ON THE INVARIANTS OF COLOURED PETRI NETS

Y. Narahari and N. Viswanadham
School of Automation
Indian Institute of Science
Bangalore 560012
INDIA

ABSTRACT

In this paper, we develop a theorem that enables computation of the place invariants of the union of a finite collection of coloured Petri Nets when the individual nets satisfy certain conditions and their invariants are known. We consider the illustrative examples of the Readers-Writers problem, a resource sharing system, and a network of databases and show how this theorem is a valuable tool in the analysis of concurrent systems.

1. INTRODUCTION

Petri Nets, also known as Place-Transition Nets (PTNs) have been extensively used in the modelling and analysis of concurrent systems [9]. This modelling tool has become very popular on account of its graphical elegance and mathematical rigour. Of late, many extensions to the basic Petri Net model have been proposed in order to enable a higher level of description of systems. Predicate-Transition Nets (Pr-T Nets) [2,3] and Coloured Petri Nets (CPNs) [5] are two important developments in this direction.

Much of the Petri Net based analysis is accomplished using the place-invariants or p-invariants (referred to henceforth as 'invariants') [4,7] of the Petri Net model. This is true of Pr-T nets and CPNs also [2,3,5,6]. Invariants of nets are very useful in investigating important properties such as absence of overflows, presence of deadlocks, and existence of mutual exclusion between events [2,3,5,6,8]. Jensen, in [5], has proposed a very elegant formalism, based on linear, integer valued functions, for handling the invariants of CPNs. He has also developed, in [6], a set of transformation rules on the incidence matrix of a CPN to find its invariants. Our work in this paper also looks at the invariants of a CPN. We develop a theorem that enables computation of all invariants of the union of a finite number of CPNs with no common transitions when all invariants of the individual CPNs are known and conversely. This theorem automatically holds for PTNs also.

After developing the theorem, we discuss three examples of concurrent systems to show the value of the theorem in the analysis of concurrent systems. The *first* example is the version of the classical Readers-Writers problem presented in [5]. In this example, we write PTN models of the 'reader' process and the 'writer' process and show that the union of these two PTNs will give the PTN of the overall system. We invoke our theorem to compute all invariants of the PTN of the overall system. In the second example, we look at a resource sharing system comprising a set of processes and a set of resources where each process involves two stages of processing. A CPN model of the above system is obtained as the union of two CPNs representing the two stages of processing of each process. The theorem is again invoked for determining the invariants of the averall CPN model. The third and the final example considers the network of databases system discussed in [2,5]. For this example also, we show how our theorem is naturally applicable. The above three examples demonstrate the great value of the theorem in the analysis of concurrent systems.

The rest of the paper is organized as follows. In Section 2, we present formal definitions of all the basic terminology that will be required in subsequent sections of the paper. These definitions cover both PTNs and CPNs and are more or less the same as in [4,5,7] except for some changes in notation. In Section 3, we first state and prove the main result of this paper. We then state two of the immediate consequences of this result. The motivation for deriving the above result has come from [1]. Finally, in Section 4, we discuss, in detail, the use of the result for the three illustrative concurrent systems listed in the previous paragraph.

2. DEFINITIONS

In this section, we introduce notation and present certain relevant definitions. These definitions follow closely those given in [4,5,7]. However, we find it necessary to present these definitions since we have found it convenient to alter the notation at certain places. In sequel, we use the symbols N and Z to denote, respectively, the set of all non-negative integers and the set of all integers.

Definition 2.1 : Place-Transition Net (PTN). A place-transition Net is a 4-tuple (P,T,IN,OUT) where

P is a set of places,

T is a set of transitions,

$P \cup T \neq \phi$, $P \cap T = \phi$,

IN : $(P \times T) \rightarrow N$ is an input function, and

OUT : $(P \times T) \rightarrow N$ is an output function.

Note 2.1 : In [4], the definition of a PTN includes two other objects namely K, the maximum capacity of the places and M_0, the initial marking of the PTN. We assume K is infinity and so omit it in the definition. Also, since the paper looks only at the invariants of nets, we do not require any information about the initial marking of the net. Hence we omit M_0 also.

Note 2.2 : Henceforth, we restrict our attention to PTNs with finite number of places and finite number of transitions. Unless otherwise specified, P and T will be the following sets:

$$P = \{P_1, P_2, \dots, P_n\}, \; n > 0$$
$$T = \{t_1, t_2, \dots, t_m\}, \; m > 0$$

Definition 2.2 : Incidence Matrix. The incidence matrix $W = (w_{ij})$ of a PTN (P,T,IN,OUT) is an (n x m) matrix of integers given by

$$w_{ij} = OUT(p_i, t_j) - IN(p_i, t_j) \;\; \forall \; i = 1,2, \dots, n \text{ and } j = 1,2, \dots, m$$

Definition 2.3 : Place Invariant. Given a PTN (P,T,IN,OUT), a (1 x n) vector of integers, $X = (e_1, e_2, \dots, e_n)$, is said to be a place invariant if and only if

$$\sum_{i=1}^{n} e_i \, w_{ij} = 0 \;\; \forall \; j = 1,2, \dots, m$$

Definition 2.4 : Union of Place-Transition Nets. Let $G_1 = (P_1, T_1, IN_1, OUT_1)$ and $G_2 = (P_2, T_2, IN_2, OUT_2)$ be two PTNs such that

$$\left.\begin{array}{l} IN_1(p,t) \neq 0 \Rightarrow IN_2(p,t) = 0, \text{ and} \\ OUT_1(p,t) \neq 0 \Rightarrow OUT_2(p,t) = 0 \end{array}\right\} \;\forall \;\; p \in P_1 \cap P_2 \text{ and } t \in T_1 \cap T_2$$

We then define the union G of G_1 and G_2 as the PTN (P , T , IN , OUT) where

$$\begin{array}{lll} P & = P_1 & \cup \; P_2 \\ T & = T_1 & \cup \; T_2 \\ IN & = IN_1 & \cup \; IN_2 \\ OUT & = OUT_1 & \cup \; OUT_2 \end{array}$$

The union of more than two PTNs is defined likewise.

Note 2.3 : In the above definition, the set notation has been used for the functions IN, OUT, IN_1, OUT_1, IN_2, and OUT_2.

Definition 2.5 : Coloured Petri Net (CPN). A CPN is a 5-tuple (P, T, C, IN, OUT) where

P is a set of places,
T is a set of transitions,
C is a colour function such that $C : P \cup T \rightarrow$ Non-empty sets of colours,

IN and OUT are functions with domain (P x T) such that for all $(p,t) \in P \times T$,

$$IN(p,t)\ ,\ OUT(p,t) : C(p) \rightarrow [C(t) \rightarrow N]_f$$

where $[C(t) \rightarrow N]_f$ denotes the set of all total functions g from C(t) to N with the support $\{ a \in C(t) : g(a) \neq 0 \}$ finite.

Note 2.4 : As in case of PTNs, we consider in this paper, only unmarked CPNs, i.e., CPNs without any initial marking.

Note 2.5 : A PTN is a special case of a CPN where all the sets of colours have only one element [5].

Note 2.6 : The above definition of a CPN allows infinite number of places, infinite number of transitions and infinite colour sets. In this paper, we restrict our attention to CPNs with finite number of places, finite number of transitions, and finite colour sets. Unless otherwise specified, we assume the following:

$P = \{ p_1, p_2, \ldots, p_n \}$, $n > 0$
$T = \{ t_1, t_2, \ldots, t_m \}$, $m > 0$
$u_i = |C(p_i)|$, the cardinality of the colour set of p_i, $i = 1,2, \ldots, n$
$v_j = |C(t_j)|$, the cardinality of the colour set of t_j, $j = 1,2, \ldots, m$
$C(p_i) = \{ a_{i1} , a_{i2} , \ldots , a_{iu_i} \}$, $i = 1,2, \ldots, n$, and
$C(t_j) = \{ b_{j1} , b_{j2} , \ldots , b_{jv_j} \}$, $j = 1,2, \ldots, m$.

Definition 2.6 : Incidence Matrix of a CPN. Given a CPN (P, T, C, IN, OUT), its incidence matrix is an (n x m) matrix $W = (w_{ij})$ of submatrices. Submatrix w_{ij} is of dimension $u_i \times v_j$ and is defined by

$$(w_{ij})_{k\ell} = OUT(p_i, t_j)(a_{ik})(b_{j\ell}) - IN(p_i, t_j)(a_{ik})(b_{j\ell})$$

where, $i = 1,2, \ldots, n$; $j = 1,2, \ldots, m$; $k = 1,2, \ldots, u_i$ and $\ell = 1,2, \ldots, v_j$

Definition 2.7 : Place Invariant of a CPN. Let W be the incidence matrix of a CPN (P,T,C, IN,OUT) and let Q be a non-empty finite set with cardinality q. A block matrix

$$X = (e_1 , e_2 , \ldots , e_n)$$

where e_i , for $i = 1,2, \ldots, n$, is a matrix of integers of dimension $q \times u_i$ is said to be a place invariant (referred to henceforth as an invariant) iff

$$\sum_{i=1}^{n} e_i * w_{ij} = \mathbf{0} \quad \forall \quad j = 1,2, \ldots, m,$$

where the '+' of $\sum$ refers to matrix addition, '*' denotes matrix multiplication, and $\mathbf{0}$ is the zero matrix of dimension q x v_j.

Definition 2.8 : Union of CPNs. Let $G_1 = (P_1, T_1, C_1, IN_1, OUT_1)$ and $G_2 = (P_2, T_2, C_2, IN_2, OUT_2)$ be CPNs such that

(1) $C_1(p) = C_2(p) \ \forall \ p \in P_1 \cap P_2$

(2) $C_1(t) = C_2(t) \ \forall \ t \in T_1 \cap T_2$

(3) $IN_1 (p,t) (c') (c'') \neq 0 \Rightarrow IN_2 (p,t) (c') (c'') = 0$, and

$OUT_1 (p,t) (c') (c'') \neq 0 \Rightarrow OUT_2 (p,t) (c') (c'') = 0$

$\forall \ p \in P_1 \cap P_2$, $t \in T_1 \cap T_2$, $c' \in C (p)$ and $c'' \in C (t)$.

Then, the union G of G_1 and G_2 is the CPN (P,T,C,IN,OUT) where

$P = P_1 \cup P_2$,
$T = T_1 \cup T_2$,
$IN = IN_1 \cup IN_2$,
$OUT = OUT_1 \cup OUT_2$, and
$C(x) = C_1 (x)$ if $x \in P_1 \cup T_1$
$= C_2 (x)$ if $x \in P_2 \cup T_2$.

The union of more than two CPNs satisfying conditions (1)-(3) is defined likewise.

3. A THEOREM ON INVARIANTS

In this section, we state and prove the main result of this paper. We also state two immediate consequences of the above result. The first consequence gives the version of the result for PTNs. The version for PTNs has been independently proved in [8]. The second consequence , which is stated informally, enables extension of the result to any finite number (≥ 2) of CPNs.

Theorem 3.1 : Suppose $G_1 = (P_1, T_1, C_1, IN_1, OUT_1)$ and $G_2 = (P_2, T_2, C_2, IN_2, OUT_2)$ are two CPNs such that

(1) $P_1 = \{p_1, p_2, \ldots, p_r, p_{r+1}, \ldots, p_s\}$, $0 < r \leq s$

(2) $P_2 = \{p_r, p_{r+1}, \ldots, p_s, p_{s+1}, \ldots, p_n\}$, $s < n$

(3) $T_1 = \{t_1, t_2, \ldots, t_k\}$, $0 < k$

(4) $T_2 = \{t_{k+1}, t_{k+2}, \ldots, t_m\}$, $k < m$, and

(5) $C_1(p) = C_2(p) \ \forall \ p \in P_1 \cap P_2$.

Let us also assume that

(6) $G = (P, T, C, IN, OUT)$ is the union of G_1 and G_2,

(7) Q is a non-empty finite set with cardinality q, and

(8) $e_1, e_2, \dots, e_n$ are matrices of integers of dimensions $q \times u_1$, $q \times u_2$, $\dots$, $q \times u_n$ respectively (u_i, it may be recalled, is the cardinality of $C(p_i)$ for $i = 1, 2, \dots, n$).

Then, $X = (e_1, e_2, \dots, e_r, e_{r+1}, \dots, e_s, e_{s+1}, \dots, e_n)$ is an invariant of G iff

$$X_1 = (e_1, e_2, \dots, e_r, e_{r+1}, \dots, e_s)$$

is an invariant of G_1 and

$$X_2 = (e_r, e_{r+1}, \dots, e_s, e_{s+1}, \dots, e_n)$$

is an invariant of G_2.

Proof :

It is given that G is the union of G_1 and G_2. Therefore, we get

$$\left.\begin{aligned} P &= \{p_1, p_2, \dots, p_r, p_{r+1}, \dots, p_s, p_{s+1}, \dots, p_n\}, \\ T &= \{t_1, t_2, \dots, t_k, t_{k+1}, \dots, t_m\}, \\ IN &= IN_1 \cup IN_2, \\ OUT &= OUT_1 \cup OUT_2, \text{ and} \\ C(x) &= C_1(x) \ \forall \ x \in P_1 \cup T_1 \\ &= C_2(x) \ \forall \ x \in P_2 \cup T_2 \end{aligned}\right\} \quad \cdots (3.1)$$

Let

$$\begin{aligned} I_1 &= \{1, 2, \dots, r, r+1, \dots, s\} \\ I_2 &= \{r, r+1, \dots, s, s+1, \dots, n\}, \\ J_1 &= \{1, 2, \dots, k\}, \\ J_2 &= \{k+1, k+2, \dots, m\}, \\ I &= \{1, 2, \dots, n\}, \text{ and} \\ J &= \{1, 2, \dots, m\}. \end{aligned}$$

Let $W_1 = ((w_1)_{i_1 j_1})$, $W_2 = ((w_2)_{i_2 j_2})$, and $W = (w_{ij})$ be the incidence matrices of G_1, G_2, and G respectively. It may be noted that $i_1 \in I_1$, $j_1 \in J_1$, $i_2 \in I_2$, $j_2 \in J_2$, $i \in I$, and $j \in J$.

Then, we have, from (3.1),

$$
\begin{aligned}
w_{ij} &= (w_1)_{ij} \text{ for } i = 1,2, \ldots, s \text{ and } j = 1,2, \ldots, k \\
&= (w_2)_{ij} \text{ for } i = r, r+1, \ldots, n \text{ and } j = k+1, k+2, \ldots, m \\
&= 0 \text{ otherwise}
\end{aligned} \tag{3.2}
$$

Let us assume that

$X = (e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_s, e_{s+1}, \ldots, e_n)$ is an invariant of G.

$$\Leftrightarrow \sum_{i=1}^{n} e_i * w_{ij} = 0 \quad \forall\ j \in J$$

$$\Leftrightarrow \sum_{i=1}^{s} e_i * w_{ij} = 0 \quad \forall\ j \in J_1 \text{ , and } \sum_{i=r}^{n} e_i * w_{ij} = 0 \quad \forall\ j \in J_2 \quad \cdots \text{by}(3.2)$$

$$\Leftrightarrow \sum_{i=1}^{s} e_i * (w_1)_{ij} = 0 \quad \forall\ j \in J_1 \text{ , and } \sum_{i=r}^{n} e_i * (w_2)_{ij} = 0 \quad \forall\ j \in J_2 \cdots \text{by}(3.2)$$

$$\Leftrightarrow \left. \begin{array}{l} X_1 = (e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_s) \text{ is an invariant of } G_1 \text{ and} \\ X_2 = (e_r, e_{r+1}, \ldots, e_s, e_{s+1}, \ldots, e_n) \text{ is an invariant of } G_2. \end{array} \right\}$$

Consequence 1 : Let $G_1 = (P_1, T_1, IN_1, OUT_1)$ and $G_2 = (P_2, T_2, IN_2, OUT_2)$ be two PTNs such that

(1) $P_1 = \{p_1, p_2, \ldots, p_r, p_{r+1}, \ldots, p_s\}$
(2) $P_2 = \{p_r, p_{r+1}, \ldots, p_s, p_{s+1}, \ldots, p_n\}$
(3) $T_1 \cap T_2 = \phi$

If G is the union of G_1 and G_2, then the row-vector of integers

$X = (e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_s, e_{s+1}, \ldots, e_n)$

is an invariant of G iff the row-vector

$X_1 = (e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_s)$

is an invariant of G_1 and the row-vector

$X_2 = (e_r, e_{r+1}, \ldots, e_s, e_{s+1}, \ldots, e_n)$

is an invariant of G_2.

Consequence 2 : Let $G_i = (P_i, T_i, C_i, IN_i, OUT_i)$ for $i = 1, 2, ..., t$ be CPNs satisfying the conditions of Theorem 3.1. If G is the union of the above CPNs, then the invariants of G can be computed from those of G_i's and vice versa by repeated application of Theorem 3.1.

4. EXAMPLES

4.1 : Readers-Writers System

We consider, here, the version of the Readers-Writers problem discussed in [5]. Figures 1(a) and 1(b) depict the PTN models G_1 and G_2 of the 'reader' process and the 'writer' process respectively. The interpretation of the places and transitions of G_1 and G_2 are as follows:

LP : Local processing, where the shared memory is not used
WR : Waiting to 'read'
WW : Waiting to 'write'
R : Reading in progress
W : Writing in progress
S : Synchronization, to enforce mutual exclusion of writers
t_1 : A 'reader' arrives
t_2 : A 'writer' arrives
t_3 : A 'reader' starts reading
t_4 : A 'writer' starts writing
t_5 : A 'reader' finishes reading
t_6 : A 'writer' finishes writing

Figures 1(a) and 1(b) show also the invariants of G_1 and G_2. Figure 1(c) gives the PTN model G of the overall system. It can be easily seen that G is the union of G_1 and G_2. To determine the invariants of G, we apply Theorem 3.1 to G_1 and G_2, which satisfy all requirements of the theorem. On applying the theorem, we get $e_1 = e_3$ and $e_2 = e_4$ and the invariants of G, which are displayed in Figure 1(c), are thus easily obtained. The second row in this figure gives the invariants in their general form whereas the third and fourth rows give two linearly independent invariants obtained on substituting $e_1 = 1, e_2 = 0$ and $e_1 = 0, e_2 = 1$ respectively. Using these invariants, the system can be analysed as described in [5].

4.2 : A Resource Sharing System

In this example, we consider a concurrent system consisting of a set of processes, $PROC = \{p_1, p_2, ..., p_s\}$, $s > 0$, and a set of resources, $RSC = \{r_1, r_2, ..., r_t\}$, $t > 0$. Each process involves two stages of processing. RES_1 and RES_2 are two (s x t) matrices which specify the resource requirement of the processes in the first and second stages of processing respectively. Formally, RES_1 and RES_2 are given by:

For $i = 1,2, \ldots, s$ and $j = 1,2, \ldots, t$,
$(RES_1)_{ij} = 1$ if p_i requires r_j for first stage
$= 0$ otherwise
$(RES_2)_{ij} = 1$ if p_i requires r_j for second stage
$= 0$ otherwise.

Figures 2(a) and 2(b) depict the CPN models G_1 and G_2 of the first stage and second stage of processing respectively. The interpretation of the places and the transitions is as follows:

P : Ready-to-run processes
R : Available resources
A_1 : First stage of processing
A_2 : Second stage of processing
W : Waiting for resources in order to enter the second stage of processing
t_1 : A process starts first stage of processing
t_2 : A process finishes first stage of processing
t_3 : A process starts second stage of processing
t_4 : A process finishes second stage of processing.

The colour set attached to the place R is RSC. The colour set PROC is attached to all other places and all transitions.

Each unlabelled arc in G_1 and G_2 has a weight equal to the identity function of the proper order. For example, the weight of the arc between the place P and the transition t_1 is the identity function of order 's' since both P and t_1 have the colour set PROC, whose cardinality is 's'.

Figures 2(a) and 2(b) show also the invariants of G_1 and G_2 respectively. In these invariants, '*' denotes matrix multiplication and '+' denotes matrix addition. The overall system has a CPN model G which is the union of G_1 and G_2. G is shown in Figure 2(c). We can invoke Theorem 3.1 to determine the invariants of G. Applying Theorem 3.1, we get $e_1 = e_3$ and $e_2 = e_4$ and hence the invariants of G can be easily obtained.

Figure 2(d) shows the invariants of G. The second row of this figure gives the invariants in a general form, using which we obtain two particular cases shown in the third and fourth rows. The invariant of the third row is obtained by substituting the s^{th} order identity matrix for e_1 and the zero matrix for e_2. This gives an invariant with respect to the set PROC. By substituting the zero matrix for e_1 and the t^{th} order identity matrix for e_2, we get the invariant shown in the fourth row. This invariant is with respect to the set RSC. The above invariants can be used in the analysis of the system if the matrices RES_1 and RES_2 are known.

4.3 : A Network of Databases

We now consider the distributed data base system discussed in [2,5]. This system comprises a set of database managers, $DBM = \{d_1, d_2, \ldots, d_n\}$, $n > 0$, who communicate among themselves, via a fixed set of message buffers, $MB = \{ \langle s,r \rangle : s,r \in DBM, s \neq r\}$. Each manager can be in three states : 'inactive', 'waiting for acknowledgements' and 'performing an update on request of another manager'. Each message buffer may be in four states : 'unused', 'sent', 'received', and 'acknowledged'. Whenever a manager, say d_1, wants to perform an update, he waits for all other managers to become 'inactive' and then performs an update on his database. Simultaneously, he will flash messages using his message buffers to all other managers informing them of his updating. These managers, on receipt of the message from d_1, update their own databases and send acknowledgements to d_1, who becomes 'inactive' after such acknowledgements are received from all managers. Now, another manager may perform an update and send messages.

The activities in the above system consist of the following events:

(1) Update and send messages,
(2) Receive messages,
(3) Send acknowledgement, and
(4) Receive acknowledgement.

These activities are represented as the CPN models G_1, G_2, G_3, and G_4 in Figures 3(a)-3(d) in that order. The transitions t_1, t_2, t_3, and t_4 represent the above four events (1)-(4) respectively. The interpretation of the places is as follows:

p_1 : 'Inactive' managers
p_2 : 'Exclusion', to enforce mutual exclusion of updating managers
p_3 : 'Unused' message buffers
p_4 : 'Sent' message buffers
p_5 : Manager 'waiting' for acknowledgements
p_6 : Managers 'performing' an update on request of another manager
p_7 : 'Received' message buffers
p_8 : 'Acknowledged' message buffers.

The colour sets attached to the places and transitions are as follows:

$C(p_1) = C(p_5) = C(p_6) = C(t_1) = C(t_4) = DBM$
$C(p_3) = C(p_4) = C(p_7) = C(p_8) = C(t_2) = C(t_3) = MB$
$C(p_2) = E$ where $E = \{\epsilon\}$,'ϵ'representing tokens without any colour.

The labels ABS, RES, and MINE on the arcs are the matrix versions of the functions of the same names described in [5]. As in the previous example, unlabelled arcs have weights equal to identity matrices of appropriate orders.

Figures 3(a)-3(d) also display the invariants of the CPN models G_1, G_2, G_3, and G_4. It may be noted again that '*' represents matrix multiplication, '+' represents matrix addition, and '-' represents matrix subtraction. On applying Theorem 3.1 to G_1, G_2, G_3, and G_4, we get

$e_1 = e_6 = e_{10} = e_{11}$,
$e_2 = e_{12}$,
$e_3 = e_{13}$,
$e_4 = e_7 = e_{14}$,
$e_5 = e_8$, and
$e_9 = e_6 * REC + e_7 - e_5 * REC.$

Using the above equations, we can immediately arrive at the invariants of the overall CPN model shown in Figure 4(a) which is only a shuffled version of Figure 13 in [5]. Figure 4(b) gives the invariants of the above CPN model. The second row gives the invariants in a general form and the next five rows give five linearly independent invariants that can be obtained, by suitable substitutions, from the general form. It may be noted that the invariants i1-i5 listed in Figure 14 of [5] can be easily obtained using the above five invariants. Using these invariants, we can analyse the system as in [5].

5. CONCLUSION

We have presented a theorem which facilitates computation of the invariants of a coloured Petri Net that is obtained by coalescing a finite number of coloured nets whose invariants are known. Since place-transition Nets are a special case of coloured Petri Nets, the theorem holds for place-transition Nets also. We have shown via three examples of concurrent systems how this theorem is useful in the analysis of concurrent systems. The theorem is particularly effective when a concurrent system is synthesized in a bottom-up fashion starting from very high level specifications. For, in such a case, at each stage of synthesis, we will be synthesizing a net model of the system by coalescing the net models of the previous stage and we can invoke the theorem to compute the invariants of the synthesized net model. The application of the theorem requires that we know the invariants of the individual nets or the low level nets which generally consist of one or two transitions. The invariants of the low level nets can be computed in most of the cases by direct calculation. In some cases, however, the computation of invariants of the low level nets is non-trivial and we are currently working towards solving this problem.

ACKNOWLEDGEMENTS

Our thanks are due to the three anonymous referees of this paper for their very valuable comments and suggestions.

REFERENCES

1. T. Agerwala and Y.C. Amphai, 'A Synthesis Rule for Concurrent Systems', Proceedings of IEEE Conference on Design Automation, 1978, pp. 305-311.

2. H.J. Genrich and K. Lautenbach, 'The Analysis of distributed systems by predicate/ transition nets', Semantics of Concurrent Computation, Evian 1979, G. Kahn (ed.), Lecture Notes in Computer Science, Vol.70, Springer verlag 1979, pp. 123-146.

3. H.J. Genrich and K. Lautenbach, 'System Modelling with High-Level Petri Nets', Theoretical Computer Science, Vol.13, 1981, pp. 109-136, North-Holland Publishing Company.

4. M. Jantzen and R. Valk, 'Formal Properties of Place/Transition Nets', Net Theory and Applications, Hamburg 1979, W. Brauer (ed.), Lecture Notes in Computer Science, Vol.84, Springer-Verlag 1980, pp. 165-212.

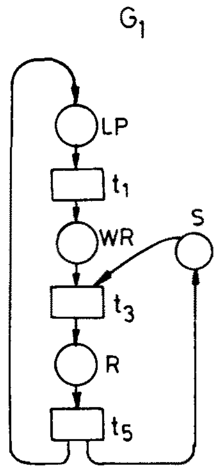
5. K. Jensen, 'Coloured Petri Nets and the Invariant Method', Theoretical Computer Science, Vol.14, 1981, pp. 317-336, North-Holland Publishing Company.

6. K. Jensen, 'How to find Invariants for Coloured Petri Nets', Mathematical Foundations of Computer Science, 1981, J. Gruska and M. Chytil (eds.), Lecture Notes in Computer Science, Vol.118, Springer-Verlag 1981, pp. 327-338.

7. G. Memmi and G. Roucairol, 'Linear Algebra in Net Theory', Net Theory and Applications, Hamburg 1979, W. Brauer (ed.), Lecture Notes in Computer Science, Vol.84, Springer-Verlag 1980, pp. 213-224.

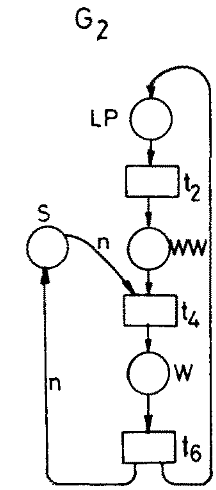
8. Y. Narahari and N. Viswanadham, 'A Petri Net Approach to the modelling and analysis of Flexible manufacturing systems', To appear in : Annals of Operations Research, Vol.3, 1986.

9. J.L. Peterson, 'Petri Nets', ACM Computing Surveys, Vol.9, No.3, September 1977, pp. 223-252.



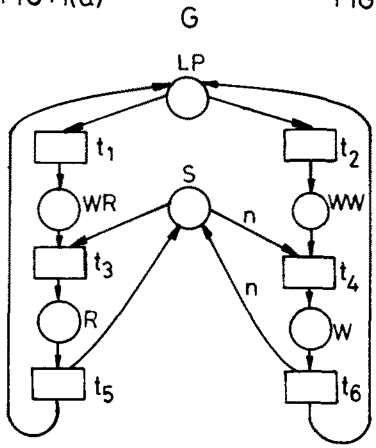
LP	WR	S	R
e_1	e_1	e_2	$e_1 + e_2$

FIG. 1(a)



LP	WW	S	W
e_3	e_3	e_4	$e_3 + ne_4$

FIG. 1(b)



LP	WR	S	R	WW	W
e_1	e_1	e_2	$e_1 + e_2$	e_1	$e_1 + ne_2$
1	1	0	1	1	1
0	0	1	1	0	n

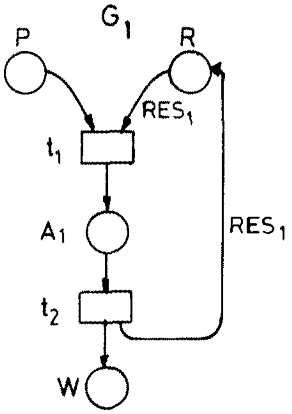
FIG. 1(c)

Figure 1. Readers-Writers problem (Example 1). PTN models and place invariants for

(a) 'Reader' process,

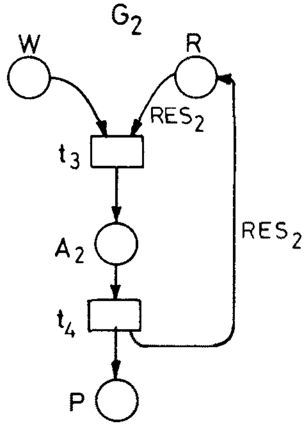
(b) 'Writer' process, and

(c) overall system .



P	R	A_1	W
e_1	e_2	$e_1+e_2 * RES_1$	e_1

FIG 2(a)



W	R	A_2	P
e_3	e_4	$e_3+e_4 * RES_2$	e_3

FIG. 2(b)

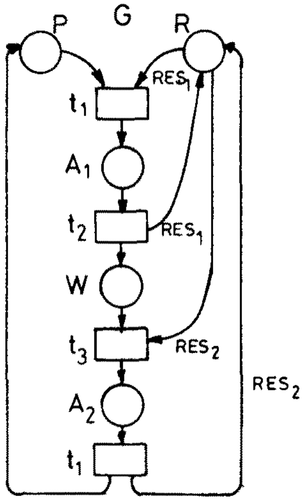


FIG 2(c)

P	R	A_1	W	A_2
e_1	e_2	$e_1+e_2*RES_1$	e_1	$e_1+e_2*RES_2$
ID	0	ID	ID	ID
0	ID	RES_1	0	RES_2

FIG. 2 (d)

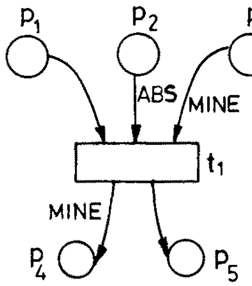
Figure 2. Resource sharing system (Example 2).

(a) CPN model and place invariants of the first stage of processing

(b) CPN model and place invariants of the second stage of processing

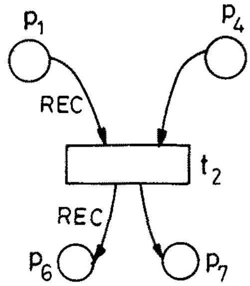
(c) CPN model of the overall system

(d) Place invariants of the above CPN model.



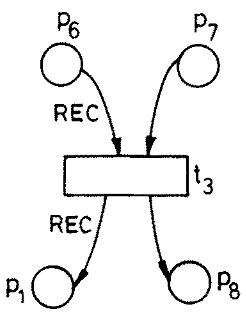
p_1	p_2	p_3	p_4	p_5
e_1	e_2	e_3	e_4	$e_1 + e_2 * ABS$ $+ e_3 * MINE - e_4 * MINE$

FIG. 3 (a)



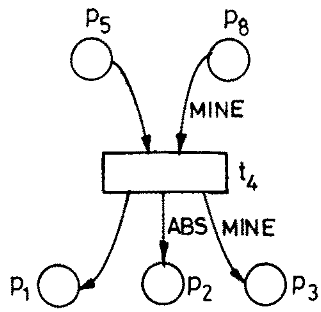
p_1	p_4	p_6	p_7
e_6	e_7	e_5	$e_6 * REC + e_7$ $- e_5 * REC$

FIG. 3 (b)



p_6	p_7	p_1	p_8
e_8	e_9	e_{10}	$e_8 * REC + e_9$ $- e_{10} * REC$

FIG. 3 (c)



p_1	p_2	p_3	p_8	p_5
e_{11}	e_{12}	e_{13}	e_{14}	$e_{11} + e_{12} * ABS$ $+ e_{13} * MINE - e_{14} * MINE$

FIG. 3 (d)

Figure 3. Network of data bases (Example 3).
CPN models and place invariants of the processes

(a) Update and send messages
(b) Receive messages
(c) Send acknowledgement
(d) Receive acknowledgement

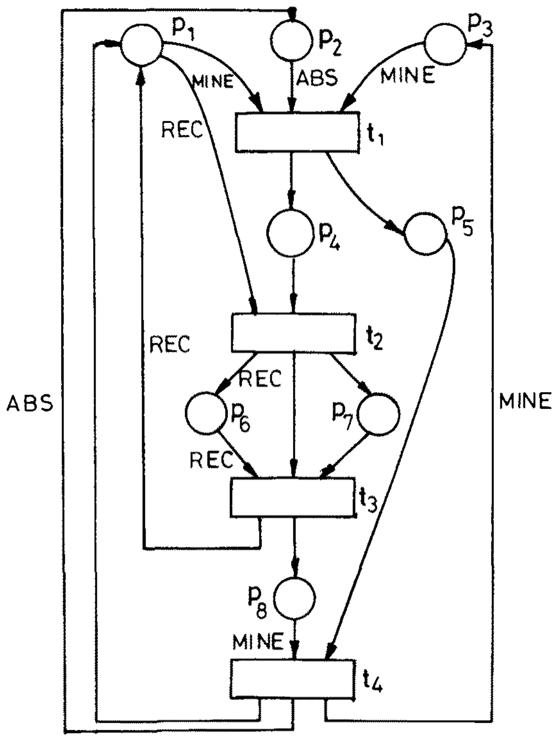


FIG. 4 (a)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
e_1	e_2	e_3	e_4	$e_1 + e_2 * ABS + e_3 * MINE - e_4 * MINE$	e_5	$e_4 + e_1 * REC - e_5 * REC$	e_4
ID	0	0	0	ID	0	REC	0
0	ID	0	0	ABS	0	0	0
0	0	ID	0	MINE	0	0	0
0	0	0	ID	-MINE	0	ID	ID
0	0	0	0	0	ID	-REC	0

FIG. 4 (b)

Figure 4. (a) Overall CPN model of the network of data bases system

(b) Place invariants of the above CPN model.