

Model development of MIMO Nets: A H-EPN Based Approach

S. Ramaswamy^{*†}, K. P. Valavanis[‡], S. Barber^{*}

Abstract

The construction and analysis of Multiple Input Multiple Output (MIMO) subnets is discussed. It is shown that Hierarchical Time-Extended Petri Nets (H-EPNs) allow the development of structured MIMO subnets through the use of bottom-up Petri Net synthesis techniques. The paper analyzes Petri Net extensions that allow a truly hybrid approach to Petri Net based systems modeling, analysis and development. The case study emphasizes the advantage of the activator arc extension in studying issues such as static priority scheduling, dynamic failure recognition, and rescheduling in manufacturing systems. The SPNP package, suitably modified to handle the H-EPN extensions, is used to analyze the properties of the derived H-EPN model.

1. Introduction

In this paper, a class of PNs, called Hierarchical Time-Extended Petri Nets (H-EPNs) are used to derive and analyze Multiple Input Multiple Output (MIMO) system models. Research on PN extensions to systems modeling / analysis have focused on the use of SISO subnet structures because they are easy to develop and analyze. More on these approaches may be found in [2, 5] and references therein. However, PN models of most real world systems tend to be MIMO nets. The H-EPN approach to PN modeling addresses this issue by adding the notion of activator arcs and modeling independent subnet initiations (separately defined subnets as well as independent subnets within a MIMO subnet) by means of conjugate places. Activator arcs have been derived from using Extended Petri Nets (EPNs) in the modeling and analysis of materials handling systems [3].

This paper provides an extension that will allow the construction and analysis of MIMO nets. This PN extension will also facilitate the modeling and analysis of two different kinds of failure situations: (i) *Static failure situations*: These are failure situations that are known or can be decided prior to scheduling critical activities, (ii) *Dynamic failure situations*: These are failure situations that occur during certain critical system operations that require the use of redundant, standby resources and / or mechanisms. Almost all PN related research and modeling techniques have neglected the importance of capturing such dynamic failure situations. Recently in [6], a PN extension called Augmented Timed Petri Nets (ATPNs) is presented to handle such dynamic failure situations. However, the activator arc extension to model MIMO nets as proposed in this paper is syntactically simpler and semantically easier to understand.

The paper is organized as follows: Section two provides an overview of H-EPN basics. More on H-EPNs may be

found in [4]. Section three deals with the development and analysis of MIMO nets, and some of their structural and timing properties. Section four presents a small example to illustrate the approach developed in section three. The H-EPN model derived for the example also illustrates the advantages gained in using activator arcs to model and analyze dynamic failure situations, priority scheduling and self-monitoring of input job queues. Section five concludes the paper.

2. H-EPNs

H-EPNs include: (i) five different types of places (status, action, decision, subnet and the source-sink places), (ii) two different types of transitions (internally and externally driven transitions), (iii) two different types of unweighted arcs (inhibitor and activator arcs), and, (iv) two different types of tokens (control and flow tokens represented as solid and dotted tokens, respectively).

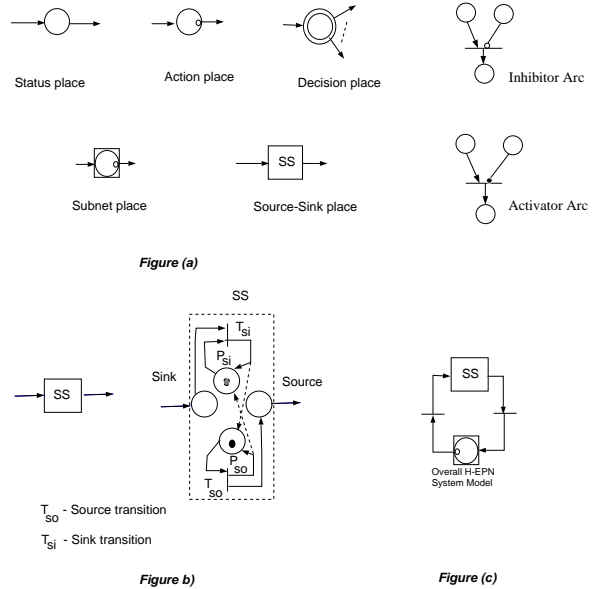


Figure 1. Places in an H-EPN

The graphical representation of the places and arcs in a H-EPN design is shown in Figure 1a. The structure for the *ss* place implicitly adheres to the property that every source place has a corresponding sink place in the net, thereby allowing for token conservation. All flow tokens that enter the system must exit the system. This factor is important for ensuring the property of system boundedness. As with the classical PN theory all transitions that have their input places marked are considered enabled and are potentially fireable. But since H-EPNs are proposed as tools that integrate with actual real-world systems for real-time control, the enabling of transitions is a two-step process as defined below: (i) **Place Enabling**: Place enabling of transitions is similar to the classical PN approach. This means that if all the input places to a H-EPN transition are appropriately marked then the transition is considered place enabled, and, (ii) **Event**

^{*} The Laboratory for Intelligent Processes and Systems, The Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712-1084. This research was supported in part by the Texas Higher Education Coordinating Board, under grant ATPD-112.

[†]Address all correspondences to Dr. S. Ramaswamy.

[‡] The Robotics and Automation Laboratory, The Center for Advanced Computer Studies and the Apparel-CIM Center, The University of Southwestern Louisiana, Lafayette, LA 70504. This research was supported in part by the Defense Logistics Agency / Department of Defense, under grant R-1043.

Enabling: Event enabling is an extension of H-EPNs to deal with discrete event dynamic systems (DEDS), where state changes are primarily derived by the occurrence of internal or external events. The occurrence of these events is signified by the enable time. Therefore, only when a transition is both *place and event enabled* it is considered to be enabled. In the H-EPN system model, an enabled transition implies that events associated with the transition are expected to occur within the specified time limits. Thus, all transitions in the H-EPN model may be represented as:

$$\{ \langle \langle P_{in} \rangle, \langle P_Q, P_R \rangle, \langle \psi(E_{in}) \rangle \rangle; \langle \langle \phi(P_{in}) \rangle, m, \langle \omega(P_{in,t}) \rangle \rangle; \langle \langle P_{out} \rangle, \langle \xi(E_{out}) \rangle \rangle \} \quad (1).$$

where, P_{in} and P_{out} are sets of input and output places to the transition, t_i , respectively, E_{in} and E_{out} are sets of input and output events associated with the transition, t_i , respectively, $P_{in,t}$ is a subset of the input places, P_{in} , to the transition, t_i , that are associated with some system operations abstracted by either action or subnet places in a H-EPN system model, $t(p_i) = \tau_i$, the time associated with place $p_i \in P_{in}$, ψ is a guard function that is true if all associated events, $e_i \in E_{in}$, are true, ω is a watchdog function that monitors the critical operations represented by the set $P_{in,t}$, and thus ensures correct system operations, ϕ is a random function that chooses some time value τ from a given set of values, to be associated with the transition t_i such that $\tau_{imin} \leq \tau \leq \tau_{imax}$, where τ_{imin} and τ_{imax} are the minimum and maximum times associated with some input places $p_i, p_j \in P_{in}$, m is the transition multiplicity, ξ is an output function which initiates some event processing thereby leading to an event generation or assigns values to appropriate events if they are B, I or A type events, P_Q and P_R are the set of places that inhibit and activate the transition, t_i , respectively.

Important system properties for the H-EPN model are analyzed using already available PN simulation tools like SPNP [1]. However, to use such tools, the H-EPN system model must first be transformed into a system model that does not contain activator arcs. This transformation essentially produces at least one extra place and arc for every activator arc in the H-EPN system model. Moreover, these tools also do not allow the incorporation of events in the system model. However, this drawback is easily offset by modeling the system with additional places that represent the generation of these events. For example, the error event that triggers the error operations in the example discussed in section four (Figure 8b), is modeled by means of additional places. However, to be used in real-time control, the transitions (tR1e, and tR1r) that model these failures are associated with real-time sensory events and are not driven by just the input places. The H-EPN extension to transitions, place and event enabling, address exactly this characteristic of real-time systems.

3. MIMO Net Development and Analysis

H-EPN MIMO subnets are defined by means of a bottom-up approach to PN model generation, similar to various other reported methods [2, 5]. Independently defined SISO subnets are combined through common places, transitions and arcs to define an interacting or a MIMO subnet. Such a MIMO subnet is then transformed to a corresponding SISO subnet for easy integration as a SISO net into a top-down decomposition of the overall system model. The

advantage of this approach is that previously established analysis techniques can be easily adapted to be used within this framework. Moreover, highly complex PN structures may be easily generated by means of this approach.

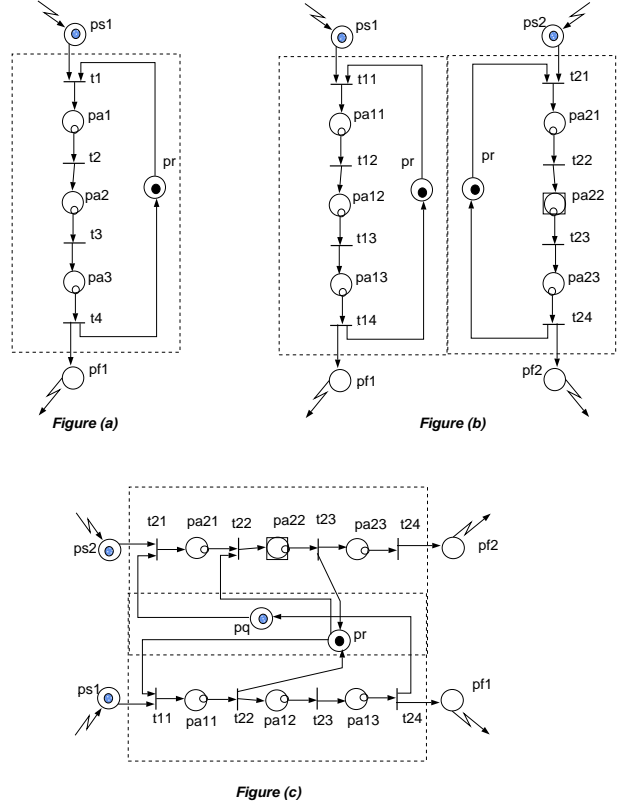


Figure 2. SISO/MIMO PN Examples

3.1. MIMO Net Generation

In H-EPNs, MIMO subnets are created due to operation abstraction with respect to the resources used. This implies that if two or more operations are performed using a resource (or a set of resources) then such operations are abstracted at a higher level of modeling detail as a single context sensitive MIMO net. The context sensitivity is achieved by the notion of conjugate place. This means that different instantiations of the same MIMO subnet corresponding to the respective SISO subnets constituting the MIMO subnet are possible. Information regarding a particular instantiation is maintained by means of the conjugate place at the higher level net. Figure 2b illustrates a simple MIMO subnet generated by a shared resource place, p_r . Figure 2c shows a more complex MIMO subnet generated by shared places, p_r and p_q , where p_r is a shared resource and p_q sequences the operations of the two SISO subnets of the MIMO subnet. It is to be noted that when such a MIMO subnet place is used at a higher level of abstraction, the two individual SISO nets are not essentially activated simultaneously. They may occur in sequence but other operations represented at the higher level net or by other MIMO nets may be inter-spaced between these two operations. Moreover, when they are not initiated alternatively by a higher level net, deadlock may occur. Although providing greater modeling flexibility, this approach to PN modeling necessitates greater care while designing higher level nets and

corresponding lower level MIMO subnets. It is to be noted that when we talk about subnet liveness we actually refer to the quasi-liveness of subnets. In the following, essential definitions for subsequent developments in this section are provided.

Definition 1: Well Formed Subnet (WFS): A well formed subnet is a SISO marked graph that is bounded, live and reversible. The net in Figure 2a is a WFS.

Definition 2: Successor ($\sigma(t_i)$): A transition t_j is said to be a successor of a transition t_i , $\sigma(t_i)$, if there exists a place, p_k , or a set of places, $p_k \dots p_{k+r}$, such that:

$$(t_i) \bullet = p_k = \bullet(t_{i+1})$$

.....

$$(t_{i+r}) \bullet = p_{k+r} = \bullet(t_j)$$

$$\text{Thus, } t_{i+1} = \sigma(t_i) \text{ and } t_{i+r} = \sigma^*(t_i), \quad (2)$$

where the * indicates that the distance is greater than a unit,

that is, the transitions are not immediate

successors but are probable distant successors.

Note: Observe that if a net is reversible, then $\sigma^*(t_i) = t_j$ and $\sigma^*(t_j) = t_i$.

Definition 3: Well Defined Block (WDB): A WDB is a SISO subnet place, p_{su}^j such that if a *ss* place p_{ss}^i is introduced such that equation (2) is true, then the combined net is live, bounded and reversible. Thus, a well formed subnet is a WDB. Figure 2a is an example of a WDB.

Let p_s and p_f be the first

and last places of the subnet place p_{su}^j

$$\bullet(p_s) = t_f = (p_{ss}^i) \bullet \text{ and } (p_f) \bullet = t_s = \bullet(p_{ss}^i) \quad (3)$$

Definition 4: Interacting Subnet (IS): An interacting subnet is made of two or more WDB's p_{su}^j $1 \leq j \leq n$; $n \geq 2$; n being the number of WDB's sharing the same set of resources.

Definition 5: MIMO Subnet: If the individual WDBs of an interacting subnet of two or more WDB's can be ordered such that: (i) equation (3) is satisfied, and, (ii) the combined net is live, bounded and reversible, then it is called a well formed MIMO subnet. In such an ordering, a set of *ss* places p_{ss}^i $1 \leq i \leq n$; $n > 2$; are introduced to form a combined net with the interacting subnet. Figures 2b and 2c are examples of well formed MIMO subnets.

Let p_{sj} and p_{fj} be the first

and last places of the subnet place p_{su}^j

$$\bullet(p_{s1}) = t_{f1} = (p_{ss}^1) \bullet \text{ and } (p_{f1}) \bullet = t_{s2} = \bullet(p_{ss}^2),$$

.....

$$\bullet(p_{sj}) = t_{fj} = (p_{ss}^j) \bullet \text{ and } (p_{fj}) \bullet = t_{si} = \bullet(p_{ss}^i),$$

.....

$$\bullet(p_{sn}) = t_{fn} = (p_{ss}^n) \bullet \text{ and } (p_{fn}) \bullet = t_{s1} = \bullet(p_{ss}^1)$$

Definition 6: SU-Connection (SUC): A SU-Connection (refer Figure 3a) (subnet connection) is said to be established between a net N and a MIMO subnet $X_{mimo} \in N$, when a WDB, $p_{su}^j \in X$ and its conjugate place in N are marked by a transition firing in N .

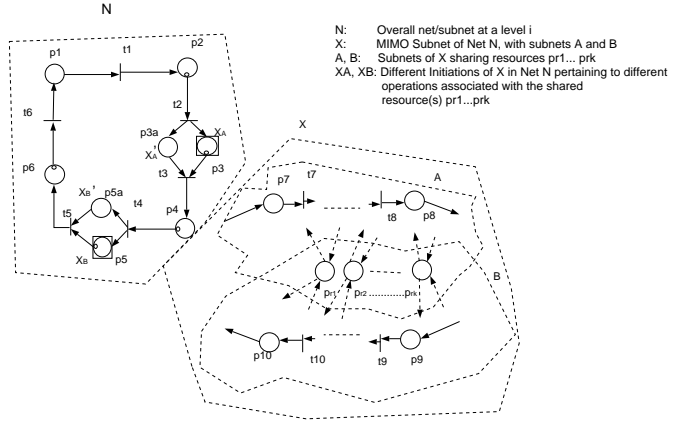


Figure (a)

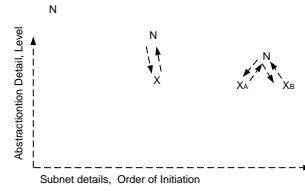


Figure (b)

Figure 3. MIMO Net structure

Definition 7: Subnet Activation Time (${}^N t_i(X_A)$): ${}^N t_i(X_A)$, represents the i th instance of activation of a subnet X_A in net N .

Definition 8: QR Set: A transition, t_i , is said to be associated with a QR set, if there exist a set of disjoint places P_Q and P_R such that P_Q is the set of places which inhibit t_i and P_R is the set of places which activate t_i .

Definition 9: Dependent and Independent WDBs: Suppose there exist 3 WDB's X_A , X_B and X_C belonging to a MIMO subnet X_{mimo} of a net N . Then, X_C is an independent WDB and X_A , X_B are dependent WDBs if:

$$\begin{aligned} \forall p_i \in X_A, p_j \in X_B, p_k \in X_C \\ X_C \cap X_A = X_C \cap X_B = P_{r_i}, 1 \leq i \leq s, s \leq k, \text{ and} \\ X_A \cap X_B = P_{r_j}, 1 \leq j \leq k, \text{ and } \exists (p_m) \in P_{r_j}: \end{aligned} \quad (5)$$

$$\bullet(p_m) \in X_A, (p_m) \bullet \in X_B, (p_m) \notin P_{r_i}, \text{ where,}$$

s : # resources shared by the WDBs,

k : Total # places shared by the WDBs.

That is, X_C is not affected by any shared places if all the shared resources are available for its initiation, whereas X_A and X_B are sequenced by a shared place.

The following are then observed from the above definitions regarding the resource, $p_r \in P_{r_i}$, $1 \leq i \leq s$, and p_m :

- p_r is used in the operations of Net N outside of X if X_A , X_B are temporally spaced in N .
- p_m is used in the communication between X_A , X_B when they are triggered alternatively or simultaneously (that is, $\bullet(X_A) = \bullet(X_B)$) in N .

Moreover, it can be noticed that the MIMO subnet in Figure 2b consists of 2 independent subnets while the MIMO subnet of Figure 2c consists of two dependent subnets, which share a common place p_q that determines the liveness of the MIMO subnet, and hence the overall net.

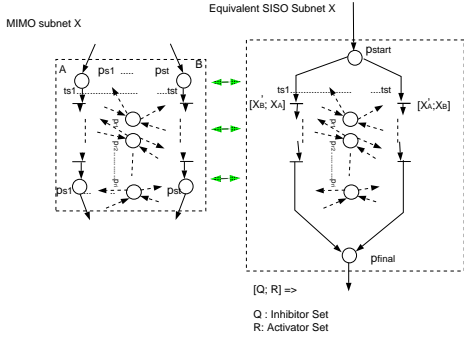


Figure 4. MIMO — SISO Translation

3.2. Properties of MIMO Nets

The following discussion considers the net in Figure 3. Let X be the MIMO subnet of a net N consisting of 2 WDBs X_A and X_B such that X_A and X_B are initiated by N at different time instances. Let X_A and X_B encapsulate a resource represented by a status place $p_{ri} \in X$.

The following facts can then be observed / established:

Fact 1: A MIMO subnet is made up of two or more WDB's.

Fact 2: If a resource (or a set of resources) represented by a place p_{ri} is required in a higher level net (Net N in Figure 3) then there exists at least 2 WDB's X_A and $X_B \in X$, $X \in N$ and $X \in P_{su}$ which oversee the request and release of the resource(s) when resource failure identification is built into the request and release operation.

Theorem 1: A *ss* place is a WDB.

Proof: The proof of this theorem is trivial. If two *ss* places p_{ss}^i and p_{ss}^j are combined using two temporary transitions t_s and t_r , one of the *ss* places, (for example, p_{ss}^i), can be considered to function as a WDB connected with a *ss* place, p_{ss}^j . Equation 5 holds for this combined net. ■

Theorem 2: For every well formed MIMO subnet, say X_{mimo} , there exists a well formed SISO subnet, X_{siso} , that represents the same set of operations, such that operations corresponding to the individual WDB's in X_{mimo} , say h , correspond to h branches that are activated due to the firing of one of the h transitions that are output to the single input place, p_s , to X_{siso} .

Proof: It is to be recalled that H-EPNs allow the use of inhibitor and activator arcs. The MIMO-SISO transformation is a 2-step process as illustrated in Figure 4:

- **Creating a SISO structure:** Remove all the input places of the WDBs and replace them by a single decision place, p_s , with output arcs running from the decision place to all the output transitions, say $t_h \in T_h$, (T_h is a set of transitions) of the places removed. Remove all output places from the individual WDBs and replace them by a single place p_f which will be the final output place for all the WDBs. The transition that this place will fire at the higher level net on completion of the subnet operation will depend on the conjugate place that was simultaneously marked

- **QR set development:** For each transition in T_h , that is output to the place p_s , the set Q consists of all the places that inhibit the transition from firing, and the set R consists of the corresponding conjugate place (places with implicit activator arcs to the corresponding transitions) at the higher level net. Sets Q and R may also contain other constraints that can affect the selection of the particular WDB of the MIMO subnet.

Thus, these new transitions with QR sets will have two more conditions to satisfy before they are enabled [4]. These conditions correspond to the activating and inhibiting places (all such places may not necessarily be conjugate places at the higher level net, they may also be other places at the higher level net) that are associated with the corresponding transitions³. ■

Theorem 3: Let N be the net that contains the MIMO subnet X_{mimo} with a correct initial marking μ_{N_0} . Let X_A and X_B be WDBs of X that are SUC in N . Let S be the net obtained after the SUC of X_A (or X_B). Then:

- S is bounded $\Leftrightarrow N$ is bounded.
- S is live $\Leftrightarrow N$ is live.
- S is reversible $\Leftrightarrow N$ is reversible.

Proof: Let μ_{S_0} be the correct initial marking of S such that for every place $p_i \in P_S$, $\mu_{S_0}(p_i) = \mu_0(p_i)$ and for every place $p_i \in P_N$, $\mu_{N_0}(p_i) = \mu_0(p_i)$.

Let X_A and X_B subnets in $X_{mimo} \in N$, and let X_A' and X_B' be the corresponding conjugate places in N . Let p_{sa} and p_{sb} be the starting places of X_A and X_B in X_{mimo} . Let $[Q_a, R_a]$ and $[Q_b, R_b]$ be the QR sets of $(p_{sa})^\bullet$ and $(p_{sb})^\bullet$ in a corresponding SISO transformation.

By Definition 5, X_{mimo} is bounded, live and reversible. Two cases may be distinguished. The WDBs $X_A, X_B \in X_{mimo}$ may be either dependent or independent WDBs.

Case 1: $X_A, X_B \in X_{mimo}$ are independent WDBs. In this case, there do not exist places in X_{mimo} , or the equivalent X_{siso} , that sequence the subnets X_A and X_B .

$$\text{Let, } P_A = P_{X_A}, P_B = P_{X_B} \text{ and } P_X = P_A \cup P_B,$$

$$N = P_N \cup T_N, \text{ and } S = P_N \cup T_N \cup P_X$$

$$\forall p_i \in P_S \cap P_A, \mu_{S_0}(p_i) = \mu_{A_0}(p_i) = \mu_0(p_i) \quad (6)$$

$$\forall p_j \in P_S \cap P_N, \mu_{S_0}(p_j) = \mu_{N_0}(p_j) = \mu_0(p_j)$$

$\mu_0(p_i)$ and $\mu_0(p_j)$ are the correct initial

markings of X_A and N respectively.

Thus, since, X_A, X_B are independent WDBs

- S is bounded $\Leftrightarrow N$ is bounded.
- S is live $\Leftrightarrow N$ is live.
- S is reversible $\Leftrightarrow N$ is reversible.

Case 2: $X_A, X_B \in X_{mimo}$ are dependent WDBs.

- N is live, bounded and reversible implies that:

- X_A, X_B have an ordering (A, B), that is, \exists a sequence of transitions and places, $t_i, p_{su}^a, t_{i+1}, p_{i+1}, \dots, t_j, p_{su}^b$ in N , such that:

$$\bullet (p_{su}^a) = \bullet (X_A') = t_i, (p_{su}^a)^\bullet = (X_A')^\bullet = t_{i+1} = \bullet (p_{i+1}) \quad (7)$$

$$\dots$$

$$t_j = \bullet (p_{su}^b) = \bullet (X_B')$$

- Thus, $N_{t_i}(X_A) < N_{t_i}(X_B)$ and $N_{t_{i+1}}(X_A) > N_{t_{i+1}}(X_B)$.
- $X_A' \in Q_b, X_B' \in Q_a, X_A' \in R_a$, and $X_B' \in R_b$ in the equivalent X_{siso} transformation.

³ To use PN simulation tools that do not allow the use of activator arcs, for simulation / analysis purposes, the technique discussed for the example in section IV is used. However, note that this technique adds to the net complexity either in terms of additional places, or additional arcs with the specification of transition priorities.

Thus, the SUC in N alternates between SUC_A and SUC_B . Thus, S is bounded, live and reversible.

- b. To prove that N is bounded, live and reversible if S is bounded, live and reversible refer to Equation 5. S consists of places and transitions from N and X_{mimo} . If S is bounded, live and reversible then X_{mimo} is bounded, live and reversible. And $\mu_{S0}(p_i) = \mu_{N0}(p_j) \cup \mu_{A0}(p_k)$, where, $p_i \in P_S$, and $p_j \in P_N$ and $p_k \in P_A$. Thus, if $\mu_{S0}(p_i)$ gives a bounded, live and reversible marking for net S , then $\mu_{N0}(p_j)$ gives a bounded, live and reversible marking for net N .

3.3. Timing Analysis of the MIMO Structure

In the sequel, we will discuss timing issues with respect to two different operations that may use the same set of resources. Restricting the number of resources represented within an individual MIMO subnet to two, allows the designer to apply widely studied resource deadlock avoidance strategies from the literature in the construction of a two input, two output subnet⁴. It is to be noted that the construction of a equivalent SISO structure and the following analysis, can be done recursively to more than two WDBs or WDBs that share more than two resources.

Before proceeding further, let us define the following terms that are used later in the sequel:

Definition 10: Let τ_a and τ_b be the time duration of the subnets X_A and X_B respectively.

Definition 11: Let τ_{ai} and τ_{ai}' $1 \leq i \leq k$ be the time duration for the acquiring and releasing of resource associated with place p_{ri} by subnet X_A , respectively.

Definition 12: Let τ_{ak} and τ_{af} be the time duration for the acquiring the last required resource and releasing the last resource used with the subnet X_A , respectively.

Definition 13: t_c^d : This denotes the absolute time for the various cases. In our analysis, $1 \leq c, d \leq 4$, where c stands for the subscripts of the times shown in Figure 5 and d denotes the various cases. Thus, t_2^4 , denotes the time t_2 in Figure 5–4.

Definition 14: t_{ki} , $k \in (a, b)$ and $1 \leq i \leq k$: This denotes the requesting time of resource p_{ri} by subnet X_A or X_B .

Definition 15: t_{ki}' , $k \in (a, b)$ and $1 \leq i \leq k$: This denotes the releasing time of resource p_{ri} by subnet X_A or X_B .

It can be noticed that:

- τ_{af} need not necessarily be τ_{ak}' .
- $\forall 1 \leq i \leq k; \tau_{ai} < \tau_{ai}'$.
- $\tau_{af} = \max \{\tau_{ai}'\}; 1 \leq i \leq k$.

Given this background, consider the timing diagram in Figure 5 for the H-EPN subnets considered in Figures 3 and 4.

There exist four general cases that may be considered to establish an upper bound (UB) on the time constraint for the duration of subnets at a higher level. These are:

Case 1: There is no overlap between the operating time durations of subnets X_A and X_B : This means the operation pertaining to subnet, X_A , is completed before the operation

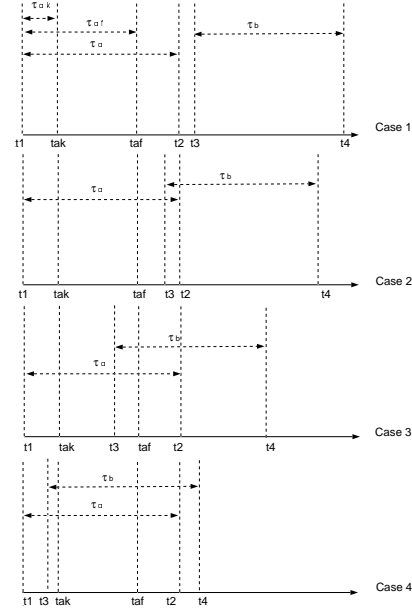


Figure 5. MIMO Subnet Timing Analysis

associated with subnet, X_B , is initiated. This is illustrated in Figure 5–1. This is also visually noticeable in Figure 4 for net, N . This means that all the resources associated with the subnet, X , are available before the initiation of operations of subnets, X_A or X_B .

Case 2: There is an overlap between the operating time durations of subnets X_A and X_B , but subnet X_B requests the use of its first resource, p_{b1} , only after all the resources gained by subnet X_A have been released: This case is illustrated in Figure 5–2. There does not exist any problem for the operation of the two subnets in this case since there is no possibility of the subnets being actually tied up waiting for some resource.

Case 3: There is an overlap between the operating time durations of subnets X_A and X_B , and subnet X_B requests the use of its first resource p_{b1} before all the resources gained by subnet X_A have been released: This situation is illustrated in Figure 5–3. In such a situation, there could exist a delay in the operations associated with subnet X_B . The largest delay occurs when the last released resource of subnet X_A is the first requested resource of subnet X_B .

Case 4: There is an overlap between the operating time durations of subnets X_A and X_B , and subnet X_B requests the use of its first resource p_{b1} before all the resources used by subnet X_A have been requested: This situation is illustrated in Figure 5–4. The largest delay occurs when the last requested resource of subnet X_A is the first requested resource of subnet X_B .

Theorem 4: When N is live and deadlock free, then

$$t_4^3 \leq t_4^2 \text{ and } t_4^4 \leq t_4^2 \quad (8).$$

⁴ Note that this does not mean that the number of resources utilized by a multi-subnet hierarchy needs to be restricted to two. It allows the definition of a resource hierarchy specification where resources may be abstracted in a subsequent subnet structure.

Proof: We will assume that all the resources $p_{r1}, p_{r2}, p_{r3}, \dots, p_{rj}, \dots, p_{rk}$ are gained and released by the two subnets. Consider that $p_{a1}, p_{a2}, p_{a3}, \dots, p_{aj}, \dots, p_{al}, j, l \leq k$, is the order of

resources acquired for subnet X_A and $p_{b1}, p_{b2}, p_{b3}, \dots, p_{bj}, \dots, p_{bl}, j, l \leq k$, is the order of resources acquired for subnet X_B . The least time value of t_4^2 is:

$$(t_4^2)_{min} = t_1 + \tau_a + \tau_b - (\tau_a - \tau_{af}) \quad (9).$$

Figure 5 illustrates the various cases involved and the proof of the theorem consists of establishing the bounds for all these cases. The proof is not derived here due to space limitations and the interested reader is referred to [5]. ■

4. Example

In this section, we will demonstrate the use of activator arcs in constructing and analyzing MIMO nets in an example assembly process. The example is illustrated in Figure 6. It consists of three workstations A, B, and C, and two robots, RA and RB. An input job is first processed by A and subsequently processed by either B or C. The robot RA is used in the operations of A and is also used to transfer intermediate jobs from the output buffer of A to the input buffer of B and C. The robot RB is used exclusively by either B or C. A and B have the capability to schedule jobs based on certain priority constraints, and thus have the ability to maintain priority queues of incoming jobs. All stations are served by an active resource R1. R1 is failure prone and has an active standby, R2, used during dynamic failure situations.

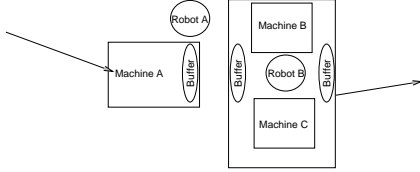


Figure 6. An Example Assembly Line

The system model is derived following the hybrid design methodology described in section three for the construction and analysis of MIMO nets. Given the general system description, first, a high-level H-EPN model is created using a top-down approach. Later, depending on individual machines capabilities, the operations that are carried out on the various machines are modeled through a bottom up synthesis technique. Solid tokens in the various subnets represent resources comprising the system description while dotted tokens represent tokens generated during system operations. The *ss* place regulates the input job flow, represented as an input job in Figure 7. This is modeled as illustrated in Figure 1c. The top level H-EPN model of Figure 7 is used in conjunction with the *ss* place.

4.1. Top-down System Decomposition

Figure 7 provides the top level H-EPN model of the system operations. The model traces the flow of an input job through the system and gives a high level description of the system operations. While Figure 7 represents workstation C with a single token, workstations A and B are represented by the numbers $n1$ and $n2$ for indicating the maximum number of jobs that the workstations can maintain in their individual job queues. These jobs are then scheduled by a priority scheduling algorithm that can handle dynamic job priorities. The subnets SA, SB and SC denote the operations of A, B, and C, respectively.

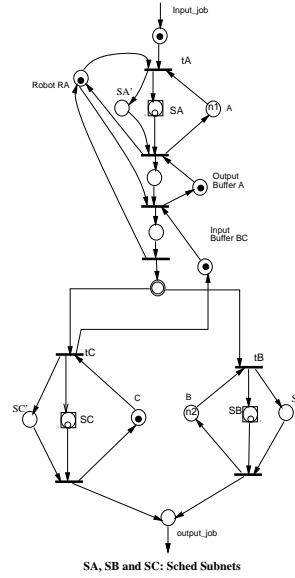


Figure 7. Top Level H-EPN System Model

The above description of system operations as modeled in Figure 7 is obtained by the general system description. Although the presence of conjugate places is shown explicitly in the diagram, the modified SPNP package handles the creation of these conjugate places by means of the *highsub* function, discussed in detail in section IV-D.

4.2. Bottom-up Synthesis of System Operations

Figures 8a and 8b present the subnets for the priority scheduling of jobs and the dynamic failure recognition and rescheduling of jobs. These two subnets form the lowest level description of system operations and model the actual job processing. Note that these two subnets are WDBs⁵ that share common places, *process job-R1* and *process job-R2*, respectively. These subnets are called interacting subnets. The actual subnet for processing an input job for the machine is illustrated in Figure 8c. The QR sets associated with the transitions $tR1$ and $tR2$ schedule incoming jobs to be processed. In case an overflow is detected for the priority job queues in workstations A and B, *Queue Overflow* blocks the acceptance of new jobs at transitions $tR1$ and $tR2$. The *Switch* place initiates the use of resource R2 in case of a failure detected in R1, and re-initiates the use of R1 once it is repaired.

Figure 9a illustrates the subnet model for actual job processing by workstations B and C. Robot RB, used for loading and unloading of intermediate parts onto workstations B and C, is represented here. This H-EPN net is a MIMO subnet. The operation of workstation A is also similar to the operations of B, except it does not use RB in its operations. This is illustrated in Figure 9b. Figures 9a and 9b are combined to give the equivalent SISO net. The synthesis and verification⁶ of the MIMO net is illustrated by Figure 10a. The equivalent SISO net is illustrated in Figure 10b.

⁵ WDBs are SISO nets that are live, safe and reversible.

⁶ The analysis follows from definition 5 in section 3.

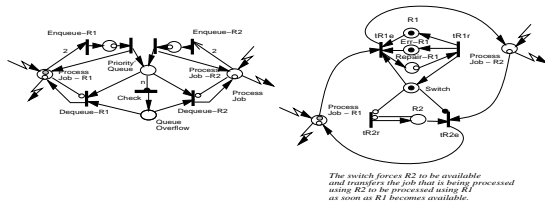


Figure 8a: Priority Scheduling of Jobs

Figure 8b: Dynamic Failure Recognition and Rescheduling of Jobs

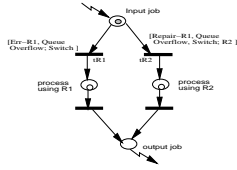


Figure 8c: Subnet Process Job

Figure 8. Actual Job Processing, Priority Scheduling and Dynamic Failure Recognition

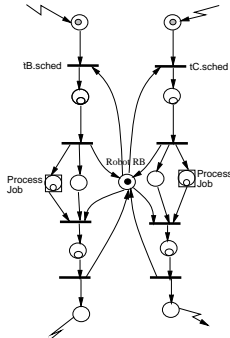


Figure 9a: Operations of Machines B & C

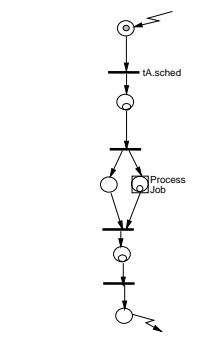


Figure 9b: Operations of Machine A

Figure 9. Subnets for Individual Machine Operations

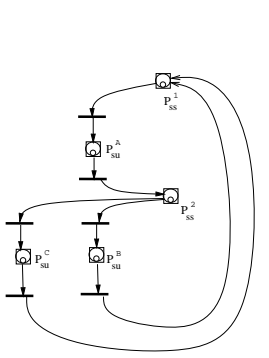


Figure 10a: MIMO Net Construction and Verification

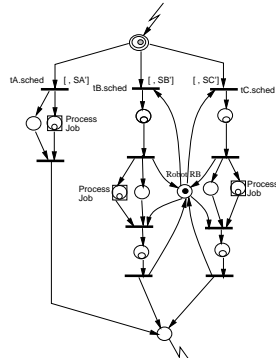


Figure 10b: Equivalent MIMO Net (SA, SB, SC)

Figure 10. Equivalent SISO Subnet for Machine Operations

4.3. Final System Model Integration

The final H-EPN system model is generated by integrating the equivalent SISO net generated in Figure 10 with the top-level decomposition of the system model generated in Figure 7. This is achieved by associating QR sets with the transitions that are output to the input place of the SISO subnet obtained in Figure 10. These QR sets indicate the actual flow of the input job through the assembly.

4.4. System Simulation

The SPNP package [1] has been used for the simulation of the above system and the important system properties of boundedness, liveness and reversibility have been verified. The performance of the system has been studied with and without the effect of dynamic failures. To execute the H-EPN system model using the SPNP package we introduce three extensions to the net descriptions that can be provided by SPNP, (i) Integration of activator arcs, (ii) Integration of QR sets, and, (iii) Integration of MIMO subnet descriptions. The guard (ψ), watchdog (ω) and random (ϕ) functions defined by the H-EPN system model in equation (1) are simulated by means of the *guard*⁷, *reward_type* and *ratefun / ratedep* functions in SPNP.

To use SPNP for simulating the H-EPN model, the activator arcs have to be transformed to SPNP specifications without affecting the expected system behavior. This is achieved defining the functions *aarc* and *maarc*. *aarc* represents the activator arc definition with arc weight 1 and *maarc* represents the definition of an activator arc with an arc multiplicity, m . These functions are defined in terms of the SPNP primitives, *iarc*, *oarc* and *priority*, used to represent input arcs, output arcs, and transition priority, respectively. The syntax for the *aarc* and *maarc* functions are: *aarc*("transition_name", "place_name", *prio*) and *maarc*("transition_name", "place_name", m , *prio*). The *priority* function establishes a default priority of *prio* instead of the default SPNP transition priority value of 0. The user has to ensure that *prio* is larger than any other priority value that is assigned for other transitions, so that H-EPNs are properly simulated by SPNP. The conversion of activator arc definitions to SPNP (with notion of transition priorities) and the classical PN definition (with no notion of transition priorities) is illustrated in Figure 11.

The QR set definitions are dependent upon the packages ability to specify activator and inhibitor arcs for a given transition. Therefore, the function *qrset* is defined. It transforms the QR set definitions into corresponding activator and inhibitor arc definitions. The *qrset* function has the following syntax. *qrset*("transition_name", "inhibitor_place_set", "activator_place_set").

Subnet description by means of the SPNP package is achieved by invoking a function call to a subnet definition in the PN specification file. However, this definition of a subnet essentially duplicates all the places and transitions in the subnet definition for every such call. However, H-EPN subnets cannot be duplicated if they model shared system resources; if the H-EPN subnet definition contains places that hold control (solid) tokens. This is achieved by the *highsub* routine, which essentially creates a conjugate place and the respective input and output arcs to both the conjugate place and the actual subnet input place. All other subnets will be called using a *lowsub* function call. The *lowsub* routine essentially duplicates the subnet along with the creation of a new conjugate place. Within the *sisofun* routine is the actual description of the low level subnet that can be duplicated. Therefore, the input and output places are not defined as parameters to *lowsub* function, but rather the subnet definition

⁷ The *enabling_type* functions in earlier SPNP versions are called the *guard* in SPNP version 4.0.

is passed as an argument to *sisofun*. However, for *highsub* the actual definition of the subnet exists within the normal net description. Moreover, *i* and *j* are used as seeds for renaming places and transitions and to specify other special subnet characteristics. For example, in Figure 10, *n* can be modified accordingly for workstations A and B for analysis depending on the actual number of jobs (*n1* and *n2* in Figure 7, respectively) that they can process simultaneously.

highsub("subnet_place_name", "input_trans", "output_trans", "in_place", "out_place", *i*)

lowsub("subnet_place_name", *sisofun*, "input_trans", "output_trans", *j*)

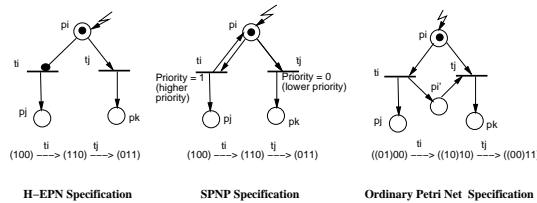


Figure 11. Activator Arc Transformations

5. Conclusions

In this paper, activator arcs have been used as a basis for the construction of QR sets essential in the generation of MIMO nets and the construction of structured PN models. QR sets have been used in deriving a transformation from MIMO nets generated by means bottom up synthesis techniques to SISO nets. The use of the context sensitive

conjugate place during subnet activations is used for reactive decision making in situations where the subnet characteristics are dictated by higher level net initiations. Such characteristics can represent either timing or semantic related behavior.

6. Acknowledgments

The authors are thankful to Prof. Kishor Trivedi for providing the SPNP package.

References

- [1] Ciardo, G., and et. al., "SPNP Users Manual - Version 4.0", March 1994.
- [2] Jeng, M. D., and DiCesare, F., "A Review of Synthesis Techniques for Petri Nets with Applications to Automated Manufacturing Systems", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 1, Jan 1993, pp. 301-312.
- [3] Ramaswamy, S., and Valavanis, K. P., "Modeling, Analysis and Simulation of Failures in a Materials Handling System with Extended Petri Nets", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 9, Sept 1994.
- [4] Ramaswamy, S., and Valavanis, K. P., "Hierarchical Time-Extended Petri Nets(H-EPNs) Based Error Identification and Recovery for Hierarchical Systems", *Accepted to the IEEE Transactions on SMC*, Jan 1995.
- [5] Ramaswamy, S., Valavanis, K. P., and Barber, K. S., "On the Construction and Analysis of Multiple Input Multiple Output Subnets", *Submitted to the IEEE Transactions on SMC*, Nov 1994.
- [6] Venkatesh, K., and et. al., "Augmented Timed Petri Nets for Modeling, Simulation and Analysis of Robotic Systems with Breakdowns", *Journal of Manufacturing Systems*, Vol. 13, No. 4, April 1994, pp. 289-301.