., and Fumagalli, I. 2008. Selective siphon control for deadlock prevention in Petri nets. *IEEE Trans. Syst. Man Cybern. A Syst. Human 38*, 6, 1337–1348.

Piroddi, L., Cossalter, M., and Ferrarini, L. 2009a. A resource decoupling approach for deadlock prevention in FMS. *Int. J. Adv. Manuf. Technol. 40*, 1–2, 157–170.

Piroddi, L., Cordone, R., and Fumagalli, I. 2009b. Combined siphon and marking generation for deadlock prevention in Petri Nets. *IEEE Trans. Syst. Man Cybern. A Syst. Human 39*, 3, 650–661.

Uzam, M. 2002. An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. *Int. J. Adv. Manuf. Technol. 19*, 192–208.

Uzam, M. 2004. The use of Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems. *Int. J. Adv. Manuf. Technol. 23*, 204–219.

Uzam, M. and Zhou, M. C. 2007. An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. A Syst. Human 37*, 3, 362–371.

Uzam, M., Li, Z. W., and Zhou, M. C. 2007. Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS. *Int. J. Adv. Manuf. Technol. 35*, 150–168.

Received March 2010; accepted June 2010