Essentially Decision Free Petri Nets for Real-Time Resource Allocation

B.H. Krogh and R.S. Sreenivas
Department of Electrical and Computer Engineering
and the Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Qualitative properties of Petri nets (such as liveness and safeness) are useful in discrete manufacturing applications since they correspond to consistent, deadlock-free control logic. In this paper we introduce the qualitative property of *essentially decision free* (EDF) places in Petri nets (PNs) to represent the absence of unresolved (nonunique) resource allocation conditions. We show that a previously developed method for PN synthesis of on-line control logic may lead to ambiguous resource allocation conditions which are exhibited in the PN model as non-EDF places. Using the theory of place-invariants, we develop a procedure for identifying non-EDF places in the synthesized logic. The procedure is illustrated for an example of assigning robots to tasks in a flexible assembly cell. For this example, allocation ambiguities are systematically resolved by introducing logical NOT conditions in the PN model.

1 Introduction

This paper concerns the use of Petri nets to synthesize control logic for discrete manufacturing systems. Petri nets (PNs) are attractive for modeling discrete event processes because qualitative PN properties (such as liveness and safeness) correspond to desirable features for the control logic (such as consistency and absence of deadlocks) [1], [2]. In this paper we introduce the qualitative property of essentially decision free (EDF) places in PNs to represent the absence of ambiguities in the firing of enabled transitions. For control of manufacturing systems, the absence of firing ambiguities indicates that the logic represented by the Petri net model corresponds to a realizable control program with deterministic decision rules.

We define the EDF property for general PNs and then develop a method for determining whether a place is EDF in the context of a particular class of PNs previously introduced to model manufacturing control logic [3]. This class of PNs, which we refer to as operation/resource PNs (O/R-nets), result from a decomposition of manufacturing processes into operations and resources leading to two corresponding types of places: operation-places (o-places) and resource-state-places (rs-places). Properties of O/R-nets and their application to manufacturing control have been presented in earlier papers [4], [5]. In this paper we extend the theory of O/R-nets to synthesize control logic which is essentially decision free, in addition to being live and safe.

The following section provides standard PN definitions and notation used throughout the paper and concludes with our definition of EDF places. In section 3 we summarize our procedure for synthesizing live and safe O/R-net models of manufacturing systems and state the relevant results which characterize the p-invariants and reachability sets for these PNs. In section 4 we develop the theory and application of EDF places. We note that a non-EDF rs-place in an O/R-net corresponds to an unresolved resource allocation decision in the control logic. We present a procedure for evaluating whether an rs-place is EDF. This procedure leads to an explicit enumeration of alternatives for eliminating non-EDF places, which we illustrate in section 5 for an example of task allocation in a flexible assembly cell. In the concluding section we discuss directions for future research including the application of EDF analysis to automatically generate on-line control programs.

2 Definitions and Notation

Formally, a Petri Net (PN) can be defined as a triple, (Π,T,Φ) where $\Pi=\{p_1,\dots,p_n\}$ is the set of n *places*, $T=\{t_1,\dots,t_m\}$ is the set of m *transitions*, and $\Phi\subseteq(\Pi \times T)\cup(T \times \Pi)$ is the set of directed arcs, or *branches*, connecting the places and transitions. We assume there are no self-loops (i.e. $\{(p_i,t_j),(t_j,p_i)\}\not\subset\Phi$ for any i,j). The *state* of the PN is defined by the distribution of *tokens* which reside in the places. A *marking* **m** of a PN is an integer-valued column-vector indicating the number of *tokens* in each place, that is $\mathbf{m}=[m_1,\dots,m_n]^T$ where $m_j \geq 0$ is the number of tokens in place p_j. In the graphical representation of PNs (see figure 1), transitions are represented by vertical bars, places are represented by circles, and tokens are represented by dots which reside in the places.

We use the following notation from Hack [6], for $x \in \Pi\cup T$:

$$x^\bullet := \{y \mid (x,y) \in \Phi\} \text{ and } {}^\bullet x := \{y \mid (y,x) \in \Phi\}.$$

This "dot" notation is also applied to denote the predecessor or successor set of a set of places or transitions. For example, if $P \subseteq \Pi$, then $P^\bullet := \{t \in T \mid (p,t) \in \Phi \text{ for some } p \in P\}$

Also, for $P\subseteq\Pi$, we define the *indicator vector* $\mathbf{i}(P)$ as an n-dimensional row vector with components

$$i_j(P) = \begin{cases} 1 & \text{if } p_j \in P \\ 0 & \text{otherwise} \end{cases}$$

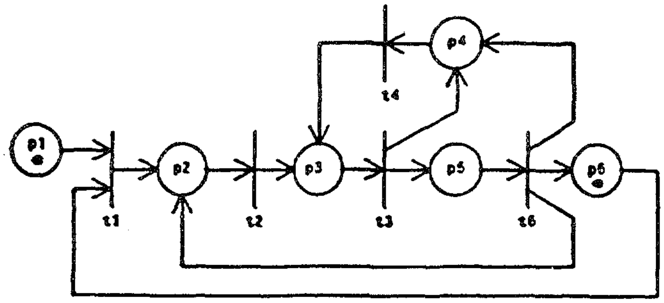


Figure 1: An ordinary marked Petri net.

Conversely, for a n-dimensional binary row-vector $\mathbf{v}$, $\mathbf{i}^{-1}(\mathbf{v}) \subseteq \Pi$, denotes the set of places p_j corresponding to the non-zero components $v_j=1$ of $\mathbf{v}$.

The components $C_{i,j}$ of the nxm *incidence matrix* $\mathbf{C}$, are defined by:

$$C_{i,j} = \begin{cases} -1 & \text{if } (p_i,t_j) \in \Phi \\ 1 & \text{if } (t_j,p_i) \in \Phi \\ 0 & \text{otherwise} \end{cases}$$

A transition is said to be *enabled* when at least one token resides in each of its input places, and an enabled transition *fires* by removing one token from each of its input places and placing one token in each of its output places. Thus, when transition t_i fires, the marking $\mathbf{m}$ is updated to marking $\mathbf{m}'$ according to

$$\mathbf{m}' = \mathbf{m} + \mathbf{C}\mathbf{e}^i \tag{1}$$

where $\mathbf{e}^i$ is the i^{th} unit column vector.

Given an initial marking $\mathbf{m}^o$, the set of *reachable markings* for $\mathbf{m}^o$, $R(\mathbf{m}^o)$, is defined as the set of markings generated by all valid firing sequences starting with marking $\mathbf{m}^o$. A *firing vector* $\mathbf{x}$ is an m-dimensional integer-valued column-vector whose elements indicate the number of times each transition has fired in a valid firing sequence. Hence, a marking $\mathbf{m}$ is in $R(\mathbf{m}^o)$ if there is a valid firing sequence with firing vector $\mathbf{x}$ such that

$$\mathbf{m} = \mathbf{m}^o + \mathbf{C}\mathbf{x} \tag{2}$$

A place p_j is said to be *safe* with respect to marking $\mathbf{m}^o$ if $m_j \leq 1$ for all markings $\mathbf{m}$ in $R(\mathbf{m}^o)$. A PN is safe if all places in the net are safe. Transition t_i is *live* with respect to marking $\mathbf{m}^o$ if from every marking in $R(\mathbf{m}^o)$ there exists a firing sequence which will lead to a marking for which t_i is enabled. A PN is live if all transitions in the net are live.

The properties of liveness and safeness for a PN are related to the concept of *invariants* for the net [7], [8]. An n-dimensional nonzero row-vector $\mathbf{y}$ with binary-valued elements is said to be a *p-invariant* (*place*-invariant) if

$$\mathbf{y}\mathbf{C} = \mathbf{0}. \tag{3}$$

A place p_j is said to be *in* the p-invariant $\mathbf{y}$ if $p_j \in \mathbf{i}^{-1}(\mathbf{y})$. If no proper subset of the places in a p-invariant $\mathbf{y}$ corresponds to another p-invariant for the PN, then $\mathbf{y}$ is called a *proper* p-invariant. Multiplying equation (2) by a p-invariant $\mathbf{y}$ and applying equation (3) gives

$$\mathbf{y}\mathbf{m} \equiv \mathbf{y}\mathbf{m}^o, \tag{4}$$

which implies *the sum of the tokens in the places in a p-invariant is constant for all markings reachable from* $\mathbf{m}^o$. Thus, if $\mathbf{y}\mathbf{m}^o=0$ or 1, all places in $\mathbf{y}$ are safe with respect to $\mathbf{m}^o$.

We now define essentially decision free (EDF) places as a qualitative property of places in PNs. As with liveness of transitions and safeness of places, the EDF property is defined with respect to an initial marking.

Definition. A place $p_j \in \Pi$, is said to be essentially decision free (EDF) with respect to an initial marking $\mathbf{m}^o$ if for every $\mathbf{m}$ in $R(\mathbf{m}^o)$ there is at most one transition enabled in $p_j^\bullet$.

Our definition of EDF places generalizes the notion of *decision-free* PNs in which each place has exactly one output transition [9]. An EDF place can have several output transitions, all of which are live with respect to $\mathbf{m}^o$. The EDF property states that no two of these transitions are ever enabled simultaneously. In other words, when a place is EDF the Petri Net firing logic is such that there is never any ambiguity about which of its output transitions will fire. In section 4 we develop the meaning of EDF and non-EDF rs-places for the class of PN models of manufacturing systems defined in the following section.

3 Operation/Resource Petri Nets

We consider applications such as discrete parts manufacture and automated assembly for which the overall manufacturing process can be decomposed into a set of distinct operations. By an *operation* we mean a single stage of the process which is executed independently from the other parts of the system. Operations can be performed concurrently, but precedence relations among the operations can impose a partial ordering on the operation sequence. Each operation requires a set of *resources*, such as raw materials, fixtures, and robots. A set of *discrete resource states* is associated with each resource representing the discrete conditions required to execute each operation. These conditions include precedence relations, availability of resources, error recovery cycles and other state-dependent sequencing decisions.

To model manufacturing systems, we define a class of PNs with two types of places, *operation places* (*o-places*) and *resource state places* (*rs-places*), corresponding to the operations and discrete resource states defined above. A token present in an o-place indicates the operation is in progress; a token in an rs-place indicates the resource is in the corresponding discrete resource state. Graphically, we represent o-places as boxes and rs-places as circles (see figure 2). The rs-places and o-places alternate along any directed path in the PN and each transition is connected to a single o-place.

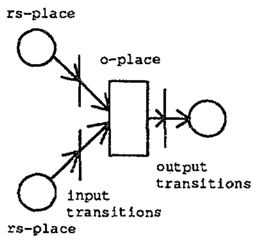


Figure 2: Graphical representation of rs-places and o-places in O/R-nets

To build a PN model of a manufacturing system, the discrete resource states are mapped into rs-places and operations are mapped into o-places. To indicate the resource state requirements for executing an operation, the appropriate rs-places are assigned as input places to an input transition for the o-place. There may be more than one set of conditions under which an operation will be executed, as, for example, when an operation is re-executed in an error recovery cycle. To represent such logical OR conditions for the execution of an operation, multiple input transitions are mapped into the o-place. The rs-places corresponding to the resource states resulting from an operation are connected to an output transition for the o-place. Multiple output transitions can be associated with an o-place to indicate the different sets of discrete resource states which can result from the execution of the operation. This admits logical OR relations for modeling operations which have conditional results, as, for example, when an operation may not be successful due to mechanical failure.

We have developed a method for *synthesizing* PN models which guarantees there are no logical inconsistencies or extraneous conditions in the control logic [3], [5]. The PN model for the complete system is synthesized by joining activity cycles for each resource along common paths determined by the AND and OR conditions for the execution of the various operations. The synthesis procedure constitutes a design of the system control logic which is consistent with the operation sequencing requirements. The synthesis procedure leads to a particular class of PNs which we call Operation/Resource nets (O/R-nets). As an O/R-net is constructed from simple resource activity cycles, the synthesis procedure gives explicit rules for constructing the set of proper p-invariants for the net.

The fundamental theoretical results for O/R-nets are summarized as follows. Given the set $\mathcal{Y} = \{\mathbf{y}_1,...,\mathbf{y}_k\}$ of k proper p-invariants for an O/R-net, let $\mathcal{R}$ be the set of n-dimensional binary column-vectors defined as:

$$\mathcal{R} = \{\, \mathbf{m} \mid \mathbf{y}\mathbf{m} = 1 \;\; \forall\, \mathbf{y} \in \mathcal{Y} \,\}$$

Then, for any $\mathbf{m}^{\mathrm{o}} \in \mathcal{R}$, the O/R-net is live and safe with respect to the marking $\mathbf{m}^{\mathrm{o}}$, and $\mathrm{R}(\mathbf{m}^{\mathrm{o}}) \subseteq \mathcal{R}$. We refer to $\mathcal{R}$ as the *reachability class* for the O/R-net.

4 EDF Places in O/R-nets

In this section we consider the significance of EDF places in the context of O/R-net models of discrete manufacturing systems. As defined in section 2, a place $\mathrm{p} \in \Pi$, is EDF with respect to an initial marking $\mathbf{m}^{\mathrm{o}}$ when at most one output transition in $\mathrm{p}_j^{\bullet}$ is enabled by any $\mathbf{m} \in \mathrm{R}(\mathbf{m}^{\mathrm{o}})$. A non-EDF o-place in an O/R-net represents a situation where different sets of resource states can result from an operation, depending on the outcome of the underlying physical process. When the O/R-net is used for on-line control, the particular outcome of an operation is ascertained by the controller from sensing data, and the internal representation of the system state is updated accordingly. When the O/R-net is used as a simulation model, the outcome of an operation may be generated randomly or it may be evaluated from an explicit simulation model of the process. In either case, one would expect the O/R-net model to have non-EDF o-places.

The situation is different, however, when one considers the EDF property for rs-places. When an rs-place is EDF it indicates that the control logic allocates the resource associated with the rs-place to a unique operation for all possible states (markings) of the system. Note that this does not mean the operation sequencing is necessarily deterministic. A resource in a particular state (rs-place) may be allocated to one operation or another depending on the states of other resources in the system, which in turn could differ according to the timing and precedence relations among concurrent operations. If, however, the states of the other resources eventually determine a unique operation assignment, then the control logic is deterministic and the corresponding rs-place is EDF.

When an rs-place is non-EDF, that indicates there are conditions under which the allocation of a resource is not uniquely determined by the control logic. Such an unresolved resource allocation decision may be critical for the system performance, or it may be a point where there is no clear preference for choosing a particular operation for a resource. For the purposes of performance analysis, there may be instances where it is appropriate to randomly select the next operation. Such ambiguities in the control rules must eventually be resolved, however, by the on-line control program. It is for this purpose, to eliminate unresolved resource allocation decisions in on-line control, that we have introduced the EDF concept.

We are interested in developing systematic procedures for:

1. determining whether or not an rs-place is EDF; and

2. if an rs-place is not EDF, eliminating the ambiguity in the control logic.

The remainder of this section deals with the first issue in the context of O/R-nets. We illustrate one approach to the second issue by an example in the following section. Our method for identifying the set of rs-places which are not-EDF is based on the p-invariant characterization of the set reachable markings $\mathcal{R}$ defined in the previous section. We state the following result [10].

Theorem. Given an O/R-net (Π,T,Φ) with set $\mathcal{Y}$ of proper p-invariants and reachability class $\mathcal{R}$, a place $p \in \Pi$ is EDF with respect to every marking $\mathbf{m} \in \mathcal{R}$ if and only if for every pair of distinct transitions (t',t'') in $p^\bullet$ there exists at least one proper p-invariant $\mathbf{y} \in \mathcal{Y}$ such that

$$\mathbf{i}({}^\bullet t' \cup {}^\bullet t'')\mathbf{y} \geq 2\,. \qquad (5)$$

Proof. Necessity: Suppose p is an EDF place and there is a pair of distinct transitions (t',t'') in $p^\bullet$ such that $\mathbf{i}({}^\bullet t' \cup {}^\bullet t'')\mathbf{y} < 2$ for every $\mathbf{y} \in \mathcal{Y}$. Then one can construct a marking $\mathbf{m}'$ such that $\mathbf{i}^{-1}(\mathbf{m}') \subseteq {}^\bullet t' \cup {}^\bullet t''$, and $\mathbf{y}\mathbf{m}' = 1$ for every $\mathbf{y} \in \mathcal{Y}$. Hence, $\mathbf{m}'$ is a valid marking which enables both t' and t''. But this contradicts the assumption that place p is EDF with respect to every $\mathbf{m} \in \mathcal{R}$.

Sufficiency: Suppose that for a given place p and each pair of distinct transitions in $p^\bullet$ there exists some $\mathbf{y} \in \mathcal{Y}$ such that inequality (5) is true. Moreover, suppose that for some $\mathbf{m}^o \in \mathcal{R}$ there exists a marking $\mathbf{m} \in R(\mathbf{m}^o)$ which enables two distinct transitions (t',t'') in $\mathbf{p}^\bullet$. This implies $\mathbf{m} - \mathbf{i}({}^\bullet t' \cup {}^\bullet t'') \geq 0$. Since $\mathbf{my} = 1$ for every $\mathbf{y} \in \mathcal{Y}$, we conclude that $\mathbf{i}({}^\bullet t' \cup {}^\bullet t'')\mathbf{y} \leq 1$ for every $\mathbf{y} \in \mathcal{Y}$, which contradicts inequality (5). QED

This theorem leads to the following test for determining whether an rs-place in an O/R-net is EDF. Let p be an rs-place with L output transitions. If L = 1, p is trivially EDF. When L is greater than one, let $\mathbf{i}_j$, j= 1,...,J be the indicator vectors corresponding to the $J = \binom{L}{2}$ distinct pairs of transitions in $p^\bullet$. Then the theorem implies p is not EDF with respect to every $\mathbf{m} \in \mathcal{R}$ if and only if for some $j \in \{1,...,J\}$

$$\mathbf{i}_j \mathbf{Y} \leq \mathbf{1} \qquad (6)$$

where $\mathbf{Y}$ is the nxk matrix with the elements of $\mathcal{Y}$ as columns, $\mathbf{1}$ is an n-dimensional row vector of ones, and the vector inequality holds componentwise.

Applying this test to each rs-place in an O/R-net, the set of non-EDF places can be identified. Moreover, for each non-EDF place, the pairs of output transitions which can be simultaneously enabled correspond to the indicator vectors $\mathbf{i}_j$ for which inequality (6) holds. In the following section we illustrate how this information can be used to eliminate the unresolved resource allocation decisions represented by the non-EDF rs-places in an O/R-net.

5 Example: Resource Allocation for Flexible Assembly

As an example, we consider task assignment in a two-robot assembly cell, illustrated in figure 3, for a simple task involving three operations: the assembly of a subassembly A (SA) and a subassembly B (SB), which in turn are assembled to form the final assembly C. To achieve this we have two robots: robot 1 (R1), and robot 2 (R2), each of which can perform all three assembly tasks. The subassembly tasks must be performed according to the precedence relations shown in figure 4.

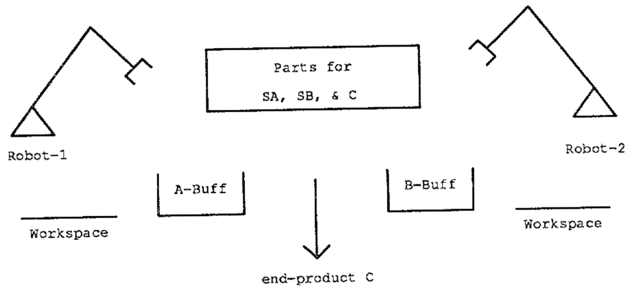


Figure 3: Flexible assembly cell example

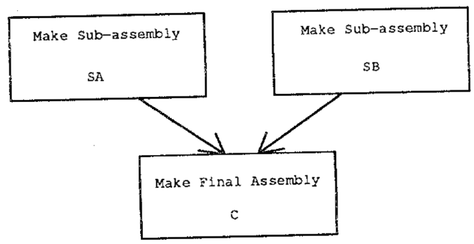


Figure 4: Precedence relationships for assembly example

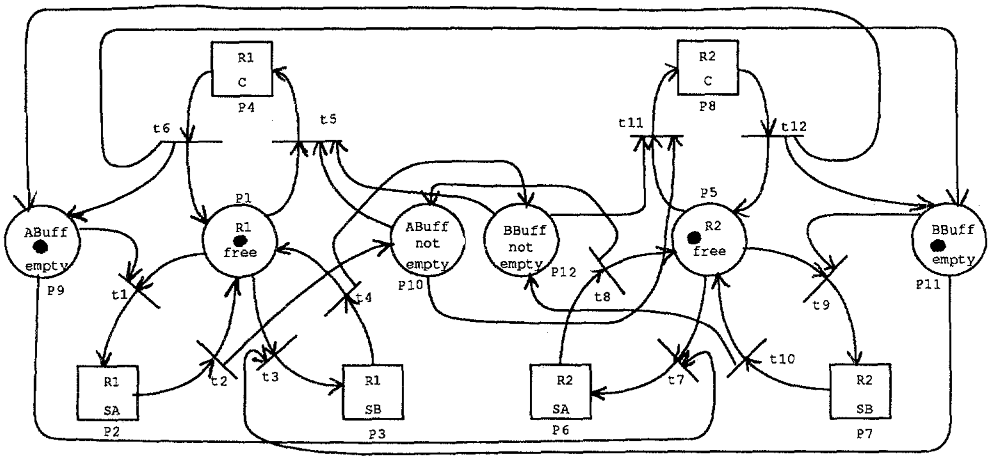
We first consider the following rules for assigning the robots to the assembly operations:

- if the robot is free and the A buffer is empty, do SA
- if the robot is free the B buffer is empty, do SB
- if the robot is free, A buffer is not empty, and B buffer is not empty, do the final assembly C

Note that these rules are are ambiguous when both buffers are empty.

The O/R-net representing these resource allocation rules, shown in figure 5, was constructed using the procedure from reference [3], which was briefly outlined in section 3. This O/R-net has places which are non-EDF with respect to the initial marking shown in figure 5, which corresponds to the ambiguous condition noted above. For the example, both the output transitions for place p_9 are enabled the given initial marking.

In the following section we apply the procedure outlined in the previous section for identifying the non-EDF places. We then illustrate a systematic procedure for resolving the ambiguous resource allocation conditions by using *"NOT"* branches [7]. A *"NOT"* branch is a directed arc from a place to a transition which inhibits the firing of the transition when there is a token in the place. We represent *"NOT"* branches by bold arcs in the graphical representation of the O/R-net.



P-invariants: {P1, P2, P3, P4}, {P5, P6, P7, P8}, {P2, P4, P6, P8, P9, P10},

and {P3, P4, P7, P8, P11, P12}

Figure 5: O/R-net for assembly example and associated p-invariants

5.1 Identifying non-EDF rs-places

The procedure for identifying a non-EDF place is as follows:

- Identify the set of rs-places A with more than one output transition.
- Enumerate all transitions $p_j^\bullet$ for each $p_j \in A$
- For each $p_j \in A$, if there is a pair of transitions $t_k, t_l \in p_j^\bullet$ such that $\mathbf{i}(\{{}^\bullet t_k \cup {}^\bullet t_l\})\mathbf{Y} \leq \mathbf{1}$ (equation (6)), then p_j is non-EDF.

For the example under consideration, the set A is:

$$A = \{p_1, p_5, p_9, p_{10}, p_{11}, p_{12}\} \tag{7}$$

and,

$$p_1^\bullet = \{t_1, t_3, t_5\} \tag{8}$$
$$p_5^\bullet = \{t_7, t_9, t_{11}\}$$
$$p_9^\bullet = \{t_1, t_7\}$$
$$p_{10}^\bullet = \{t_5, t_{11}\}$$
$$p_{11}^\bullet = \{t_3, t_9\}$$
$$p_{12}^\bullet = \{t_5, t_{11}\}$$

Applying the test in inequality (6) we find that every place in A is non-EDF. Table 1 shows the pair of transitions for each place in A which can be simultaneously enabled by some marking in $\mathcal{R}$.

5.2 Resolving Conflicts via "NOT" branches

For this example, we illustrate a procedure of enumerating all methods of resolving the conflict situations in the net by using *"NOT"* branches.

First, we identify the subsets of places in A which cannot contain tokens simultaneously. In general, these sets are given by $\mathbf{i}^{-1}(\mathbf{y}) \cap A$ for $\mathbf{y} \in \mathcal{Y}$, the set of proper p-invariants. For the assembly system example, there are four such sets, namely:

$$A_1 = \{ p_1 \}$$
$$A_2 = \{ p_5 \}$$
$$A_3 = \{ p_9, p_{10} \}$$
$$A_4 = \{ p_{11}, p_{12} \}$$

We note that for this example any marking with one token in each of the four sets above is a valid marking for the O/R-net since there are four proper p-invariants. Moreover, there are four markings with exactly one token in each A_j, j=1, . . . ,4, and these are the only valid markings for which multiple transitions are enabled for places in A (cf. Table 1). We consider each of these markings in turn.

η	$(P_j)^\bullet$	t_k, t_l
P_1	$\{t_1, t_3, t_5\}$	t_1, t_3
P_5	$\{t_7, t_9, t_{11}\}$	t_7, t_9
P_9	$\{t_1, t_7\}$	t_1, t_7
P_{10}	$\{t_5, t_{11}\}$	t_5, t_{11}
P_{11}	$\{t_3, t_9\}$	t_3, t_9
P_{12}	$\{t_5, t_{11}\}$	t_5, t_{11}

Table 1: Non-EDF places and simulataneously enabled output transitions for the assembly system example

$\mathbf{m}_1 = \mathrm{i}^{-1}(p_1, p_5, p_9, p_{11})$: This happens to be initial marking shown in figure 5 for which all four pairs of transitions listed in table 1 are enabled for the four places p_1, p_5, p_9, and p_{11}. To eliminate the ambiguity in the control logic for this marking while retaining the liveness of the O/R-net, we use "NOT" branches to "disable" transitions t_3 and t_7, while leaving the other two transitions, t_1 and t_9, enabled. This is accomplished with two "NOT" branches: p_5 to t_3 and p_1 to t_7.

$\mathbf{m}_2 = \mathrm{i}^{-1}(p_1, p_5, p_9, p_{12})$: For this marking the two output transitions from p_9 are simultaneously enabled. This marking corresponds to the situation where SB has been completed and it is not specified which of the robots should do SA. However, the "NOT" branch from p_1 to t_7, introduced for $\mathbf{m}_1$ disables t_7 for $\mathbf{m}_2$, thus eliminating the ambiguity.

$\mathbf{m}_3 = \mathrm{i}^{-1}(p_1, p_5, p_{10}, p_{11})$: This marking is similar to $\mathbf{m}_2$ in that it represents the situation when SA is completed and one of the robots must be assigned to SB. The ambiguity arises from the simultaneous enabling of both output transitions for p_{11}. This ambiguity is resolved by the "NOT" branch from p_5 to t_3 which was introduced for $\mathbf{m}_1$.

$\mathbf{m}_4 = \mathrm{i}^{-1}(p_1, p_5, p_{10}, p_{12})$: For this marking t_5 and t_{11} are both enabled, both of which are output transitions for p_{10} and p_{12}. This marking corresponds to the situation where both SA and SB have been completed and one of the robots must be assigned to do the final assembly C. We eliminate the ambiguity in the control logic by introducing a "NOT" branch from p_1 to t_{11}. This essentially assigns robot R1 to do the final assembly.

The resulting O/R-net with the "NOT" branches is shown in figure 6. We note that this represents just one of several ways in which the non-EDF places could be eliminated.

6 Conclusions

This paper introduces the qualitative property of essentially decision free places in PNs and illustrates the application of this concept for analyzing the presence of ambiguities in the control logic for discrete systems. A procedure has been developed for identifying the set of non-EDF places in a class of PNs which represent operation sequencing logic in manufacturing applications. The example illustrates how unresolved resource allocation conditions in flexible manufacturing systems can be identified and eliminated by introducing NOT conditions in the PN model.

There are several directions for research in both the theory and application of the concepts introduced in this paper. Theoretical issues currently being investigated include:

- relationships between EDF places and transition invariants for the PN;
- conditions guaranteeing liveness of PNs with NOT branches;
- integration of tests for EDF places with the O/R-net synthesis procedure;
- EDF places in timed PNs.

One potential application of the EDF concept is in the area of automatic verification of on-line control logic. Given a high-level specification of the operation sequencing logic for a flexible manufacturing system, tests for EDF places could be used to identify unresolved conditions in the specification. This diagnostic information could provide options from which the system designer could choose to resolve the ambiguities in the logic, or the computer could automatically resolve the conditions. This type of automatic analysis and diagnosis of qualitative PN properties could play an important role in the automatic generation of on-line control programs for discrete manufacturing systems [11].

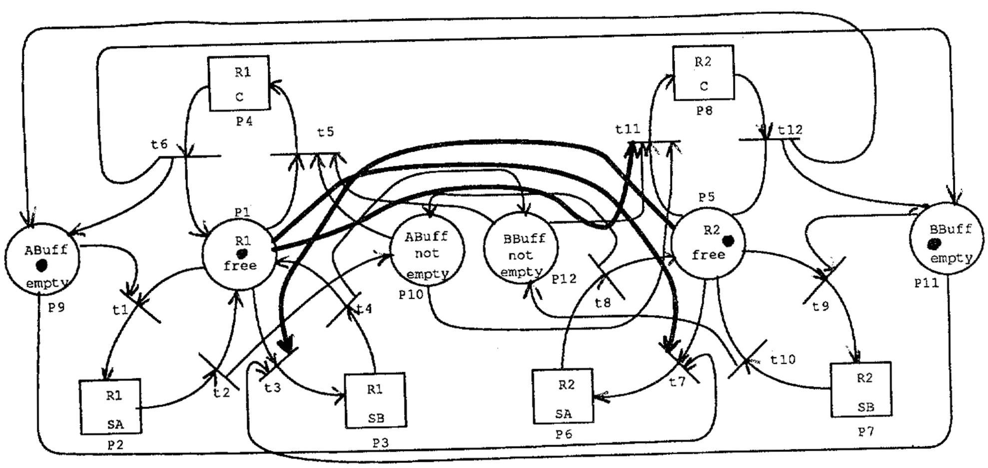


Figure 6: The O/R-net with "NOT" branches

Acknowledgments

This research was supported in part by Xerox Corporation, Philips North America, and the National Science Foundation under research grant DMC-8451493.

References

1. M. Kamath and N. Vishwanadham, "Applications of Petri Net Based Models in the Modeling and Analysis of Flexible Manufacturing Systems", *IEEE International Conf on Robotics and Automation*, Apr 1986, pp. 312-317.

2. J.L. Peterson, *Petri Net Theory and the Modeling of Systems,* Prentice-Hall, Inc., Englewood Clifss, NJ, 1981.

3. C.L. Beck, "Modeling and Simulation of Flexible Control Structures for Automated Manufacturing Systems", Tech. report, Robotics Institute, Carnegie-Mellon University, 1985.

4. C.L. Beck and B.H. Krogh, "Models for Simulation and Discrete Control of Manufacturing Systems", *IEEE International Conf on Robotics and Automation*, Apr 1986, pp. 305-310.

5. B.H. Krogh and C.L. Beck, "Synthesis of Place/Transition Nets for Simulation and Control of Manufacturing Systems", *Proceedings 4th IFAC/IFORS Symposium Large Scale Systems,* Aug 1986.

6. M.H.T. Hack, "Analysis of Production Schemata by Petri nets", Tech. report 94, Project MAC, M.I.T., 1972.

7. T. Agerwala, "Some Applications of Petri Nets", *Proc. '78 National Electronics Conf.*, Vol. 23, October 1978, pp. 149-154.

8. Y. Narahari, and N. Vishwanadham, "A Petri Net Approach to the Modelling and Analysis of Flexible Maunufacturing Systems", *Annals of Operations Research,* Vol. 3,1985, pp. 449-472.

9. K.E. Stecke, and D. Dubois, "Using Petri Nets to Represent Production Processes", *Proc. of the 22nd IEEE Conf. on Decision & Control*, December 1983, pp. 1062-1067.

10. R.S. Sreenivas and B.H. Krogh, "Petri Net Synthesis for Conflict Free Resource Allocation in Discrete State Systems", Tech. report, Robotics Institute, Carnegie Mellon University, March 1987.

11. G. Ekberg and B.H. Krogh, "Software for Automatic Generation of Computer Programs for Manufacturing Control", Tech. report, Robotics Institute, Carnegie-Mellon University, Sept 1986.