DANKO KEZIĆ, Ph.D.
E-mail: danko.kezic@pfst.hr
ANITA GUDELJ, M.Sc.
E-mail: anita.gudelj@pfst.hr
University of Split, Faculty of Maritime Studies
Zrinsko-Frankopanska 38, HR-21000 Split,
Republic of Croatia

Science in Traffic and Transport
Preliminary communication
Accepted: Sep. 23, 2009
Approved: May 17, 2010

DESIGN OF RIVER SYSTEM DEADLOCK AVOIDANCE SUPERVISOR BY USING PETRI NET

ABSTRACT

Advanced function of the computer-based river traffic management system should automatically predict and prevent possible conflict and deadlock states between vessels by using adequate control policy (supervisor). This paper proposes a formal method for calculating maximally permissive deadlock prevention supervisor. To model the river system, the authors use a class of Petri net suitable for describing multiple re-entrant flowlines with disjoint sets of resources, jobs and control places, and matrix-based formal method to analyze the system. By using matrix algebra, the structural characteristics of the Petri net (circular waits, P-invariants, critical siphons and subsystem, key resource) have been analyzed and the steps for supervisor design proposed. The first and the second level deadlocks can be avoided by maintaining the number of tokens in the critical subsystems and ensuring that the key resource would not be the last available resource in the system. The derived supervisor has been verified by a computer simulation using MATLAB environment.

KEYWORDS

traffic management system, deadlock avoidance, discrete event system, Petri net

1. INTRODUCTION

River traffic system is a natural vessel passage which is used to connect two lakes, seas or oceans. Such a river system, for example, enables the connection of two water surfaces that have different levels of water. The vessels passing through the river system in opposite directions use various resources (canals, locks, basins). Some of these resources are non-shared (resource that can be occupied by the vessels moving in only one direction), and others are shared resources (resources that can be occupied by the vessels moving in different directions). The number of vessels in the resources is limited by a rule.

Some of the problems that need to be solved are: a) How to control traffic in a way that vessels moving in opposite directions make as few stops as possible during the passage through the river traffic system (maximally permissive control policy)? b) How to resolve possible conflicts in case when more vessels try to acquire shared resource at the same time? c) How to avoid possible deadlocks in the dense traffic?

The vessels moving through the river traffic systems can generally be described as a set of discrete states and events (discrete event dynamic systems - DEDS). These events and states are normally observed by the river traffic management system (RTMS) which receives data from vessels automatic identification systems (AIS), using wireless data communication. Some of these states, such as conflicts and deadlocks are undesirable (even dangerous). To avoid these forbidden states, the RTMS software should have the so-called “supervisor”. The supervisor monitors vessels’ movements and forbids only such moving that can lead to forbidden states or, in other words, restricts the set of all reachable discrete states in the system to the set of allowable states (without forbidden states). The supervisor can simply advise the man responsible for traffic regulation, or can be connected to the computer-based traffic lights system which is driven by the RTMS software. In both cases, the supervisor must apply the appropriate control policy. In this paper the Petri net theory has been used, a well known tool for analysing DEDS, to calculate the desired control policy and design an appropriate supervisor.

Many authors have also tried to solve the deadlock problem by Petri nets. Barkaoui [1] developed the method of deadlock prevention by control places. [1] Also Ezpelta et al. [2] developed an algorithm for deadlock prevention for the ordinary and conservative S^3PR class of Petri nets. The work by Ezpelta et al. [2] is usually considered to be the first that used structural analysis to design monitor-based livness-enforcing Petri net supervisor for the flexible manufacturing systems (FMS). Lautenbach [3] investigated the algorithm for finding the minimal siphons inside the net as well as the algorithm for deadlock prevention by control places for ordinary Petri nets which do not contain source places. Further, Lewis [4] developed an efficient algorithm for deadlock prevention in the specific class of Petri nets that describes FMS. A deadlock prevention which uses iterative siphon control method is described in Kezić [5] and Iordache [6] and Huang [7]. Similar, but older approaches, can be found in Barkaoui et al. [8,9] and Tricas et al. [10] The latest iterative deadlock prevention policy, T-policy for short, is reported by Tricas et al. [11]

The goal is to find maximally permissive deadlock prevention supervisor which stops vessels only in case of immediate dangerous situations in dense traffic, and this paper describes the necessary steps to achieve this. Modelling the river traffic system is the first task. The class of multiple re-entrant flowlines Petri net (MRF1PN), subclass of *flowline system Petri net* (FPN) is used in this paper. FPN is a class of Petri net which is basically designed for analysing finite buffer multi class re-entrant flowline systems (MRF) - a large class of flexible manufacturing systems (FMS) [12]. MRF1PN contains some specific properties, which will be discussed in section II.

Next, we find simple and cyclic circular waits between the resources in the MRF1PN. The set of places called critical siphons, which are important in deadlock prevention and finally critical subsystems in the net are found. To avoid a deadlock, it is necessary to control markings in every critical subsystem. This can be achieved by adding additional control places or a supervisor which blocks firing of particular transitions and restricts the number of tokens in critical systems by adding and removing tokens in control places. But still the deadlock can exist if the system is irregular and contains the so-called key resource. The final step is to find key resources. The supervisor must ensure that the key resource is not the last available resource in the net.

The paper is organized as follows: section II reviews the basics of P/T Petri net properties and also describes notations which are used throughout the paper. In this section special attention is dedicated to class MRF1PN, subclass of FPN Petri net. Section III describes the matrix approach of supervisor design. Finally, Section IV shows a case study of a river system traffic control. The moving of vessels and adequate control policy of the supervisor is verified using P-timed Petri nets and computer simulation.

2. PETRI NETS AND DEADLOCK AVOIDANCE

Place - transition P/T Petri net is a 6-tuple: [13]

$$Q = (P, T, \mathbf{I}, \mathbf{O}, \mathbf{M}, \mathbf{m}_0) \qquad (1)$$

where:

$P = \{p_1, p_2, \ldots, p_m\}$ - set of places,

$T = \{t_1, t_2, \ldots, t_n\}$ - set of transitions,

$P \cap T = \varnothing$,

$\mathbf{I}_{(nxm)}: T \times P \to \{0,1\}$ – an input incidence matrix,

$\mathbf{O}_{(nxm)}: T \times P \to \{0,1\}$ – an output incidence matrix,

$w: [\mathbf{I} \vdots \mathbf{O}] \mapsto \mathbf{M}$ – is a weight function, where $\mathbf{M}$ is a matrix of the same type as $[\mathbf{I} \vdots \mathbf{O}]$, with items declared with

$$(\mathbf{M})_{ij} = w(t_i, p_j), w(t_i, p_j) \in \{0, 1, 2, 3 \ldots\},$$
$$i = 1 \ldots n, j = 1 \ldots m.$$

$\mathbf{m}_0$ - initial marking.

A Petri net is represented by a bipartite muligraph containing two types of nodes, places (drawn as circles) and transitions (drawn as bars) connected with directed arcs. Places and transitions $v \in P \cup T$ and denote states and events in the DEDS. Given any node v, let $\bullet v$ and $v \bullet$ respectively denote the pre-set and post-set of in usual way, i.e. the set of nodes that have arcs to and from v, respectively. A marking $m(p_i) = l$ is characterized by l black dots (tokens) inside a circle representing place p_i. An available resource or an ongoing job in DEDS is indicated by tokens in respective places. A transition $t \in T$ is enabled by marking $m(p)$ iff $\forall p \in \bullet t, m(p) \geq w(p,t)$, where $\bullet t$ is a set of input places to transition t, and $w(p,t)$ is weight of the arc between p, t. A transition t that meets the enabled condition is free to fire. When a transition t fires, all of its input places lose a number of tokens, and all of its output places gain a number of tokens. In a P/T Petri net with m places and n transitions, the incidence matrix $\mathbf{W}$ is a $n \times m$ matrix defined by:

$$\mathbf{W} = \mathbf{O} - \mathbf{I} \qquad (2)$$

The $\mathbf{I}$(input matrix) and $\mathbf{O}$(output matrix) are of size $n \times m$, and provide a complete description of the structure of a Petri net. If there are no self-loops, the structure may be described by $\mathbf{W}$ only. The incidence matrix allows an algebraic description of the evolution of the marking of a Petri net. The marking of Petri net changes from marking $\mathbf{m}_k$ to marking $\mathbf{m}_{k+1}$:

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \mathbf{W}^{T} \cdot \sigma \qquad (3)$$

where:

$\mathbf{W}$ –incidence matrix,

σ –transition vector.

The transition vector σ is composed of non-negative integers that correspond with the number of times a particular transition has been fired between markings $\mathbf{m}_k$ and $\mathbf{m}_{k+1}$.

Reachability set $\Re(\mathbf{m})$ shows the set of all possible markings reachable from $\mathbf{m}_0$. Transition $t \in T$ is said to be *dead* at $\mathbf{m}$ if there exists no $\mathbf{m}' \in \Re(\mathbf{m})$ that enables it, with $\Re(\mathbf{m})$ as the set of markings reachable from $\mathbf{m}$. Marking $\mathbf{m}$ is said to be dead if no $t \in T$ is enabled at $\mathbf{m}$. A place $p \in P$ is said to be dead or deadlocked at $\mathbf{m}$ if $m(p) = m'(p) = 0$ for all $m'(p) \in \Re(\mathbf{m})$.

Structural characteristics of Petri net are P and T *invariants*, *siphons* and *traps*. *P – invariant* corresponds to a set of places whose weighted token count remains a constant for all possible markings. If Petri net has only one *P*-invariant, one can obtain a *P*-invariant from non-negative integer place vector $\mathbf{P}_{(mx1)}$ that is a solution of:

$$\mathbf{W} \cdot \mathbf{P} = \mathbf{0} \tag{4}$$

Positive elements in $\mathbf{P}_{(mx1)}$ correspond to a set of places whose weighted token count remains a constant for all possible markings. *Siphon S* is a set of places such that every transition that has an output arc to one of these places also has an input arc from one of these places. For siphon S it is true that $\bullet S \subseteq S\bullet$. *Trap T* is a set of places such that every transition that has an input arc from one of these places also has an output arc to one of these places. For trap T it is true that $T\bullet \subseteq \bullet T$.Once the trap becomes marked, it will always be marked for all the future reachable markings. Once the siphon becomes empty, it will always remain empty. More about siphons and traps can be found in Murata [14].

The flowline system Petri net (FPN) is a subclass of P/T Petri net, which is especially designed for analysing MRF type of flexible manufacturing systems (FMS) [13]. Let Π be a set of distinct types of parts produced (or customers served) by FMS. Each part type $k \in \Pi$ is characterized by predetermined sequence of jobs $J^k = \{J_1^k, J_2^k, \ldots, J_{L_k}^k\}$, with at least one resource for each job (L_k is the number of jobs for a particular part type). Let R denote the set of system resources, with each $r \in R$ a pool of multiple copies of a given resource. In the FPN places P are divided as $P = R \cup J \cup J_{in} \cup J_{out}$ with R, J_{in}, J_{out} as the set of places respectively representing the availability of resources, units waiting for arrival and finished units, and J as the set of places representing the ongoing jobs. The set of transitions T can be partitioned as $T = \cup_{k \in \Pi} T^k$, where $T^k = \{t_1^k, t_2^k, \ldots, t_{L_k+1}^k\}$, with $t_i^k = \bullet J_i^k = J_{i\text{-}1}^k \bullet$, for $i \notin \{1, L_k\}$; while $t_1^k = \bullet J_1^k = J_{in}^k \bullet$ and $t_{L_k+1}^k = \bullet J_{out}^k = J_{L_k}^k \bullet$. Transition t is said to be job (resource) enabled if $m(\bullet t \cap J) > 0$ and $m(\bullet t \cap R) > 0$. For any $r \in R$, define the job set $J(r)$ as the set of jobs using r, and resource loop $L(r) = r \cup J(r)$. Given a set of resources $Q \subset R$, define the job set of Q as $J(Q) = \cup_{r \in Q} J(r)$. Denote by $R(J_i^k)$ resources used by job J_i^k.

The systems described in this paper belong to the class of MRF1PN. MRF1PN is a subclass of MRF systems described by FPN which satisfy: (i) $\forall p \in P$, $\bullet p \cap p \bullet = \varnothing$; (ii) $\forall k \in \Pi$, $t_1^k \bullet \cap P \setminus J = \varnothing$ and $\bullet t_{L_{k+1}}^k \cap P \setminus J = \varnothing$; (iii) $\forall J_i^k \in J$, $\left| R(J_i^k) \right| = 1$ and $R(J_i^k) \neq R(J_{i+1}^k)$; (iv) $\forall J_i^k \in J$, $\left| J_i^k \bullet \right| = 1$; (v) $\forall t_i^k \in J$, $\left| \bullet t_i^k \cap J \right| \leq 1$; (vi) $\forall r \in R$, $\left| J(r) \right| \geq 1$. This means that (i) there are no self loops, (ii) each unit-path has a well defined beginning and an end, (iii) every job requires one and only one resource with no two consequent jobs using the same resource, (iv) and (v), there are no choice jobs and no assembly jobs, (vi) there are shared resources. In MRF1 system (MRF1PN), for any $r \in R$, $J(r) = r \bullet\bullet \cap J = \bullet\bullet r \cap J$ and $R(J_i^k) = \bullet\bullet J_i^k \cap R = J_i^k \bullet\bullet \cap R$. For any two $r_i, r_j \in R$, r_i is said to wait r_j, denoted $r_i \rightarrow r_j$, if the availability or r_j is immediate requirement for the release or r_i, i.e., $\bullet r_i \cap r_j \bullet \neq \varnothing$, or equivalently, if there exists at least one transition $t \in \bullet r_i \cup r_j \bullet$.

Any set of resources is called *circular wait* (*CW*), if among the set of resources $r_a, r_b, \ldots, r_w$ there exist wait relations among them such that $r_a \rightarrow r_b \rightarrow \ldots \rightarrow r_w$ and $r_w \rightarrow r_a$. CW relations are characteristic among shared and non-shared resources in MRF1PN and contain at least one shared resource. Simple circular wait (SCW) is composed of different resources while cyclic circular wait (CCW) is composed of unions of non-disjoint SCWs. [3]

The deadlock in MRF1PN is connected with the system condition called *circular blocking* CB, which is a consequence of the existence of *circular wait relations* CW among resources in the system. A $C \in CW$ is said to be in CB if (i) $m(C) = 0$; and (ii) for each $r \in C$, $\forall p \in J(r)$ with $m(p) \neq 0$, $p \bullet \in C \bullet$. Avoiding CB is necessary but generally not sufficient for deadlock-free dispatching policy. To prevent deadlock in MRF1PN we must first avoid CB conditions, which are closely related to the critical siphon. A critical siphon S is a minimal siphon that does not contain any resource loops. The next step is to find sets of job places, the so-called *critical subsystems* $J_0(C)$, which are very important in deadlock prevention. A $C \in CW$ is in CB at any $m_0 \in \Re(\mathbf{m})$ if and only if critical siphon becomes empty ($m(S_C) = 0$). The critical siphon is empty if and only if $m(J_0(C)) = m_0(C)$; or equivalently, to avoid deadlock we must ensure that $m(S_C) \neq 0$ by applying constraint $m(J_0(C)) < m_0(C)$ to the set $\Re(\mathbf{m}_0)$. The token sum in the critical subsystem $J_0(C)$ must be limited above $m_0(C) - 1$. To avoid such first level deadlocks we must connect control places to the transitions before and after any critical subsystems $J_0(C)$ which make sure that the token sum in the critical subsystem $J_0(C)$ is limited to $m_0(C) - 1$.

The above system is the so-called *regular system*, for which it is true that avoiding CB is a necessary and sufficient condition for avoiding system deadlock. The above is not true for irregular systems. The irregular system contains *key resource*. It must be noted that when a system contains key resource, the system may run in the so-called *cyclic circular wait relation* CCW and also in the so-called *second level deadlock* several steps before any CB actually occurs. To avoid the second level deadlock we must find the key resource and apply control policy to make sure that the key resource does not remain the last available resource [13].

Figure 1 shows MRF1PN with one input and one output place J_{in}^1 and J_{out}^1. There is a set of job places $J^1 = \{J_1^1, J_2^1, J_3^1, J_4^1\}$, a set of resource places $R = \{r_1, r_2, r_3\}$, one simple circular wait (SCW) $C=\{r_1,r_2,r_3\}$ and one critical system $J_0=\{J_1^1,J_2^1,J_3^1\}$. To avoid deadlock, control places c_1 and c_2 maintain the number of tokens in the critical system to maximum 2 (less than initial marking in SCW).

To find the desired deadlock avoidance control policy, matrix-based description of MRF1PN is required. The structural properties of PN can be read from a system matrix. There are two sets of system matrices: $\mathbf{F}_\mathrm{u}$, $\mathbf{F}_\mathrm{y}$,$\mathbf{F}_\mathrm{v}$,$\mathbf{F}_\mathrm{r}$ and $\mathbf{S}_\mathrm{u}$, $\mathbf{S}_\mathrm{v}$, $\mathbf{S}_\mathrm{r}$, $\mathbf{S}_\mathrm{y}$. Matrices **F** capture conditions that must be full field before firing of transitions, while matrices **S** are responsible for actions after firing of transitions. A number of rows of $\mathbf{F}_\mathrm{u}$, $\mathbf{F}_\mathrm{v}$, $\mathbf{F}_\mathrm{r}$, $\mathbf{F}_\mathrm{y}$ define the number of transitions, while the number of columns defines the number of input places, jobs, resources and output places respectively. A number of columns of $\mathbf{S}_\mathrm{u}$, $\mathbf{S}_\mathrm{v}$, $\mathbf{S}_\mathrm{r}$, $\mathbf{S}_\mathrm{y}$ define the number of transitions, while the number of rows defines the number of input places, jobs, resources and output places respectively.

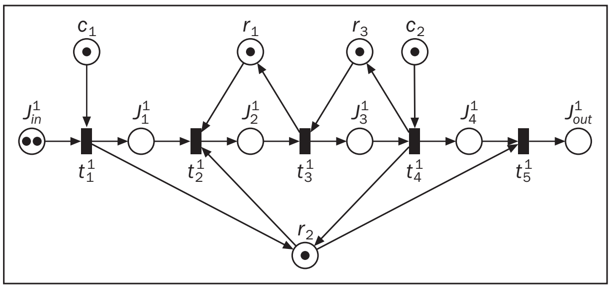

Figure 1 - MRF1PN type of FPN Petri net

Source: S Bogdan, F.L. Lewis, Z. Kovačić, J. Mireles: "Manufacturing systems control design"

Each entry $(f_r)_{i,j}$ in the resource-requirements matrix $\mathbf{F}_\mathrm{r}$ is associated with an arc connecting a place, representing resource availability, with the corresponding transition; each entry $(s_r)_{i,j}$ in the resource-release matrix $\mathbf{S}_\mathrm{r}$ expresses the connections between transitions and places that hold tokens when resources are idle. Correspondingly, each entry $(f_v)_{i,j}$ and $(s_v)_{i,j}$ in job-sequencing matrix $\mathbf{F}_\mathrm{v}$ and job-start matrix $\mathbf{S}_\mathrm{v}$ represent arcs connecting transitions and places associated operations executed by resources. The input matrix $\mathbf{F}_\mathrm{u}$ portrays output arcs from input places, while output matrix $\mathbf{S}_\mathrm{y}$ depicts input arcs to output places. Since we assume that input places are source places (places with no input transitions) and output places are sink places (places with no output transitions),v matrices $\mathbf{F}_\mathrm{y}$ and $\mathbf{S}_\mathrm{u}$ are null matrices, $\mathbf{F}_\mathrm{y}=\mathbf{S}_\mathrm{u}=[0]$. As a result, input and output incidence matrices **I** and **O** can be obtained from the system matrices:

$$\begin{aligned}\mathbf{I}&=[\mathbf{F}_\mathrm{u}\ \ \mathbf{F}_\mathrm{v}\ \ \mathbf{F}_\mathrm{r}\ \ \mathbf{F}_\mathrm{y}]=\mathbf{F}\\ \mathbf{O}&=\left[\mathbf{S}_\mathrm{u}^\mathrm{T}\ \ \mathbf{S}_\mathrm{v}^\mathrm{T}\ \ \mathbf{S}_\mathrm{r}^\mathrm{T}\ \ \mathbf{S}_\mathrm{y}^\mathrm{T}\right]=\mathbf{S}^\mathrm{T}\end{aligned}\qquad(5)$$

Matrices $\mathbf{F}_\mathrm{u}$, $\mathbf{F}_\mathrm{y}$,$\mathbf{F}_\mathrm{v}$,$\mathbf{F}_\mathrm{r}$ and $\mathbf{S}_\mathrm{u}$, $\mathbf{S}_\mathrm{v}$, $\mathbf{S}_\mathrm{r}$, $\mathbf{S}_\mathrm{y}$ are binary matrices.

3. MATRIX APPROACH OF SUPERVISOR DESIGN

Matrix approach of deadlock supervisor design begins with modelling of traffic system by using MRF1PN. Then, the structural properties of MRF1PN must be explored to find conflict and deadlock free control policy. To achieve this, system matrices $\mathbf{F}_\mathrm{u}$, $\mathbf{F}_\mathrm{y}$,$\mathbf{F}_\mathrm{v}$,$\mathbf{F}_\mathrm{r}$ and $\mathbf{S}_\mathrm{u}$, $\mathbf{S}_\mathrm{v}$, $\mathbf{S}_\mathrm{r}$, $\mathbf{S}_\mathrm{y}$ can be derived from MRF1PN. The procedure for finding all structural properties of MRF1PN and for deadlock free control policy can be divided in several steps: *Step 1* - Find all resource loops $L(r)$ via computing their covering binary **P**-invariants; *Step 2* - Find wait relation matrix $\mathbf{G}_\mathrm{W}$, all SCW and CCW; *Step 3* - Find critical siphons matrix $\mathbf{S}_{\mathrm{C_i}}$ and critical subsystem matrix $\mathbf{J}_0(\mathbf{C})$; *Step 4* - Ensure that the token count in each critical subsystem $J_0(C_i)$ is limited above $m_0(C_i)-1$.

The DEDS modelled by MRF1PN can be regular or irregular. For regular system the only condition for deadlock free policy is to control the token count in the critical subsystem (see Section II), and the first 4 steps are sufficient. If the system is irregular, then the second level deadlock can arise, and we must find key resources and proceed with Steps 5 and 6: *Step 5* - Find key resource if the system is irregular; *Step 6* - Ensure that the key resource does not remain the last available resource. Here are brief explanations of Steps 1-6:

Step 1 – Determination of the resource loops $L(r)$ via computing their covering binary **P**-invariants. The binary basis for **P**-invariants is given by the columns of matrix **P**:

$$P=\begin{bmatrix}-(\hat{\mathbf{S}}_\mathrm{v}^\mathrm{T}-\hat{\mathbf{F}}_\mathrm{v})^{-1}\cdot(\hat{\mathbf{S}}_\mathrm{r}^\mathrm{T}-\hat{\mathbf{F}}_\mathrm{r})\\ \mathbf{I}_{\mathrm{r}\times\mathrm{r}}\end{bmatrix}\qquad(6)$$

where:

$\mathbf{I}_{\mathrm{r}\times\mathrm{r}}$ – identity matrix with *r* resources in the system.

Matrices $\hat{\mathbf{F}}_\mathrm{v}$ and $\hat{\mathbf{F}}_\mathrm{r}$ are formed by deleting rows that correspond to the terminal transitions. Matrices $\hat{\mathbf{S}}_\mathrm{v}$and $\hat{\mathbf{S}}_\mathrm{r}$ are formed by deleting columns that correspond to the terminal transitions. Terminal transitions are transitions which have arcs to J_{out}.

Step 2 - Find all simple resource circuits and all the CWs which the former constitute. To achieve this, the wait relation matrix between all resources in MRF1PN must be determined. All the wait relations are captured by the wait relation matrix:

$$\mathbf{G}_\mathrm{w}=\mathbf{S}_\mathrm{r}\otimes\mathbf{F}_\mathrm{r}\qquad(7)$$

Where the matrix operation ⊗ is defined in and/or algebra, i.e. standard addition and multiplication of matrices elements are replaced by the logical "and" and "or", respectively.

Having obtained matrix $\mathbf{G}_\mathrm{w}$, there are standard efficient techniques of polynomial complexity, such as string algebra [15], for identifying matrices **C** and $\boldsymbol{\psi}$.

From $\mathbf{C}_{r\times(scw+ccw)}$ it is possible to determine all $\mathbf{C}_\mathrm{i}$, and from $\psi_{scw\times(scw+ccw)}$ it is possible to detect which SCWs are involved in particular CCW. Columns of matrix **C** which contain non-shared resources denote vectors c_j, and columns of matrix **C** which contain shared resources denote vectors $\mathbf{c}_\mathrm{sj}$.

Step 3 - Find critical siphons matrix $\mathbf{S}_{\mathrm{C_i}}$ and critical subsystem matrix $\mathbf{J}_0(\mathbf{C})$ using these equations:

$$\mathbf{S}_{\mathrm{C_i}} = \begin{bmatrix} \mathbf{c}_\mathrm{sj}^\mathrm{T} \otimes \mathbf{S}_\mathrm{r} \otimes \mathbf{F}_\mathrm{v} \wedge \mathbf{c}_\mathrm{i}^\mathrm{T} \otimes \mathbf{F}_\mathrm{r}^\mathrm{T} \otimes \mathbf{F}_\mathrm{v} \\ \mathbf{c}_\mathrm{i} \end{bmatrix} \tag{8}$$

$$\mathbf{J}_0(\mathbf{C}) = \mathbf{P} \otimes \mathbf{C} \wedge \overline{\mathbf{S}_\mathrm{C}} \tag{9}$$

Where matrix operation $\wedge$ denotes element-by-element matrix logical “and” operation.

Columns of matrix $\mathbf{S}_{\mathrm{C_i}}$ are critical siphons, and column of matrix $\mathbf{J}_0(\mathbf{C})$ is a critical subsystem.

Step 4 - Ensure that the token count in each critical subsystem $J_0(C_i)$ is limited above $m_0(C_i) - 1$. This can be achieved by adding control places (supervisor) to MRF1PN. By adding or taking away tokens in control places, one can control the number of tokens in critical subsystems.

Step 5 - Before a particular control policy is applied, one has to check if MRF1PN is irregular. Key resources can be identified by analyzing interconnections of CWs and their siphons. To confirm the existence of key resources, we must determine the presence of CCW loops. These structures specify a particular sharing among circular waits, and are requisite for the existence of key resources.

The matrix test to find CCW among all CWs in the system is to find $\mathbf{C}_\mathrm{CW}$:

$$\mathbf{C}_{\mathrm{CW}_{(\mathrm{CW}\times\mathrm{CW})}} = (\mathbf{T}_{\mathrm{S_C}}^{-\mathrm{T}} \otimes \mathbf{T}_{\mathrm{S_C}}^{+})^\mathrm{T} \wedge (\mathbf{T}_{\mathrm{S_C}}^{-\mathrm{T}} \otimes \mathbf{T}_{\mathrm{S_C}}^{+}) \tag{10}$$

where:

$\mathbf{T}_{\mathrm{S_C}}^{-\mathrm{T}} = (\mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{S}_\mathrm{v} - \mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{S}_\mathrm{v} \wedge \mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{F}_\mathrm{v}^\mathrm{T})$ – matrix which determines the set of transitions which decrease token counts in every critical siphon,

$\mathbf{T}_{\mathrm{S_C}}^{+\mathrm{T}} = (\mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{F}_\mathrm{v}^\mathrm{T} - \mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{S}_\mathrm{v} \wedge \mathbf{v}_\mathrm{oc}^\mathrm{T} \otimes \mathbf{F}_\mathrm{v}^\mathrm{T})$ – matrix which determines the set of transitions which increase token counts in every critical siphon,

$\mathbf{v}_\mathrm{OC}$ – critical subsystems job set matrix of type $(CW \times J)$. The critical subsystem job set matrix denotes job places which belong to CWs.[3]

When $\mathbf{C}_\mathrm{CW} = [0]$ the system is regular, otherwise element $\mathbf{C}_\mathrm{CW}(i,j) = 1$ indicates that $\mathbf{C}_\mathrm{i}$ and $\mathbf{C}_\mathrm{j}$ form a CCW. Obviously $\mathbf{C}_\mathrm{CW}$ is symmetric matrix. To identify the key resource we must apply the following straightforward matrix formula:

$$\mathbf{R}_{\mathrm{CCW}_{(cw\times r)}} = (\mathbf{F}_\mathrm{r}^\mathrm{T} \otimes \mathbf{T}_\mathrm{CCW}^{+}) \wedge (\mathbf{F}_\mathrm{r}^\mathrm{T} \otimes \mathbf{T}_\mathrm{CCW}^{-}) \tag{11}$$

where:

$\mathbf{T}_\mathrm{CCW}^{-} = (\mathbf{T}_{\mathrm{S_C}}^{+} \otimes \mathbf{C}_\mathrm{CW}) \wedge \mathbf{T}_{\mathrm{S_C}}^{-}$ – matrix which determines the set of transitions which decrease token counts in CCWs

$\mathbf{T}_\mathrm{CCW}^{+} = (\mathbf{T}_{\mathrm{S_C}}^{-} \otimes \mathbf{C}_\mathrm{CW}) \wedge \mathbf{T}_{\mathrm{S_C}}^{+}$ – matrix which determines the set of transitions which increase token counts in CCWs

Matrix $\mathbf{R}_\mathrm{CCW}$ provides for each CW the corresponding vector of key resources shared with other CWs in one or more CCW. If this matrix is zero, there are no key resources in the system [3].

4. CASE STUDY – RIVER TRAFFIC SYSTEM

This chapter deals with a supervisor design for the river traffic system (*Figure 2*). The presented case study is relatively simple. The intention of this example is to clarify the theory in the previous section. However, the above theory is applicable to more complex systems.

Suppose that a river traffic system consists of three canals K_1, K_2, K_3 and four basins B_{1A}, B_{2A}, B_{1B} and B_{2B}. All canals and basins represent resources of the river system. The vessels at the left end of the river system wait for the passage to the right in *Direction A*, and vessels on the right side wait for the passage to the left in *Direction B*.

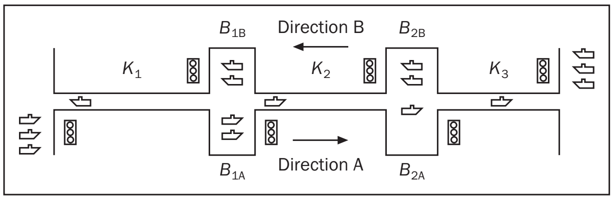

Figure 2 - River traffic system

The vessels can move through the canals using their own propulsion plant. The vessel in direction A must pass $K_1 \to B_{1A} \to K_2 \to B_{2A} \to K_3$, and the vessel in direction B must pass $K_3 \to B_{2B} \to K_2 \to B_{1B} \to K_1$. The vessels in both directions share the canals. The basins are designed only for one direction and for waiting for the availability of the next canal. If a particular resource is occupied at a moment of time, and if there are vessels waiting to use them, then these vessels wait for the availability of the occupied resource at the exit of the resource where they are at the moment of time. When the resource becomes available, it is occupied by the awaiting vessels. The moving of the vessels in the marine canal traffic system in *Figure 2* is limited due to the capacity of resources. The canal capacities are $K_1, K_2, K_3 = 1$, the basin capacities are $B_{1A} = B_{2A} = B_{1B} = B_{2B} = 2$ vessels.

The traffic in the river system can be fully controlled using RTMS which controls traffic lights at the entrance into resource in directions A and B. RTMS detects vessels passing from one resource to another and the number of vessels at a particular resource. The traffic light signalization system should not let vessels enter a resource in order to avoid forbidden states. The supervisor, which is implemented in RTMS software is re-

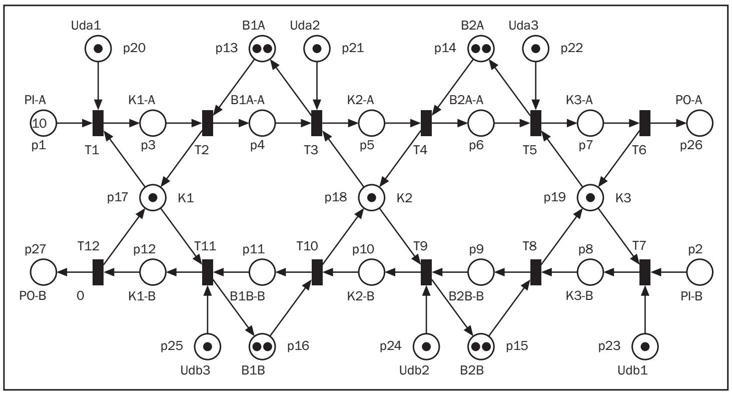

Figure 3 - Petri net model of river traffic system with control places p_{20},...,p_{25}

quired to be maximally permissible i.e. not hinder the passage of vessels. To design a supervisor, the first step is to make an MRF1PN model of the river system. *Figure 3* shows Petri net model of the river system in *Figure 2* with control places. *Figure 3* shows the model of empty river traffic system (without vessels in canals and basins), with 10 vessels waiting for direction A, 10 vessels waiting for direction B. The capacities of canals and basins are the same as in *Figure 2*.

The tokens in input places $\{p_1, p_2\}$ will represent the vessels waiting for entering in the system. The tokens in output places $\{p_{26}, p_{27}\}$ will represent the vessels leaving the system. The set of all places that represent jobs in the system is $\{p_3, \ldots, p_{12}\}$ (the number of tokens in a job place will represent the number of vessels in particular resource), and the number of tokens in resource places $\{p_{13}, \ldots, p_{19}\}$ will represent how many resources are available in CWS. Control places $\{p_{20}, \ldots, p_{25}\}$ belong to the supervisor which ensures the conflict and deadlock free operation by disabling or enabling particular transitions. All places are marked by labels with a few characters (*Figure 3*) for easier understandability.

Conflicts arise when vessels from both directions try to occupy the same shared resources $\{p_{17}, p_{18}, p_{19}\}$. In this situation the transitions t_1, t_{11} and t_3, t_9 and t_5, t_7 are in conflict (both transitions are enabled at the same time). A conflict-free supervisor enables only one direction (direction A or direction B) with equal probability. The second problem is how to design a deadlock-free supervisor. To achieve this we must apply the matrix approach described in Section II. Here are the results:

Step 1: **P**-invariants can be calculated applying (6). There are 7 **P**-invariants in the net: $P_1 = \{p_4, p_{13}\}$, $P_2 = \{p_6, p_{14}\}$, $P_3 = \{p_9, p_{15}\}$, $P_4 = \{p_{11}, p_{16}\}$, $P_5 = \{p_3, p_{12}, p_{17}\}$, $P_6 = \{p_5, p_{10}, p_{18}\}$, $P_7 = \{p_7, p_8, p_{19}\}$.

Step 2: Applying (7) and string algebra [12] we can find 2 SCWs $C_1 = \{p_{13}, p_{16}, p_{17}, p_{18}\}$, $C_2 = \{p_{14}, p_{15}, p_{18}, p_{19}\}$ and 1 CCW $C_3 = C_1 + C_2 = \{p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$.

Step 3: By applying (8) and (9) it is possible to find all critical siphons and critical subsystems. There are 3 critical siphons:

$$S_{C_1} = \{p_5, p_{12}, p_{13}, p_{16}, p_{17}, p_{18}\}\ ,$$
$$S_{C_2} = \{p_7, p_{10}, p_{14}, p_{15}, p_{18}, p_{19}\},$$
$$S_{C_3} = \{p_7, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\},$$

and 3 critical subsystems

$$J_{0_1} = \{p_3, p_4, p_{10}, p_{11}\},$$
$$J_{0_2} = \{p_5, p_6, p_8, p_9\},$$
$$J_{0_3} = \{p_3, p_4, p_5, p_6, p_8, p_9, p_{10}, p_{11}\}.$$

Step 4: The initial marking of C_1, C_2 and C_3 are $m_0(C_1) = 6$, $m_0(C_2) = 6$ and $m_0(C_3) = 11$.To avoid the first level deadlock we must limit the number of vessels in critical subsystems $m(J_{0_1}) \leq 5$, $m(J_{0_2}) \leq 5$ and $m(J_{0_3}) \leq 10$.

Step 5: To check the regularity of the system, matrix $\mathbf{C}_{CW}$ must be calculated by applying (10). From $\mathbf{C}_{CW}$ it is possible to see that the system is irregular and that SCWs C_1, C_2 and C_3. From $\mathbf{R}_{CCW}$ applying (11) it is possible to see that the key resource in the system is p_{18} (canal K2) .

Step 6: *To ensure the absence of the second level deadlock the supervisor has to take care of the availability of key resource p_{18} in such a way that canal K2 does not remain the last available resource in the system.*

The deadlock prevention supervisor which applies the control policy derived in Steps 4 and 6 is verified using computer simulation of vessels moving through the river system. We are simulating the process of moving 10 vessels in direction A, and 10 vessels in direction B at the same time. Let us assume that every vessel must remain in the resource K1_A = 1 h, B1A_A = 0.5 h, K2_A = 2 h, B2A_A = 0.5 h, K3_A = 1 h, K3_B = 1 h, B2B_B = 0.5 h, K2_B = 2 h, B1B_B = 0.5 h K1_B=1 h. In case of conflict, we do not give priority to any direction. The simulation starts from the initial state of Petri net displayed in *Figure 3*. The upper 10 diagrams in *Figure 4* show the number of vessels in job places $p_3 - p_{12}$, and lower 6 diagrams (uda1-udb3) show markings of control places $p_{20} - p_{25}$. From *Figure 4* it is possible to see that the number of tokens in all places can be 0 (low level) or 1 (high level), except B1A-A and B2B-B (maximum 2 tokens, which corresponds to one step above high level). There are no deadlocks and conflicts in the system. Control places are "traffic lights" which disable firing of transitions (logical "1" means green light – token in the control place, logical "0" means red light – no tokens in control place). *Figure 5* shows markings of input places in direction A and B (PI-A, PI-B) and output places in direction A and B (PO-A, PO-A). From *Figure 5* it is possible to see that the last vessel leaves the canal system 43 hours after the process of vessels moving through the canal system has begun. The applied control policy ensures maximally permissive behaviour of the supervisor with no conflicts and no deadlocks in the system.

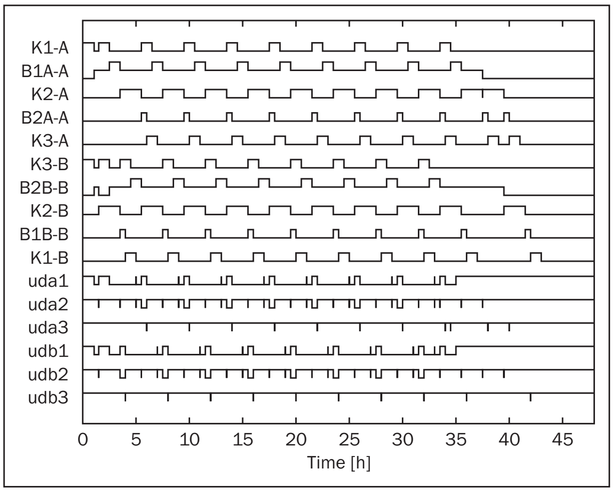

Figure 4 - Number of vessels in job places and tokens in control places

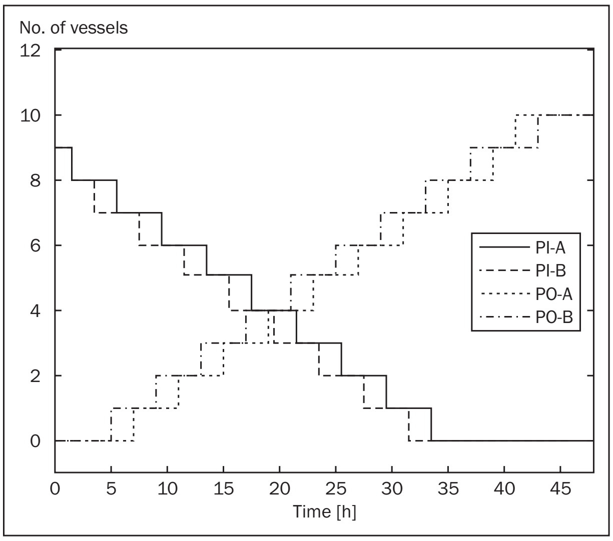

Figure 5 - Number of vessels in input and output places

5. CONCLUSION

The paper shows a straightforward matrix-based method for calculating the maximally permissible conflict and deadlock prevention control policy, which can be easily implemented by people or by traffic lights in the river traffic system. To achieve this, the first step is to make a suitable Petri net model of river traffic system using MRF_1 type of flowline Petri net. Then, the structural properties of the net like P-invariants, circular waits, critical siphons and critical subsystems are investigated. To avoid conflicts, the first and second level deadlocks, the authors propose adding of control places (supervisor). To avoid conflicts and the first level deadlock, the control places disable firing of particular transitions and limit the number of vessels in critical subsystems. But still the second level deadlock can exist if the system is irregular and if it contains the so-called key resource. To avoid the second level deadlock, the supervisor must take care that the key resource is not the last available resource in the net. The calculated controller is verified using a P-timed Petri net, and computer simulation of dense traffic by using MATLAB environment. The proposed matrix-based method of supervisor design is not time-consuming, and it is suitable for complex traffic systems. Future research will be focused on deadlock avoidance problem of complex systems with more key resources.

DANKO KEZIĆ, Ph.D.
E-mail: danko.kezic@pfst.hr
ANITA GUDELJ, M.Sc.
E-mail: anita.gudelj@pfst.hr
University of Split, Faculty of Maritime Studies
Zrinsko-Frankopanska 38, HR-21000 Split,
Republic of Croatia

SAŽETAK

PROJEKTIRANJE NADZORNIKA ZA IZBJEGAVANJE ZAGLAVLJENJA RIJEČNOG SUSTAVA KORIŠTENJEM PETRIJEVIH MREŽA

Napredna funkcija računalnog sustava za upravljanje riječnim prometom treba automatski previdjeti i spriječiti moguća stanja konflikta i zastoja među brodovima na način da koristi primjerenu strategiju upravljanja (nadzornik). U ovom članku se predlaže formalna metoda za proračun najviše dozvoljavajućeg nadzornika za sprječavanje zaglavljenja. Za modeliranje riječnog sustava autori koriste klasu Petrijevih mreža pogodnu za opis višeulaznih proizvodnih linija sa razdvojenim skupovima resursa, poslova, i kontrolnih mjesta, te matričnu metodu za analizu sustava. Korištenjem matrične algebre, autori analiziraju strukturne karakteristike Petrijeve mreže (kružna čekanja, P-invarijante, kritične sifone i podsustave, ključni resurs) i predlažu korake za projektiranje nadzornika. Zaglavljenja prve i druge razine se mogu izbjeći održavanjem broja oznaka u kritičnim podsustavima i osiguravanjem da ključni resurs ne ostane zadnji slobodni resurs u sustavu. Dobiveni nadzornik je verificiran uz pomoć računalne simulacije koristeći program MATLAB.

KLJUČNE RIJEČI

sustav za nadzor prometa, izbjegavanje zaglavljenja, sustav diskretnih događaja, Petrijeva mreža

LITERATURE

[1] **Barkaoui, K.**, **Abdallah J.**: "*Deadlock avoidance in FMS based on structural theory of Petri Nets*", IEEE Symposium on Emerging Technologies and Factory Automation, 1995, pp 499-510.

[2] **Ezpelta, J.**, **Colom, J.**, **Martinez, J.**: "*A Petri net based deadlock prevention policy for flexible manufacturing systems*", IEEE Trans. on Robotics and Automation, Vol. 11, No. 2, April 1995, pp. 173-184.

[3] **Lautenbach, L.**, **Ridder, H.**: "*The linear algebra of deadlock avoidance – a Petri net approach*", Research Report of Institute for Computer Science, University of Koblenz, Germany, 1996.

[4] **Lewis, F.**, **Huang, H.**, **Tacconi, D.**, **Gurel, A.**, **Pastravanu, O.**: "*Analysis of deadlocks and circular waits using a matrix model for discrete event systems*", Automatica, Vol. 34, No. 9, 1998, pp. 0-19.

[5] **Kezić, D.**, **Perić, N.**, **Petrović, I.**: "*An algorithm for deadlock prevention based on iterative siphon control of Petri net*" Journal Automatica. Vol. 47, No. 1-2, Zagreb 2006, pp. 19–30.

[6] **Iordache, M.**, **Moody, J.**, **Antsaklis P.**: "*Automated synthesis of deadlock prevention supervisor using Petri nets*", Technical report of the ISIS Group at the University of Notre Dame, ISIS-2000-003, May 2000.

[7] **Huang, Y.**, **Jeng., M.**, **Xie, X.**, **Chung, S.**: "*Deadlock prevention policy based on Petri nets and siphons*", International Journal of Production Research, Vol 39, No. 2, January 2001, pp. 283-305(23).

[8] **Barkaoui, K.**, **Chaoui, A.**, **Zuoari, B.**: "*Supervisory control of discrete event systems based on structure theory of Petri nets*", in Proc. IEEE Int. Conf. System, Man, Cybernetic, Orlando, FL, 1997, pp. 3750-3755.

[9] **Barkaoui, K. Petrucci, L.**: "*Structral analysis of workflow nets with shared resources*", in Proc. Workshop Workflow Management: Net – Based Concepts, Models, Tech, Tools, Lisabon, Portugal, June 22, 1988, pp. 82-85.

[10] **Tricas, F.**, **Garcia-Valles, F.**, **Colom, J. M.**, **Ezpelta J.**: "*An iterative method for deadlock prevention in FMS*", in Proc. 5th Workshop Discrete Event System, R. Boel and G.Stremersch, Eds. Ghent, Belgium, Aug 21-23, 2000, pp. 139 – 148.

[11] **Tricas, F.**, **Garcia-Valles, F.**, **Colom, J. M.**, **Ezpelta J.**: "*A Petri net structure based deadlock prevention solution for sequential resource allocation systems*", in Proc. IEEE Int. Conf. Robotic and Automation, Barcelona, Spain, Apr, 18-22, 2005, pp. 271-277.

[12] **Bogdan, S.**, **Lewis, F.L**: "*Matrix approach to deadlock avoidance of dispatching in multi-class finite buffer reentrant flow lines*" Intelligent Control, 1997. Proceedings of the 1997 IEEE International Symposium , Issue 16-18 July 1997 pp. 397 – 402.

[13] **Bogdan, S.**, **Lewis, F.L.**, **Kovačić, Z.**, **Mireles, J.**: "*Manufacturing systems control design – A matrix-based approach*", Springer, 2006.

[14] **Murata, T.**: "*Petri Nets: Properties, Analysis and Applications*", in Proc. of the IEEE, 1989, 77(4), pp. 541-580.