HIERARCHICAL SUPERVISORY CONTROL OF COMPLEX PETRI NETS

BY

NISHA SOMNATH

THESIS

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering in the Graduate College of the University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Adviser:

Professor Ramavarapu S. Sreenivas
ABSTRACT

Large, complex systems are prone to the phenomenon of *livelocks*. Once a system enters a livelocked-state, there is at least one activity of the modeled system that cannot be executed from all subsequent states of the system. This phenomenon is common to many operating systems where some process enters into a state of suspended animation for all perpetuity, and the user is left with no other option than to forcibly kill the suspended job, or reboot the machine.

This thesis is about finding *supervisory control policies* that enforce livelock-freedom in large complex systems that are modeled using *Petri nets*. The supervisory policy, when it exists, prevents the occurrence of certain *events* (i.e. activities) at specific states in such a way that the supervised system is livelock free. A hierarchical approach is used to find a supervisory policy for petri nets.

This theory finds application for concurrent systems like computer operating systems which are complex to analyze. The complex system is (recursively) represented as the combination of two smaller systems. Under favorable conditions identified in this thesis, local supervisory policies that enforces livelock-freedom in each of the smaller systems will suffice to enforce livelock-freedom in the larger system.
To my parents, for their love and support.
ACKNOWLEDGMENTS

I would like to express profound gratitude to my advisor Professor Ramavarapu Sreenivas for giving me an opportunity to work on this thesis, for his assistance in the preparation of this manuscript and for all the guidance and support. I am inspired by his attention to detail and his intense commitment to his work. I have thoroughly enjoyed working with him and have learnt a lot. Words alone cannot express my gratitude towards him.

I greatly appreciate the support I received from Professor Lawrence Angrave and the department of Computer science.

I would like to thank my family for their love, support, and especially for their incredible patience.
# TABLE OF CONTENTS

LIST OF FIGURES ......................................................... vi

CHAPTER 1 INTRODUCTION .............................................. 1
  1.1 Properties of Petri Nets ......................................... 2
  1.2 Methods for analyzing petri nets ............................... 4
  1.3 Problem statement ................................................ 11

CHAPTER 2 NOTATIONS AND DEFINITIONS AND SOME PRELIMINARY OBSERVATIONS .............................................. 13
  2.1 Stepwise Refinement and Abstraction of Petri Nets of Suzuki and Murata [1] ........................................... 14
  2.2 Supervisory Control of PNs ....................................... 16

CHAPTER 3 ON SUPERVISORY POLICIES THAT ENFORCE LIVENESS IN A PN OBTAINED BY THE REFINEMENT .............. 18
  3.1 Deciding Properties $P1$, $P3$ and $P3$ in arbitrary PNs ....... 18
  3.2 Main Result ....................................................... 20

CHAPTER 4 CONCLUSIONS .................................................. 30
  4.1 Future work ....................................................... 30

REFERENCES ............................................................. 32
LIST OF FIGURES

1.1 The petri net shown in 1a denotes a trap while the net shown in 1b denotes a siphon. .......................... 4
1.2 A petri net that is not bounded and not live (cf. section V.C, [2]) ...................................................... 6
1.3 Reachability graph for the petri net in Figure 1.2. It can be seen that even though the graph that the transition $t_1$ is never fired ................................................................. 7
1.4 A petri net with infinitely large reachable set of markings [3]. ............................................................ 8
1.5 Finite reachability graph of the petri net with infinitely large reachable set of markings [3] ..................... 9
1.6 An example of a non free choice petri net where transition $t_3$ has two input arcs (cf. section V.C, [2]) . . . 10
1.7 Converted free choice petri net for the net shown in Figure 6 with addition transition $t_5$ and place $p_5$ .......... 10
1.8 A petri net that consists of both a siphon and a trap. In the presence of a token in each siphon will result in a live net [4] ......................................................................................... 10
1.9 A petri net that does not contain a trap and hence is not live (cf. section V.C, [2]). ............................... 11

3.1 A (fully controlled) FCPN that is constructed from the FCPN $N_2(m_2^0)$. ............................................... 19
3.2 (a) An FCPN $N_1(m_1^0)$, that meets requirement $P1$. (b) An FCPN $N_2(m_2^0)$ that meets requirements $P2$ and $P3$. (c) $\hat{N}_2(m_2^0)$, which is also an FCPN, and (d) The FCPN $N_3(m_3^0)$ obtained by the refinement process of section 2.1. ...... 29
CHAPTER 1

INTRODUCTION

*Petri nets* were invented by Carl Adam Petri in 1939 which are used for description of distributed systems that are independent, interaction and have concurrent components. A petri net is a bipartite graph that represents a system that is being modeled. A petri net combines a well defined mathematical model with the graphical representation of the dynamic behavior of the systems. The theoretical aspect of petri nets can be used for analyzing system behavior while the graphical aspect can be used for analyzing system state changes. Hence, petri nets can be used to model various kinds of event driven systems such as computer networks, communication systems, manufacturing plants, real time computing systems and workflow. The different components of a petri net are:

1. **Place** - which is represented by a circle which contain a discrete number of tokens
2. **Tokens** - which are represented by dots
3. **Transitions** - which are represented by rectangles
4. **Arcs** - are lines that show connections between transition and places and are labeled with their weights.

The distribution of tokens in a petri net represent a configuration in a net which is called *marking*. An *input place* is a place from which an arc runs to a transition while an arc runs from a transition to an *output place*. When a transition fires, tokens move from the input place to the output place. There can be several interpretations of transitions and places. A transition can be interpreted as an event and the input and output places of the transition represent the pre-conditions and post-conditions of an event. In another interpretation a transition can be a computation step with input and output places as input and output data respectively or a transition could be a
job with input and output places representing the resources needed and the resources released.

Chapter 2 contains a formal definition of the petri nets. For subsequent discussion, we informally define a petri net structure as a 3-tuple \( N = (\Pi, T, \Phi) \), where \( \Pi (T/\Phi) \) represents the set of places (transition/arcs). The initial distribution of tokens, also called the initial marking, is denoted by \( m^0 \). The petri net \( N(m^0) \) can be thought of as the combination of a petri net structure \( N \) and an associated initial marking \( m^0 \).

1.1 Properties of Petri Nets

A transition is said to be enabled when the input place has at least the same number of tokens as the weight of the arc from the input place to the transition. When the transition fires tokens (as much as the weight of the arc) are removed from the input place and added to the output place. In this thesis we consider petri nets with arcs with weight one. An enabled transition is not required to fire. The firing of the transition depends upon the occurrence of the event that the transition represents. That is, the Petri net model only represents the set of possible event occurrences, but it does not model which one of these possible event occurrences is selected to be executed within the modeled system.

1. Reachability - A marking \( m^n \) is said to be reachable when a series of firing from initial marking \( m^0 \) results in \( m^n \). A set of firing sequence is represented by \( \sigma = t_1t_2 \cdots t_n \), where \( t_j \in T(j \in \{1, 2, \ldots, n\}) \). The set of all possible firing sequence is denoted by \( T^* \). The reachability problem of a petri net is to decide given a petri net \( N = (\Pi, T, \Phi) \) and a marking \( m^n \) whether \( m^n \in \mathcal{R}(N, m^0) \). The reachability problem is decidable however, it cannot be decided if the set of reachable markings of one petri net is contained in that of another. That is, given two petri nets \( N_i(m^0_i) \), where \( N_i = (\Pi_i, T_i, \Phi_i)(i = 1, 2) \), in general it is not possible to decide if \( \mathcal{R}(N_1, m_1^0) \subseteq \mathcal{R}(N_2, m_2^0) \).

2. Boundedness - A petri net is said to be \( k \)-bounded if the number of tokens in any place does not exceed \( k \) for any reachable marking. The petri net in Figure 1.2 is an example of a petri net that is not bounded.
When the transition $t_4$ is fired the tokens in the place $p_1$ keeps increasing unboundedly. Place $p_1$ also receive tokens when transition $t_3$ is fired. Since the number of tokens in $p_1$ can exceed $k$ which is a finite number the petri net is not bounded.

3. **Liveness** - A petri net is said to be *live* if it is possible to fire any transition from any reachable marking. It is not necessary that the transition has to fire immediately. As stated in (cf. section V.C, [2]) a transition in a petri net is said to be

(a) *dead* ($L_0$ live) if $t$ can never be fired in any sequence in $T^*$.

(b) $L_1$ live if $t$ can be fired at least once in any sequence of $T^*$. i.e it is potentially fireable.

(c) $L_2$ live when $t$ can be fired $k$ times given that $k$ is a positive integer in any sequence of $T^*$.

(d) $L_3$ live if $t$ appears infinitely often in any sequence of $T^*$.

(e) $L_4$ live or just live if $t$ is $L_1$ live for every marking $m^j$ in $\mathcal{R}(N, m^0)$.

This notion of liveness will be used in the rest of the thesis.

4. **Siphon** - Siphon is a part of a petri net in which every transition having an output place has an input place. The property of a siphon is that if at some point the siphon becomes token-free it will remain token-free through out. Figure 1.1(b) denotes a siphon. Considering that the net has only one token as seen from the figure once transition $t_1$ is fired the petri net will be drained of all tokens and will remain token free there after.

5. **Trap** - Trap is a part of a petri net in which every transition having an input place has an output place. The property of a trap is that if there are tokens in the net there will always be tokens in it. Figure 1.1(a) represents a trap. If the petri net has tokens they will always remain in the net as transition $t_1$ is an incoming transition which when fired will result in more tokens in the net however, there is no output transition to drain out the tokens from the net. For a petri net which is a siphon and also a trap there needs to be at least one token to keep the net live.
Figure 1.1: The petri net shown in 1a denotes a trap while the net shown in 1b denotes a siphon.

1.2 Methods for analyzing petri nets

1. Controlled petri nets - A petri net in which a subset of transitions can be prevented from firing by a supervisor [4]. The transitions that can be prevented from firing (controllable) is usually represented by a colored rectangle while the transitions that are uncontrollable are represented by a rectangle without any filling. Figure 1.4 is an example of a controlled petri net. The supervisory policy for this net prevents transition \( t_5 \) from being fired in a manner such that all the tokens are not drained out. The petri net would be live since \( t_5 \) can be fired at some point of time when there are enough tokens in the petri net.

2. Reachability/Coverability graph - The graph consisting of all possible markings that can be reached when the transitions in a net are fired. When the transitions from an initial marking \( m^0 \) are fired it gives rise to new markings. From these new markings further markings are reached as transitions are enabled. This leads to a tree structure that could be infinitely large. The procedure listed below can be interpreted as a finite-characterization of this tree structure, which is known as the reachability tree (cf. section 4.2.1, [5]). The vertex set of this tree is
V, and each vertex v ∈ V has an (extended) marking of the petri net, μ(v), associated with it. An extended marking can be thought of as markings where some places can have infinite tokens. The symbol ω is used to represent the presence of infinite tokens. Each edge of this tree has a transition associated with it. The tree is constructed as follows:

1: The root vertex is v₀. V ← {v₀}, and μ(v₀) = m₀.
2: for vi ∈ V do
3:   if μ(vi) is identical to μ(vj) for some vj ∈ V then
4:      vi has no children, and is marked as the duplicate of vj.
5:   end if
6:   if no transition is enabled under the marking μ(vi) then
7:      vi has no children, and is marked as a terminal vertex.
8:   end if
9:   if vi is not a duplicate-vertex then
10:      for t_j that is enabled under μ(vi) do
11:         Create a new vertex v_k. V ← V ∪ {v_k}.
12:         Create a new directed edge starting from vi and ending at v_k.
13:            Label this edge with the transition t_j.
14:            if The number of tokens in p is ω under μ(vi), for some p ∈ Π then
15:               The number of tokens in p is ω under μ(v_k) too.
16:            else
17:               The number of tokens in p under μ(v_k) is what results when t_j is fired under μ(v_k)
18:            end if
19:      end for
20: end if
21: end for
If the duplicate nodes are merged with the parent node in a reachability graph, we get the coverability graph. A petri net is unbounded if and only if there are \( \omega \) symbols in its coverability graph. The coverability graph is finite for any petri net. Figure 1.2 represents a petri net that is not bounded and not live. The reason the net is not bounded is because the number of tokens are not finite for all reachable markings. This can be seen from the coverability graph in Figure 1.3. The net is not live since the transition \( t_1 \) is not fired even once.

The reachable set of markings can be infinitely large. The petri net in Figure 1.4 has an infinite set of marking for the initial marking of \((1, 0, 0, 0)\). \( t_5 \) is a controllable transition. When transition \( t_4 \) is fired places \( p_2 \) and \( p_3 \) can have infinite number of tokens. However Figure 1.5 shows that the reachability graph of this petri net is finite.

3. Free choice petri net - A petri net, where every arc from a place to a transition is either the unique output arc from that place or it is the unique input arc to that transition [6]. A non free choice petri net can be converted to a free choice petri net by adding an extra place and transition to it. An example of non free choice petri net can be seen
Figure 1.3: Reachability graph for the petri net in Figure 1.2. It can be seen that even though the graph that the transition $t_1$ is never fired in Figure 1.6. The transition $t_3$ has two input arcs to it. This net can be converted into a free choice petri net by converting the arc from $p_2$ to $t_3$ into an additional place $p_5$ and an additional transition $t_5$. The converted free choice petri net is shown in Figure 1.7. However, this conversion might not be permitted in all cases for practical reasons.

*Commoner's Liveness Theorem*- Commoner’s live theorem states that a free choice petri net is live if and only if every siphon contains a marked trap at the initial marking. Any free choice net that does not contain a trap will not be live no matter how the transitions are fired. Figure 1.8 is an example of a free choice petri net that is both a siphon and a trap. The sets $\{p_1, p_3\}, \{p_2, p_3\}, \{p_1, p_2, p_3\}$ form both siphons and traps. Consider the set $P = \{p_1, p_3\}$ the input transitions to these places are $^*P = \{t_1, t_2\}$ and the output transitions are $P^* = \{t_1, t_2\}$. Since $^*P \subseteq P^*$ this set forms a siphon, and since $P^* \subseteq ^*P$ this set forms a trap. Similarly the sets $\{p_2, p_3\}, \{p_1, p_2, p_3\}$ are both siphons and traps. Hence in the presence of a token these parts of the petri net at initialization guarantees liveness. However, Figure 1.9 is an example of a petri net where all siphons do not contain a trap. In this petri net the sets $\{p_1, p_2, p_3, p_4\}, \{p_1, p_2, p_4, p_5\}, \{p_1, p_2, p_3, p_4, p_5\}$ are siphons.
4. Right closed set - A set of markings $\Omega$ is right-closed if $\mathbf{m}^1 \in \Omega \Rightarrow \mathbf{m}^2 \in \Omega$ for all $\mathbf{m}^2 \geq \mathbf{m}^1$. That is, if a marking is in the set, then all larger markings are also in the set. Right-closed sets are uniquely defined by its finite set of minimal elements. For controllable petri nets, a supervisory policy that enforces livelock freedom (if it exists) is characterized by an appropriately selected right-closed set [3]. The policy prevents the occurrence of any transition at a marking if its firing will result in a new marking that is not in the right-closed set. For Figure 1.4 the set of minimal elements are $\{(0 \ 0 \ 0 \ 1), (0 \ 0 \ 1 \ 0), (0 \ 1 \ 0 \ 0), (1 \ 0 \ 0 \ 0)\}$. The supervisory policy of this petri net would prevent the firing of the transition $t_5$ at the marking $(0 \ 0 \ 0 \ 1)$. This is because the firing $t_5$ at $(0 \ 0 \ 0 \ 1)$ would result in the marking $(0 \ 0 \ 0 \ 0)$, which is not in the right-closed set defined by the minimal elements $\{(0 \ 0 \ 0 \ 1), (0 \ 0 \ 1 \ 0), (0 \ 1 \ 0 \ 0), (1 \ 0 \ 0 \ 0)\}$.

Figure 1.4: A petri net with infinitely large reachable set of markings [3].

however, they do not contain any traps. Hence at some point this petri net will cease to be live.
Figure 1.5: Finite reachability graph of the petri net with infinitely large reachable set of markings [3].

Figure 1.6: An example of a non free choice petri net where transition $t_3$ has two input arcs (cf. section V.C, [2]).
Figure 1.7: Converted free choice petri net for the net shown in Figure 6 with addition transition $t_5$ and place $p_5$.

Figure 1.8: A petri net that consists of both a siphon and a trap. In the presence of a token in each siphon will result in a live net [4].
1.3 Problem statement

The motivation behind this thesis is the elimination of livelocks in complex systems represented using Petri nets. By analyzing the petri nets and developing a supervisory policy to ensure liveness we can assure that these systems will never reach a livelock. However, the drawback is that for a large system the petri nets becomes too large to handle and analyzing a large petri net tends to be tedious. This thesis uses stepwise refinement of the large system into smaller systems. Under suitable conditions that are specified in this thesis the entire system can be made live by finding local policies of liveness in smaller systems. An example of one such complex concurrent system is an operating system. The hierarchical model proposed in this thesis can be used to prevent operating systems from encountering a livelock. The system can be interpreted as a collection of sub routines. While the petri net for the system is modeled the sub routines will be the refined model of the larger net. By ensuring liveness for the subroutines we could ensure liveness for the entire system. Even though this thesis develops a two level model for control the theory can be applied recursively to model large complex systems. The driving force behind this thesis was to develop supervisory control for large
operating systems by analyzing the smaller subroutines. By using a petri net model for operating systems this can be accomplished as shown in the further sections.
CHAPTER 2

NOTATIONS AND DEFINITIONS AND SOME PRELIMINARY OBSERVATIONS

We use $\mathcal{N}$ ($\mathcal{N}^+$) to denote the set of non-negative (positive) integers. A Petri net structure $N = (\Pi, T, \Phi)$ is an ordered 3-tuple, where $\Pi = \{p_1, \ldots, p_n\}$ is a set of $n$ places, $T = \{t_1, \ldots, t_m\}$ is a collection of $m$ transitions, and $\Phi \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of arcs. The initial marking function (or the initial marking) of a PN structure $N$ is a function $m^0 : \Pi \rightarrow \mathbb{N}$. We will use the term Petri net (PN) to denote a PN structure along with its initial marking $m^0$, and is denoted by the symbol $N(m^0)$. In graphical representation of PNs places (transitions) are represented by circles (boxes), and each member of $\phi \in \Phi$ is denoted by a directed arc. If $\phi = (p, t)$ ($(t, p)$) the arc is directed from $p$ ($t$) to $t$ ($p$). The initial marking is represented by an appropriate integer, $m^0(p)$, within each place $p \in \Pi$.

The marking of a PN $N$, $m^i : \Pi \rightarrow \mathcal{N}$, identifies the number of tokens in each place. For a given marking $m^i$, a transition $t \in T$ is said to be enabled if $\forall p \in (\cdot t)_N, m^i(p) \geq 1$, where $\cdot N := \{y \mid (y, x) \in \Phi, \text{where } N = (\Pi, T, \Phi)\}$. The set of enabled transitions at marking $m^i$ is denoted by the symbol $T_e(N, m^i)$. An enabled transition $t \in T_e(N, m^i)$ can fire, which changes the marking $m^i$ to $m^{i+1}$ according to the equation

$$m^{i+1}(p) = m^i(p) - \text{card}((\cdot t)_N \cap \{p\}) + \text{card}((t^\ast)_N \cap \{p\})$$

where $\cdot_N := \{y \mid (x, y) \in \Phi, \text{where } N = (\Pi, T, \Phi)\}$ and the symbol $\text{card}(\cdot)$ is used to denote the cardinality of the set argument.

A string of transitions $\sigma = t_1 t_2 \cdots t_k$, where $t_j \in T(j \in \{1, 2, \ldots, k\})$ is said to be a valid firing string starting from the marking $m^i$, if, (1) the transition $t_1 \in T_e(N, m^i)$, and (2) for $j \in \{1, 2, \ldots, k - 1\}$ the firing of the transition $t_j$ produces a marking $m^{i+j}$ and $t_{j+1} \in T_e(N, m^{i+j})$ is enabled. If $m^{i+k}$ results from the firing of $\sigma \in T^*$ starting from the initial marking $m^i$, we represent it symbolically as $m^i \rightarrow \sigma \rightarrow m^{i+k}$. Given an initial marking $m^0$ the set
of reachable markings for $m^0$ denoted by $\mathcal{R}(N, m^0)$, is defined as the set of markings generated by all valid firing strings starting with marking $m^0$ in the PN $N$. A PN $N(m^0)$ is said to be live if
\[ \forall t \in T, \forall m^i \in \mathcal{R}(N, m^0), \exists m^j \in \mathcal{R}(N, m^i) \text{ such that } t \in T_e(N, m^j). \]
A transition $t \in T$ is said to be $k$-enabled if $\exists m \in \mathcal{R}(N, m^0)$, such that $\forall p \in \bullet t, m(p) \geq k$.

A PN structure $N = (\Pi, T, \Phi)$ is a Free-Choice if
\[ \forall p \in \Pi, \text{card}((p^\bullet)_N) > 1 \Rightarrow (p^\bullet)_N = \{p\}. \]
In other words, a PN structure is Free-Choice if and only if an arc from a place to a transition is either the unique output arc from that place, or, is the unique input arc to the transition. A PN $N(m^0)$ where $N = (\Pi, T, \Phi)$ is free choice, is said to be a Free-Choice Petri net (FCPN).

There are several abstraction procedures, where a large PN is systematically reduced to a smaller PN, while preserving some relevant property in the process. The reverse procedure, where a small PN is progressively transformed into a large PN, while retaining some property in course of this transformation, is referred to as the process of refinement (cf. section V.C, [2]). We present an overview of the abstraction/refinement results in reference [1], which is stated in the more general context of PNs with weighted-arcs. These results apply equally to PNs with unit weights on arcs that we consider in this thesis.

2.1 Stepwise Refinement and Abstraction of Petri Nets of Suzuki and Murata [1]

Let $t_{in}, t_{out} \in T$ be two distinct transitions in a PN $N(m^0)$, where $N = (\Pi, T, \Phi)$, and $k \in \mathcal{N}^+$ be a positive integer. We construct a PN structure $\hat{N} = (\hat{\Pi}, \hat{T}, \hat{\Phi})$ where $\hat{\Pi} = \Pi \cup \{\pi_0\}$ ($\pi_0 \notin \Pi$), $\hat{T} = T$, and $\hat{\Phi} = \Phi \cup \{(\pi_0, t_{in}), (t_{out}, \pi_0)\}$. The PN structure $\hat{N}$ is initialized with the marking $\hat{m}_k^0$, where
\[
\hat{m}_k^0(p) = \begin{cases} 
m^0(p) & \text{if } p \in \Pi \\
k & \text{if } p = \pi_0. \end{cases}
\]
The PN \( N(m^0) \) is said to be \( k \)-well behaved (\( k \)-WB) with respect to \( t_{in}, t_{out} \in T \) if and only if the following conditions hold –

1. (WB1) \( t_{in} \) is live in \( \widehat{N}(\widehat{m}_k^0) \),

2. (WB2) For any valid firing string \( \sigma_1 \) in \( \widehat{N}(\widehat{m}_k^0) \) such that \( #(\sigma_1, t_{in}) > #(\sigma_1, t_{out}) \), \( \exists \sigma_2 \in (T - \{t_{in}\})^* \) such that \( \sigma_1\sigma_2 \) is a valid firing string in \( \widehat{N}(m_k^0) \) and \( #(\sigma_1\sigma_2, t_{in}) = #(\sigma_1\sigma_2, t_{out}) \), where \( #(\sigma, t) \) denotes the number of occurrences of transition \( t \) in string \( \sigma \).

3. (WB3) For any valid firing string \( \sigma \in T^* \) in \( \widehat{N}(\widehat{m}_k^0) \), \( #(\sigma, t_{in}) \geq #(\sigma, t_{out}) \).

If \( N(m^0) \) is \((k+1)\)-WB with respect to two distinct transitions \( t_{in}, t_{out} \in T \) for some \( k \geq 1 \), then \( N(m^0) \) is also \( k \)-WB with respect to \( t_{in}, t_{out} \in T \) (cf. Property 1, [1]).

In chapter 3 of this thesis we restrict attention to PNs that satisfy the 1-WB property (i.e. \( k \)-WB, for \( k = 1 \)). To simplify the notation for this special case in subsequent text, we use the notation \( \widetilde{m}^0 \) to denote the initial marking \( \widehat{m}_1^0 \) (cf. equation 2.1, when \( k = 1 \)). That is,

\[
\widetilde{m}^0(p) = \begin{cases} 
m^0(p) & \text{if } p \in \Pi \\
1 & \text{if } p = \pi_0. \end{cases}
\]

Consider two PNs \( N_i(m^0) \) \( (i = 1, 2) \), where \( N_i = (\Pi_i, T_i, \Phi_i) \), \( (i = 1, 2) \), where \( \Pi_1 \cap \Pi_2 = T_1 \cap T_2 = \emptyset \), along with a transition \( t_0 \in T_1 \) that is \( k \)-enabled, but not \((k+1)\)-enabled. In addition, the PN \( N_2(m^0) \) is assumed to be \( k \)-WB with respect to two distinct transitions \( t_{in}, t_{out} \in T_2 \) for some \( k \in \mathcal{N}^+ \). The transition \( t_0 \in T_1 \) is refined by the PN structure \( N_2 \) to yield a new structure \( N_3 = (\Pi_3, T_3, \Phi_3) \) as follows

1. \( \Pi_3 = \Pi_1 \cup \Pi_2 \),

2. \( T_3 = (T_1 \cup T_2) - \{t_0\} \), and

3. \( \Phi_3 = \Phi_1 \cup \Phi_2 - (\Pi_1 \times \{t_0\}) - (\{t_0\} \times \Pi_1) \cup (t_{out} \times (t_0\star)_{N_1}) \cup (t_{in} \times (t_0\star)_{N_1}). \)

The structure \( N_3 \) is initialized with the marking \( m_3^0 \), where

\[
m_3^0(p) = \begin{cases} 
m_1^0(p) & \text{if } p \in \Pi_1 \\
m_2^0(p) & \text{if } p \in \Pi_2 \end{cases}
\]
Testing if $t_0 \in T_1$ is $(k + 1)$-enabled in $N_1(m^0_i)$ is decidable (cf. theorem 20, [1]). Additionally, testing if the PN $N_2(m^0_2)$ is k-WB is also decidable (cf. theorem 21 and Corollary 22, [1]). When these preconditions on $N_1(m^0_i)$ and $N_2(m^0_2)$ are satisfied, it can be shown that the liveness of $N_3(m^0_3)$ implies the liveness of $N_1(m^0_i)$ and $\widehat{N}(m^0_2k)$. In addition, if $\forall m^1_1 \in \mathcal{R}(N_1, m^0_1), \exists m^2_1 \in \mathcal{R}(N_1, m^1_1)$, such that $\forall p \in^* t_0, m^2_1(p) \geq k$ (cf. Condition A, [1]), then the liveness of $N_1(m^0_i)$ and $\widehat{N}(m^0_2k)$ implies the liveness of $N_3(m^0_3)$ (cf. Theorem 11, [1]). Testing if $N_1(m^0_i)$ satisfies Condition A is also decidable (cf. theorem 23, [1]).

The next subsection contains relevant results from the theory of supervisory control of PNs.

### 2.2 Supervisory Control of PNs

The paradigm of supervisory control of PNs assumes a subset of controllable transitions, denoted by $T_c \subseteq T$, can be prevented from firing by an external agent called the supervisor. The set of uncontrollable transitions, denoted by $T_u \subseteq T$, is given by $T_u = T - T_c$. The controllable (uncontrollable) transitions are represented as filled (unfilled) boxes in graphical representation of PNs.

A supervisory policy $P : \mathcal{N}^n \times T \rightarrow \{0, 1\}$, is a function that returns a 0 or 1 for each transition and each reachable marking. The supervisory policy $P$ permits the firing of transition $t_j$ at marking $m^i$, only if $P(m^i, t_j) = 1$. If $t_j \in T_e(N, m^i)$ for some marking $m^i$, we say the transition $t_j$ is state-enabled at $m^i$. If $P(m^i, t_j) = 1$, we say the transition $t_j$ is control-enabled at $m^i$. A transition has to be state- and control-enabled before it can fire. The fact that uncontrollable transitions cannot be prevented from firing by the supervisory policy is captured by the requirement that $\forall m^i \in \mathcal{N}^n, P(m^i, t_j) = 1$, if $t_j \in T_u$. This is implicitly assumed of any supervisory policy in this paper.

A string of transitions $\sigma = t_1t_2 \cdots t_k$, where $t_j \in T(j \in \{1, 2, \ldots, k\})$ is said to be a valid firing string starting from the marking $m^i$, if,

1. $t_1 \in T_e(N, m^i), P(m^i, t_1) = 1$, and

2. for $j \in \{1, 2, \ldots, k-1\}$ the firing of the transition $t_j$ produces a marking $m^{i+j}$ and $t_{j+1} \in T_e(N, m^{i+j})$ and $P(m^{i+j}, t_{j+1}) = 1$. 


The set of reachable markings under the supervision of $\mathcal{P}$ in $N$ from the initial marking $m^0$ is denoted by $\mathcal{R}(N, m^0, \mathcal{P})$. A transition $t_k$ is live under the supervision of $\mathcal{P}$ if

$$\forall m^i \in \mathcal{R}(N, m^0, \mathcal{P}), \exists m^j \in \mathcal{R}(N, m^i, \mathcal{P}) \text{ such that } t_k \in T_e(N, m^j) \text{ and } \mathcal{P}(m^j, t_k) = 1.$$ 

A supervisory policy $\mathcal{P}$ enforces liveness if all transitions in $N(m^0)$ are live under $\mathcal{P}$. The existence of a supervisory policy that enforces liveness in an arbitrary FCPN is decidable [3].

In the next chapter we consider supervisory policies that enforce liveness in a family of PNs obtained by the refinement process of Suzuki and Murata defined in the previous subsection.
CHAPTER 3

ON SUPERVISORY POLICIES THAT ENFORCE LIVENESS IN A PN OBTAINED BY THE REFINEMENT

Following the discussion in sections 2.1 and 2.2, we impose a restriction on
the PN $N_1(m_1^0)$, where $N_1 = (\Pi_1, T_1, \Phi_1)$, and $T_1 = T_{1c} \cup T_{1u}$, where $T_{1c}$ ($T_{1u}$) denotes the set of controllable (uncontrollable) transitions – (P1) the transition $t_0 \in T_{1c}$.

For the PN $N_2(m_2^0)$, $N_2 = (\Pi_2, T_2, \Phi_2)$, $\{t_{in}, t_{out}\} \subseteq T_2$, and $T_2 = T_{2c} \cup T_{2u}$, where $T_{2c}$ ($T_{2u}$) denotes the set of controllable (uncontrollable) transitions, we require – (P2) $t_{in} \in T_{2c}$, and (P3) for any valid firing string $\sigma_2 \in T_2^*$ in $\widehat{N}_2(m_2^0), 0 \leq (#(\sigma_2, t_{in}) - #(\sigma_2, t_{out})) \leq 1$ (i.e. WB3 property of section 2.1 holds for $k = 1$).

The next section addresses the issue of deciding these properties in an arbitrary PN.

3.1 Deciding Properties P1, P3 and P3 in arbitrary PNs

Requirements P1 and P2 are straightforward to verify, and are therefore decidable. The following observation notes that requirement P3 is decidable too.

Observation 3.1.1. Testing if $N_2(m_2^0)$ satisfies requirement P3 is decidable.

Proof. Since $\widehat{m}_2^0(\pi_0) = 1$, $\pi_0 = \{t_{out}\}$, and $\pi_0 \in* t_{in}$,

$$(#(\sigma_2, t_{in}) - #(\sigma_2, t_{out})) \leq 1$$

for any valid firing string $\sigma_2 \in T_2^*$ in $\widehat{N}_2(m_2^0)$. Therefore, requirement P3 is not met if and only if $\exists \sigma_2 \in T_2^*$ that is valid in $\widehat{N}_2(m_2^0)$, such that $#(\sigma_2, t_{out}) \geq #(\sigma_2, t_{in})$. Equivalently, requirement P3 is not met if and only if $\exists \sigma_2 \in T_2^*$ such that $\widehat{m}_2^0 \rightarrow \sigma_2 \rightarrow \widehat{m}_2^1$ in $\widehat{N}_2(m_2^0)$ such that $\widehat{m}_2^1(\pi_0) \geq 2$. 18
Consider the PN shown in figure 3.1, which involves additions to the existing FC structure of $N_2 = (\Pi_2, T_2, \Phi_2)$ that are defined as: $\Pi_2 \leftarrow \Pi_2 \cup \{\pi_i\}_{i=0}^{5}$, $T_2 \leftarrow T_2 \cup \{\tau_i\}_{i=0}^{5}$, and $\Phi_2 \leftarrow \Phi_2 \cup \{(\pi_0, t_{in}), (t_{out}, \pi_0)\} \cup \{(\pi_1, \pi_i)\}_{i=0}^{4} \cup \{(\tau_i, \pi_{i+1})\}_{i=0}^{3} \cup \{(\pi_i, t_{in})\}_{i=0}^{4} \cup \{(\tau_i, \pi_{i+1})\}_{i=0}^{3} \cup \{(\tau_4, p)\}_{p \in \Pi_2}$. Additionally, all transitions in this net are controllable (i.e. for this PN $T_{2u} = \emptyset$).

There is a supervisory policy that enforces liveness in this PN if and only if $\exists \sigma_2 \in T_2^* \text{ such that } \tilde{m}_2^0 \rightarrow \sigma_2 \rightarrow \tilde{m}_2^1 \text{ in } \tilde{N}_2(\tilde{m}_2^0) \text{ such that } \tilde{m}_2^1(\pi_0) \geq 2$. Since all transitions are controllable, the “if” part of the proof involves control-disabling $\tau_0$, and control-enabling transitions in $T_2$ in such a manner that the only firing string that is valid under supervision is $\sigma_2$. When there are at least two tokens in $\pi_0$, the supervisory policy directs tokens appropriately till transition $\tau_4$ is state-enabled. Since all places in the PN are output places of $\tau_4$, the supervisory policy that control-enables all transitions at the marking that state-enables $\tau_4$, enforces liveness in the PN. The reverse implication is established by noting that there can be no supervisory policy that enforces liveness of transitions $\{\tau_i\}_{i=0}^{4}$ if there is no firing string of $\tilde{N}_2(\tilde{m}_2^0)$ that places at least two tokens in $\pi_0$.

All transitions of this PN are controllable, and consequently the existence of a supervisory policy that enforces liveness in the PN of figure 3.1 is decidable [4].

\[\text{Figure 3.1: A (fully controlled) FCPN that is constructed from the FCPN } N_2(\text{m}_2^0).\]
3.2 Main Result

The remainder of this chapter is about the various components of the proof of the main result of this thesis, which is stated below.

**Theorem 3.2.1.** Let \( N_1(m_1^0) (N_2(m_2^0)) \) be an PN, where \( N_1 = (\Pi_1, T_1, \Phi_1) \) \( (N_2 = (\Pi_2, T_2, \Phi_2)) \) and \( t_0 \in T_1 \ (\{t_{in}, t_{out}\} \subseteq T_2) \). Suppose \( N_1(m_1^0) (N_2(m_2^0)) \) satisfies requirement P1 (P2 and P3), and \( \hat{N}_2(m_2^0) \) is the PN that results when the construction of section 2.1 is applied to \( N_2(m_2^0) \) for \( k = 1 \). \( N_3(m_3^0) \) is the PN that is obtained by the refinement process of section 2.1 using these constituent FCPNs. There is a supervisory policy that enforces liveness in \( N_3(m_3^0) \) if and only if there are similar policies for the FCPNs \( N_1(m_1^0) \) and \( \hat{N}_2(m_2^0) \).

We first show that if there is a supervisory policy \( P_3 \) that enforces liveness in \( N_3(m_3^0) \), there is a supervisory policy \( P_1 (P_2) \) that enforces liveness in \( N_2(m_2^0) (\hat{N}_2(m_2^0)) \). Towards this end, we will need the functions \( f_1 : T_3^* \rightarrow T_1^* \) and \( f_2 : T_3^* \rightarrow T_2^* \), where \( f_1(\lambda) = f_2(\lambda) = \lambda \), where \( \lambda \) is the empty string. For any \( \sigma \in T_3^* \),

\[
f_1(\sigma t) = \begin{cases} f_1(\sigma) & t \in T_2 - \{t_{in}\} \\ f_1(\sigma)t_0 & t = t_{in} \\ f_1(\sigma)t & t \in T_1 \end{cases} \quad f_2(\sigma t) = \begin{cases} f_2(\sigma) & t \in T_1 \\ f_2(\sigma)t & t \in T_2 \end{cases}.
\]

For a supervisory policy \( P_3 : N^{card(\Pi_3)} \times T_3 \rightarrow \{0, 1\} \), the supervisory policy \( P_1 : N^{card(\Pi_1)} \times T_1 \rightarrow \{0, 1\} \) is defined as

\[
(P_1(m_1^1, t) = 1) \Leftrightarrow (t \in T_{1w}) \lor \{\exists \sigma_3 \in T_3^*, \text{ such that} \\
(m_3^0 \rightarrow \sigma_3 \rightarrow m_3^1 \text{ under } P_3 \text{ in } N_3(m_3^0)) \\
\land (P_3(m_3^1, t) = 1) \\
\land (m_1^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^0 \text{ under } P_1 \text{ in } N_1(m_1^0)) \\
\land (\forall p \in \Pi_1, m_1^1(p) = \\
m_1^1(p) \land \text{card}(\{r_{out}^{\bullet}$\}_N_3 \times \{p\}) \times \\
(\#(\sigma_3, t_{in}) - \#(\sigma_3, t_{out})) \}
\]}.
\]
The supervisory policy \( \mathcal{P}_2 : \mathcal{N}^{\text{card}(\Pi_2 \cup \{\pi_0\})} \times T_2 \to \{0, 1\} \) is defined as

\[
(P_2(m_2^1, t) = 1) \iff (t \in T_{2a}) \lor \{\exists \sigma_3 \in T_3^* \text{ such that } (m_3^0 \to \sigma_3 \to m_3^1 \text{ under } \mathcal{P}_3 \text{ in } N_3(m_3^0))
\]

\[
\land (P_3(m_3^1, t) = 1)
\]

\[
\land (\tilde{m}_2^0 \to f_2(\sigma_3) \to \tilde{m}_2^1 \text{ under } \mathcal{P}_2 \text{ in } \tilde{N}_2(\tilde{m}_2^0))
\]

\[
\land (\forall p \in \Pi_1, \tilde{m}_2^1(p) = m_3^1(p)) \}.
\]

The following observation notes that for every string that is valid under the supervision of \( \mathcal{P}_2 \) in \( \tilde{N}_2(\tilde{m}_2^0) \) has a corresponding string that is valid under the supervision of \( \mathcal{P}_3 \) in \( N_3(m_3^0) \).

**Observation 3.2.2.** If \( \tilde{m}_2^0 \to \sigma_2 \to \tilde{m}_2^1 \) under the supervision of \( \mathcal{P}_2 \) of equation 3.2, then \( \exists \sigma_3 \in T_3^* \) such that (1) \( f_2(\sigma_3) = \sigma_2 \), (2) \( m_3^0 \to \sigma_3 \to m_3^1 \) under the supervision of \( \mathcal{P}_3 \) in \( N_3(m_3^0) \), and (3) \( \forall p \in \Pi_2, m_3^1(p) = m_2^1(p) \).

**Proof.** We use an induction argument based on the length of the string \( \sigma_2 \).

The base case is established by taking \( \sigma_2 = \lambda \), the empty string. The induction hypothesis assumes the observations holds for any \( \sigma_2 \) of a particular length, and the induction step assumes \( \tilde{m}_2^0 \to \sigma_2 \to \tilde{m}_2^1 \to t \to \tilde{m}_2^2 \) under \( \mathcal{P}_2 \) of equation 3.2, for some \( t \in T_2 \).

If \( t \in T_{2a} \), the existence of the of a string \( \sigma_3 \in T_3^* \) follows directly from the induction hypothesis and equation 3.2. Therefore, \( m_3^0 \to \sigma_3 \to m_3^1 \to t \to m_3^2 \) under \( \mathcal{P}_3 \) in \( N_3(m_3^0) \), \( f_2(\sigma_3 t) = \sigma_2 t \), and from the construction of \( N_3(m_3^0) \), we have \( \forall p \in \Pi_2, m_3^2(p) = \tilde{m}_2^2(p) \).

If \( t \in T_{2a} \), then from requirement P2 we have \( t \in T_2 - \{t_{in}\} \). From the induction hypothesis, we have \( \forall p \in \Pi_2, m_3^1(p) = \tilde{m}_2^1(p) \), therefore \( m_3^0 \to \sigma_3 \to m_3^1 \to t \to m_3^2 \) under the supervision of \( \mathcal{P}_3 \) in \( N_3(m_3^0) \). Furthermore, from the construction of \( N_3(m_3^0) \), we have \( \forall p \in \Pi_2, m_3^2(p) = \tilde{m}_2^2(p) \), and \( f_2(\sigma_3 t) = f_2(\sigma_3) t = \sigma_2 t \). \( \Box \)

The following observation notes that any string that is valid under the supervision of \( \mathcal{P}_3 \) in \( N_3(m_3^0) \) has a corresponding valid firing string under the supervision of \( \mathcal{P}_2 \).

**Observation 3.2.3.** If \( m_3^0 \to \sigma_3 \to m_3^1 \) under the supervision of \( \mathcal{P}_3 \) in \( N_3(m_3^0) \), then (1) \( \tilde{m}_2^0 \to f_2(\sigma_3) \to \tilde{m}_2^1 \) under the supervision of \( \mathcal{P}_2 \) of equation 3.2, and (2) \( \forall p \in \Pi_2, m_3^1(p) = m_2^1(p) \).
This observation is established by an induction argument over the length of \( \sigma_3 \), and is skipped for brevity. Observations 3.2.2 and 3.2.3 imply the following observation about the existence of liveness enforcing policies in \( \tilde{N}_2(\tilde{m}_2^0) \), if there is a similar policy for \( N_3(m_3^0) \).

**Observation 3.2.4.** If the supervisory policy \( P_3 \) enforces liveness in \( N_3(m_3^0) \), then the supervisory policy \( P_2 \) of equation 3.2 enforces liveness in \( \tilde{N}_2(\tilde{m}_2^0) \).

**Proof.** Suppose \( \tilde{m}_2^0 \rightarrow \sigma_2 \rightarrow \tilde{m}_2^1 \) under the supervision of \( P_2 \) in \( \tilde{N}_2(\tilde{m}_2^0) \). From observation 3.2.2, there exists \( \sigma_3^1 \in T_3^* \) such that (1) \( f_2(\sigma_3^1) = \sigma_2 \), (2) \( m_3^0 \rightarrow \sigma_3^1 \rightarrow m_3^1 \) under the supervision of \( P_3 \) in \( N_3(m_3^0) \), and (3) \( \forall p \in \Pi_2, m_3^1(p) = m_2^1(p) \).

Since \( P_3 \) enforces liveness in \( N_3(m_3^0) \), \( \forall t \in T_2(\subseteq T_3) \), there exists \( \sigma_3^2 \in T_3^* \) such that \( m_3^1 \rightarrow \sigma_3^2 t \rightarrow m_3^2 \) under the supervision of \( P_3 \) in \( N_3(m_3^0) \). Since \( m_3^0 \rightarrow \sigma_3^1 \sigma_3^2 t \rightarrow m_3^2 \), from observation 3.2.3 we have \( \tilde{m}_2^0 \rightarrow f_2(\sigma_3^1 \sigma_3^2 t) \rightarrow \tilde{m}_2^1 \rightarrow \tilde{m}_2^0 \rightarrow \sigma_2 f_1(\sigma_3^2) t \rightarrow \tilde{m}_2^1 \) under the supervision of \( P_2 \) in \( \tilde{N}_2(\tilde{m}_2^0) \). That is, \( P_2 \) enforces liveness in \( \tilde{N}_2(\tilde{m}_2^0) \). \( \square \)

The following observation notes that any policy \( P_2 \) that enforces liveness in \( \tilde{N}_2(\tilde{m}_2^0) \) also enforces the 1-WB property of section 2.1 in \( \tilde{N}_2(\tilde{m}_2^0) \).

**Observation 3.2.5.** If the supervisory policy \( P_2 \) enforces liveness in \( \tilde{N}_2(\tilde{m}_2^0) \), then it also enforces (WB1), (WB2) and (WB3) property of section 2.1 in \( \tilde{N}_2(\tilde{m}_2^0) \).

**Proof.** The unsupervised behavior of \( \tilde{N}_2(\tilde{m}_2^0) \) satisfies requirement P3. This property holds under supervision too. Therefore, \( P_2 \) enforces the (WB3) property. Since \( P_2 \) enforces liveness in \( \tilde{N}_2(\tilde{m}_2^0) \), it enforces the (WB1) property. The (WB3) property, and the liveness of \( t_{out} \) under the supervision of \( P_2 \), implies that the (WB2) property is also true under supervision. \( \square \)

The following observation notes that if \( P_3 \) enforces liveness in \( N_3(m_3^0) \), then for every valid firing string under the supervision of \( P_1 \) of equation 3.1 in \( N_1(m_1^0) \), there exists a corresponding firing string that is valid under the supervision of \( P_3 \) in \( N_3(m_3^0) \).

**Observation 3.2.6.** If the supervisory policy \( P_3 \) enforces liveness in \( N_3(m_3^0) \), and \( m_1^0 \rightarrow \sigma_1 \rightarrow m_1^1 \) under the supervision of \( P_1 \) of equation 3.1 in \( N_1(m_1^0) \), then \( \exists \sigma_3 \in T_3^* \) such that (1) \( f_1(\sigma_3) = \sigma_1 \), and (2) \( m_3^0 \rightarrow \sigma_3 \rightarrow m_3^1 \) under the supervision of \( P_3 \) in \( N_3(m_3^0) \).
Proof. (Sketch) The proof is via an induction argument over \(\#(\sigma_1, t_0)\). The base case is established by an new induction argument over the length of \(\sigma_1\), for any \(\sigma_1\) where \(\#(\sigma_1, t_0) = 0\). The induction hypothesis assumes the observation to be true for any \(\sigma_1 \in T_1^*\) where \(\#(\sigma_1, t_0) \leq k\) for some \(k \in \mathcal{N}\). For the induction step, without loss in generality, we suppose \(m_0^0 \rightarrow \sigma_1 \rightarrow m_1^0 \rightarrow t_0 \rightarrow m_2^0\) under the supervision of \(\mathcal{P}_1\) in \(N_1(m_3^0)\). By observations 3.2.4 and 3.2.5, the supervisory policy \(\mathcal{P}_2\) enforces the \((\text{WB1}), (\text{WB2})\) and \((\text{WB3})\) property of section 2.1. From observations 3.2.3 and 3.2.2 \(\exists \sigma_1^3, \sigma_2^3 \in T_2^*\) such that \(\#(\sigma_1^3, t_{in}) = \#(\sigma_2^3, t_{out}) = 1, \#(\sigma_1^3, t_{out}) = \#(\sigma_2^3, t_{in}) = 0\), and \(m_0^0 \rightarrow \sigma_3 \rightarrow m_3^1 \rightarrow \sigma_2^3 \rightarrow m_2^3\) under \(\mathcal{P}_3\) in \(N_3(m_3^0)\). Additionally, \(f_1(\sigma_3^1 \sigma_2^3) = f_1(\sigma_3) f_1(\sigma_2^3) = \sigma_3 t\), which proves the induction step. \(\square\)

The next observation is about the existence of a valid firing string under the supervision of \(\mathcal{P}_1\) of equation 3.1 in \(N_1(m_0^0)\) for each valid string under the supervision of \(\mathcal{P}_3\) in \(N_3(m_3^0)\).

**Observation 3.2.7.** If \(m_3^0 \rightarrow \sigma_3 \rightarrow m_3^1\) under the supervision of \(\mathcal{P}_3\) in \(N_3(m_3^0)\), then (1) \(m_0^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1\) under the supervision of \(\mathcal{P}_1\) of equation 3.1, and (2) \(\forall \sigma \in \Pi_1, m_1^1(p) = m_3^1(p) + \text{card}((t_{out})_{N_3} \cap \{p\}) \times (\#(\sigma, t_{in}) - \#(\sigma, t_{out}))\).

Proof. By an induction argument over the length of \(\sigma_3\). The base case is established for \(\sigma_3 = \lambda\). As the induction hypothesis, we assume the observation holds for all cases where the length of \(\sigma_3\) is less than or equal to some value. For the induction step we assume \(m_0^0 \rightarrow \sigma_3 \rightarrow m_3^1 \rightarrow t \rightarrow m_3^3\) under the supervision of \(\mathcal{P}_3\) in \(N_3(m_3^0)\).

Since \(f_1(t) = \lambda\), if \(t \in T_2 - \{t_{in}, t_{out}\}\), the induction step is easily established for this case. If \(t \in T_1\) \(\Rightarrow t \neq t_0\), \(f_1(t) = t\), and the induction step is proven by replacing the string \(\sigma_3\) with the string \(\sigma_3 t\).

If \(t = t_{in}, f_1(t_{in}) = t_0\). Since \(\forall \sigma \in \Pi_1, m_3^1(p) \geq m_3^1(p), t \in T_2(N_1, m_1^1)\). Additionally, from equation 3.1, \(\mathcal{P}_1(m_1^1, t) = 1\). So, \(m_0^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1 \rightarrow t_0 \rightarrow m_2^0\) under the supervision of \(\mathcal{P}_1\) in \(N_1(m_3^0)\). Using the fact that \((t_{in})_{N_1} = (t_{out})_{N_3} \cap \Pi_1, \text{card}((t_{out})_{N_3} \cap \Pi_1) = 0\), we can obtain the expression \(\forall \sigma \in \Pi_2, m_2^3(p) = m_3^3(p) + \text{card}((t_{out})_{N_3} \cap \{p\}) \times (\#(\sigma t_{in}, t_{in}) - \#(\sigma t_{in}, t_{out}))\).

If \(t = t_{out}, f_1(t_{out}) = \lambda\), then the fact that \(m_0^0 \rightarrow f_1(\sigma_3 t_{out}) \rightarrow m_1^1\) under the supervision of \(\mathcal{P}_1\) in \(N_1(m_3^0)\) follows from the induction hypothesis.
For any $p \in \Pi_2$, we have the following expression for $m_i^2(p) = m_i^1(p) - m_i^2(p) = m_i^1(p) + \text{card}((t^*_\text{out})_{N_3} \cap \{p\}) \times \#(\sigma_3, t_{in}) - \#(\sigma_3, t_{out}) = m_i^2(p) + \text{card}((t^*_\text{out})_{N_3} \cap \{p\}) \times \#(\sigma_3 t_{out}, t_{in}) - \#(\sigma_3 t_{out}, t_{out})$, which completes the induction step.

The following observation notes that the supervisory policy $P_1$ of equation 3.1 enforces liveness in $N_1(m_i^0)$, if $P_3$ enforces liveness in $N_3(m_i^0)$.

**Observation 3.2.8.** If the supervisory policy $P_3$ enforces liveness in $N_3(m_i^0)$, then the supervisory policy $P_1$ of equation 3.1 enforces liveness in $N_1(m_i^0)$.

The proof of this observation parallels that of observation 3.2.4 with appropriate changes, and is skipped for brevity. Observations 3.2.8 and 3.2.4 together imply the following lemma.

**Lemma 3.2.9.** The existence of a supervisory policy that enforces liveness in the PNs $N_1(m_i^0)$ and $\tilde{N}_2(m_i^0)$ is necessary for the existence of a similar policy for the PN $N_3(m_i^0)$.

To show the sufficiency of the above observation we define a supervisory policy $\tilde{P}_3 : \mathcal{N}^{\text{card}(\Pi_3)} \times T_3 \to \{0, 1\}$ in terms of policies $\tilde{P}_1 : \mathcal{N}^{\text{card}(\Pi_1)} \times T_1 \to \{0, 1\}$ and $\tilde{P}_2 : \mathcal{N}^{\text{card}(\Pi_2)} \times T_3 \to \{0, 1\}$ as follows

$$\tilde{P}_3(m_i^3, t) = 1 \iff (t \in T_{3u}) \lor \left\{ \exists \sigma_3 \in T_3^* \text{ such that } \left( m_i^0 \rightarrow \sigma_3 \rightarrow m_i^3 \text{ under } \tilde{P}_3 \text{ in } N_3 \right) \land \left( m_i^0 \rightarrow f_1(\sigma_3) \rightarrow m_i^1 \rightarrow f_1(t) \rightarrow m_i^2 \text{ under } \tilde{P}_1 \text{ in } N_1 \right) \land \left( \tilde{m}_i^0 \rightarrow f_2(\sigma_3) \rightarrow \tilde{m}_i^1 \rightarrow f_2(t) \rightarrow \tilde{m}_i^2 \text{ under } \tilde{P}_2 \text{ in } N_2 \right) \right\}.$$  \hspace{1cm} (3.3)

The following observation about valid firing strings under the supervision of $\tilde{P}_3$ in $N_3(m_i^0)$ and their corresponding strings in $N_1(m_i^0)$ under the supervision of $\tilde{P}_1$, and $\tilde{N}_1(m_i^0)$ under the supervision of $\tilde{P}_2$.

**Observation 3.2.10.** Suppose $m_i^0 \rightarrow \sigma_3 \rightarrow m_i^3$ under the supervision of $\tilde{P}_3$ of equation 3.3 in $N_3(m_i^0)$. Then (1) $m_i^0 \rightarrow f_1(\sigma_3) \rightarrow m_i^1$ under the supervision of $\tilde{P}_1$ in $N_1(m_i^0)$, (2) $\forall p \in \Pi_1, m_i^1(p) = m_i^4(p) + \text{card}((t^*_\text{out})_{N_3} \cap \{p\}) \times \#(\sigma_3, t_{in}) - \#(\sigma_3, t_{out})$, (3) $\tilde{m}_i^0 \rightarrow f_2(\sigma_3) \rightarrow \tilde{m}_i^1$ under the supervision of $\tilde{P}_2$ in $\tilde{N}_1(\tilde{m}_i^0)$, and (4) $\forall p \in \Pi_2, m_i^4(p) = m_i^3(p)$.
Proof. This result is established by induction on the length of $\sigma_3$. The base case is established by letting $\sigma_3 = \lambda$, the empty string. The induction hypothesis assumes the observation to be true for all cases where the length of $\sigma_3$ is less than or equal to some integer. The induction step supposes $m_3^0 \rightarrow \sigma_3 \rightarrow m_3^1 \rightarrow t \rightarrow m_3^2$ under the supervision of $\wedge_3$ of equation 3.3 in $N_3(m_3^0)$.

If $t \in T_3$, from equation 3.3 we infer $m_1^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1 \rightarrow f_1(t) \rightarrow m_1^1$ under $\wedge_1$ in $N_1(m_1^1)$, and $\tilde{m}_1^0 \rightarrow f_2(\sigma_3) \rightarrow \tilde{m}_2^1 \rightarrow f_2(t) \rightarrow \tilde{m}_2^2$ under $\wedge_2$ in $N_2(\tilde{m}_2^0)$. The remainder of the observation follows directly from these two facts.

If $t \in T_3 \{ \exists \neq t_{in} \}$. Suppose $t \in T_1 \{ t_{in} \}$, then $f_1(t) = t$ and $f_2(t) = \lambda$. If $t \in T_2 \{ t_{in} \}$, then $f_1(t) = \lambda$ and $f_2(t) = t$. In either case, from the induction hypothesis, and the fact that $t \in T_e(N_3, m_3^0)$, we have $f_1(t) \in T_e(N_1, m_1^1)$ and $f_2(t) \in T_e(N_2, m_2^0)$. The remainder of the observation follows directly from the fact that $m_1^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1 \rightarrow f_1(t) \rightarrow m_1^1$ under $\wedge_1$ in $N_1(m_1^0)$, and $\tilde{m}_1^0 \rightarrow f_2(\sigma_3) \rightarrow \tilde{m}_2^1 \rightarrow f_2(t) \rightarrow \tilde{m}_2^2$ under $\wedge_2$ in $N_2(\tilde{m}_2^0)$, which completes the proof.

The following observation will find use in the proof of lemma 3.2.12.

**Observation 3.2.11.** If the supervisory policy $\wedge_2$ enforces liveness in $N_2(\tilde{m}_2^0)$, and if $m_3^0 \rightarrow \sigma_3^1 \rightarrow m_3^1$ under the supervision of $\wedge_2$ in $N_3(m_3^0)$, then $\exists \sigma_3^2 \in (T_3 - \{ t_{in} \})^*$ such that $m_3^0 \rightarrow \sigma_3^1 \rightarrow m_3^1 \rightarrow \sigma_3^2 \rightarrow m_3^2$ under the supervision of $\wedge_2$ in $N_3(m_3^0)$ such that $\#(\sigma_3^1 \sigma_3^2, t_{in}) = \#(\sigma_3^1 \sigma_3^2, t_{out})$.

Proof. Following observation 3.2.10, $m_1^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1$ under the supervision of $\wedge_1$ in $N_1(m_1^0)$, and $\tilde{m}_2^0 \rightarrow f_2(\sigma_3) \rightarrow \tilde{m}_2^1$ under the supervision of $\wedge_2$ in $N_1(\tilde{m}_2^0)$.

Since $\wedge_2$ enforces liveness in $N_2(\tilde{m}_2^0)$, from observation 3.2.5, it also enforces the $(WB2)$ property. Therefore, $\exists \sigma_2 \in (T_2 - \{ t_{in} \})^*$ such that $\tilde{m}_2^0 \rightarrow f_2(\sigma_2) \rightarrow \tilde{m}_2^1 \rightarrow \sigma_2 \rightarrow \tilde{m}_2^2$ under $\wedge_2$ in $N_2(\tilde{m}_2^0)$, and $\#(f_2(\sigma_2) \sigma_2, t_{in}) = \#(f_2(\sigma_2) \sigma_2, t_{out})$. Since $f_1(\sigma_2) = \lambda$, $m_1^0 \rightarrow f_1(\sigma_3) \rightarrow m_1^1 \rightarrow f_1(\sigma_2) \rightarrow m_1^1$ under the supervision of $\wedge_1$ in $N_1(m_1^0)$.

From observation 3.2.10, $\tilde{m}_2^1(p) = m_1^1(p), \forall p \in \Pi_1$. This, along with equation 3.3, implies that $m_3^0 \rightarrow \sigma_3^1 \rightarrow m_3^1 \rightarrow \sigma_2 \rightarrow m_3^2$ under the supervision of $\wedge_2$ in $N_3(m_3^0)$.

The observation follows from letting $\sigma_3^2 = \sigma_2$.  \hfill $\blacksquare$
The following lemma notes that the existence of a supervisory policy that enforces liveness in $N_1(m_0^n)$ and $\widehat{N}_2(\tilde{m}_0^0)$ is sufficient for the existence of a similar policy for $N_3(m_0^3)$.

**Lemma 3.2.12.** If $\widehat{P}_1$ and $\widehat{P}_2$ enforce liveness in $N_1(m_0^n)$ and $\widehat{N}_2(\tilde{m}_0^0)$ respectively, then $\widehat{P}_3$ of equation 3.3 enforces liveness in $N_3(m_0^3)$.

**Proof.** Suppose $m_0^3 \rightarrow \sigma_3^1 \rightarrow m_3^3$ under $\widehat{P}_3$ in $N_3(m_0^3)$. By observation 3.2.11, $\exists \sigma_3^2 \in (T_2 - \{t_{in}\})^*$ such that $m_3^3 \rightarrow \sigma_3^1 \rightarrow m_3^4 \rightarrow \sigma_3^2 \rightarrow m_3^3$ under $\widehat{P}_3$ in $N_3(m_0^3)$, and $\#(\sigma_3^1\sigma_3^2, t_{in}) = \#(\sigma_3^1\sigma_3^2, t_{out})$. From observation 3.2.10, we have $m_0^0 \rightarrow f_1(\sigma_3^1) \rightarrow m_1^1 \rightarrow f_1(\sigma_3^2) \rightarrow m_1^2$ under $\widehat{P}_1$ in $N_1(m_0^0)$, where $\forall p \in \Pi_1, m_1^2(p) = m_1^1(p)$. Also, $\tilde{m}_1^0 \rightarrow f_2(\sigma_3^1) \rightarrow \tilde{m}_2^1 \rightarrow f_2(\sigma_3^2) \rightarrow \tilde{m}_2^2$ under $\widehat{P}_2$ in $\widehat{N}_1(\tilde{m}_1^0)$, where $\forall p \in \Pi_2, m_2^2(p) = \tilde{m}_2^2(p)$.

Since $\widehat{P}_1$ enforces liveness in $N_1(m_1^1)$, $\forall t \in T_1, \exists \sigma_1 \in T^*$ such that $m_0^1 \rightarrow f_1(\sigma_3^1) \rightarrow m_1^1 \rightarrow f_1(\sigma_3^2) \rightarrow m_1^2 \rightarrow \sigma t \rightarrow m_1^3$ under $\widehat{P}_1$ in $N_1(m_0^0)$. We claim that $\exists \sigma_3^2 \in T^*$ such that $m_3^3 \rightarrow \sigma_3^1 \rightarrow m_3^4 \rightarrow \sigma_3^2 \rightarrow m_3^5 \rightarrow \sigma_3^3 \rightarrow m_3^3$ under $\widehat{P}_3$ in $N_3(m_0^3)$, where $f_1(\sigma_3^3) = \sigma_1 t$ and $\forall p \in \Pi_1, m_3^5(p) = m_3^3(p)$. As a consequence, $\tilde{m}_1^0 \rightarrow f_2(\sigma_3^1) \rightarrow \tilde{m}_2^1 \rightarrow f_2(\sigma_3^2) \rightarrow \tilde{m}_2^2 \rightarrow f_2(\sigma_3^3) \rightarrow \tilde{m}_2^3$ under the supervision of $\widehat{P}_2$ in $\widehat{N}_1(\tilde{m}_1^0)$, where $\forall p \in \Pi_2, m_3^5(p) = \tilde{m}_3^5(p)$.

This claim is established by an induction argument over $\#(\sigma_1 t, t_0)$. The base case is established when $\#(\sigma_1 t, t_0) = 0$ as $\forall p \in \Pi_1, m_3^5(p) = m_3^3(p)$. Therefore, $m_3^3 \rightarrow \sigma_3^1 \rightarrow m_3^4 \rightarrow \sigma_3^2 \rightarrow m_3^5 \rightarrow \sigma_2 t \rightarrow m_3^3$ under $\widehat{P}_3$ in $N_3(m_0^3)$. Additionally, $\forall p \in \Pi_1, m_3^5(p) = m_3^3(p)$.

The induction hypothesis assumes the claim to be true when $\#(\sigma_1 t, t_0) \leq k$ for some $k \in \mathcal{N}$.

Without loss of generality (along with some abuse of notation), the induction step supposes $m_0^0 \rightarrow f_1(\sigma_3^1) \rightarrow m_1^1 \rightarrow f_1(\sigma_3^2) \rightarrow m_1^2 \rightarrow \sigma t \rightarrow m_1^3 \rightarrow t_0 \rightarrow m_1^4$, where $\#(\sigma_1, t_0) = k$. This coincides with (1) $m_0^0 \rightarrow \sigma_3^1 \rightarrow m_3^1 \rightarrow \sigma_3^2 \rightarrow m_3^2 \rightarrow \sigma_3^3 \rightarrow m_3^3$ under $\widehat{P}_3$ in $N_3(m_0^3)$, where $f_1(\sigma_3^3) = \sigma_1 t$ and $\forall p \in \Pi_1, m_3^5(p) = m_3^3(p)$, and (2) $\tilde{m}_1^0 \rightarrow f_2(\sigma_3^1) \rightarrow \tilde{m}_2^1 \rightarrow f_2(\sigma_3^2) \rightarrow \tilde{m}_2^2 \rightarrow f_2(\sigma_3^3) \rightarrow \tilde{m}_2^3$ under $\widehat{P}_2$ in $\widehat{N}_1(\tilde{m}_1^0)$, where $\forall p \in \Pi_2, m_3^5(p) = \tilde{m}_3^5(p)$.

Since $t_{in}$ is live under the supervision of $\widehat{P}_2$ in $\widehat{N}_2(\tilde{m}_2^0)$, it follows that $\exists \sigma_2 \in (T_2 - \{t_{in}\})^*$ such that $\tilde{m}_1^0 \rightarrow f_2(\sigma_3^1) \rightarrow \tilde{m}_2^1 \rightarrow f_2(\sigma_3^2) \rightarrow \tilde{m}_2^2 \rightarrow f_2(\sigma_3^3) \rightarrow \tilde{m}_2^3 \rightarrow \sigma t \rightarrow \tilde{m}_2^4 \rightarrow t_{in} \rightarrow \tilde{m}_2^5$ under the supervision of $\widehat{P}_2$ in $\widehat{N}_1(\tilde{m}_1^0)$. Since $f_1(\sigma_2) = \lambda$, from equation 3.3, it follows that $m_3^3 \rightarrow \sigma_3^1 \rightarrow m_3^4 \rightarrow \sigma_3^2 \rightarrow m_3^5 \rightarrow \sigma_3^3 \rightarrow m_3^3 \rightarrow \sigma_2 \rightarrow m_3^4 \rightarrow t_{in} \rightarrow m_3^3$, completing the induction step. Therefore, all transitions in $T_1$ are live under $\widehat{P}_3$ in $N_3(m_0^3)$. 


With appropriate changes, the above argument can also be used to show that all transitions in $T_2$ are live under $\hat{P}_3$ in $N_3(\mathbf{m}_3^0)$, which completes the proof. \hfill \Box

Lemma 3.2.12 and 3.2.9 together imply theorem 3.2.1 introduced at the beginning of this chapter. If there is a supervisory policy that enforces liveness in $N_3(\mathbf{m}_3^0)$, then there is always a distributed implementation of a liveness enforcing policy. To see this, we note that in those instances where there is a supervisory policy that does the same [4]. If the FCPNs

$$\hat{P}_1$$

and the similar policy $\hat{P}_2$ for $\tilde{N}_2(\tilde{m}_2^0)$ are combined to yield a similar policy $\hat{P}_3$ for $N_3(\mathbf{m}_3^0)$. Under this scheme, for a marking $\mathbf{m}_3^1$, where $\mathbf{m}_3^0 \rightarrow \sigma_3 \rightarrow \mathbf{m}_3^1$ under $\hat{P}_3$,

$$\hat{P}_3(\mathbf{m}_3^1, t) = \begin{cases} 
\hat{P}_1(\mathbf{m}_3^1 \restriction \Pi_1, t) & \text{if } t \in T_e(N_3, \mathbf{m}_3^1) \cap (T_1 - \{t_0\}) \\
\hat{P}_2(\Delta(\mathbf{m}_3^1, \sigma_3), t) & \text{if } t \in T_e(N_3, \mathbf{m}_3^1) \cap (T_2 - \{t_{in}\}) \\
\hat{P}_1(\mathbf{m}_3^1 \restriction \Pi_1, t_0) \wedge \\
\hat{P}_2(\Delta(\mathbf{m}_3^1, \sigma_3), t_{in}) & \text{if } t \in T_e(N_3, \mathbf{m}_3^1) \cap \{t_{in}\} \\
0 & \text{otherwise,}
\end{cases}$$

where $\mathbf{m}_3^1 \restriction \Pi_1$ is the restriction of the marking $\mathbf{m}_3^1$ to places in $\Pi_1$ (i.e. $\mathbf{m}_3^1 \restriction \Pi_1(p) = \mathbf{m}_3^1(p)$, $\forall p \in \Pi_1$); $\Delta : N^{card(\Pi_1)} \times T_3^* \rightarrow N^{card(\Pi_2 \cup \{\pi_0\})}$ is a marking function defined as $\forall p \in \Pi_2, \Delta(\mathbf{m}_3^1, \sigma_3)(p) = \mathbf{m}_3^1(p)$, and $\Delta(\mathbf{m}_3^1, \sigma_3)(\pi_0) = 1 - (\#(\sigma_3, t_{in}) - \#(\sigma_3, t_{out}))$.

In addition to $P1$, $P2$ and $P3$, suppose we required $N_1(\mathbf{m}_1^0)$ and $N_2(\mathbf{m}_2^0)$ be FCPNs. Additionally, let us require that $t_{in} \in T_{e}$ be a non-choice transition\(^1\), then $\tilde{N}_2(\tilde{m}_2^0)$ is guaranteed to be an FCPN too. From theorem 3.2.1, we gather that there is a supervisory policy that enforces liveness in $N_3(\mathbf{m}_3^0)$ if and only if the FCPNs $N_1(\mathbf{m}_1^0)$ and $\tilde{N}_2(\tilde{m}_2^0)$ have similar supervisory policies. Since the existence of a supervisory policy in an arbitrary FCPN is decidable [3], it follows that we can decide the existence of liveness enforcing supervisory policies for $N_3(\mathbf{m}_3^0)$. In addition, if there is a liveness enforcing policy for an arbitrary PN, then there is a unique minimally restrictive policy that does the same [4]. If the FCPNs $\tilde{N}_2(\tilde{m}_2^0)$ and $N_1(\mathbf{m}_1^0)$ can be made live by supervision by $\hat{P}_2$ and $\hat{P}_1$ respectively. Without loss in generality, we can assume these policies are minimally restrictive. Since minimally restric-

\(^1\)That is, $(\bullet t_{in})_{N_2} = \emptyset$, or, $(\bullet t_{in})_{N_2} = \{t_{in}\}
tive policies that enforce liveness in FCPNs do not control-disable non-choice transitions [7], it follows that $\hat{P}_2$ will never control-disable $t_{in}$. From equation 3.3, we gather that the transition $t_{in}$ is control-disabled in $N_3(m_0^3)$ if and only if it is control-disabled by $\hat{P}_1$ for the equivalent marking in $N_1(m_1^0)$.

As an illustration consider the FCPN $N_1(m_1^0)$ shown in figure 3.2(a) and the FCPN $N_2(m_2^0)$ shown in figure 3.2(b). The FCPN $N_1(m_1^0)$ meets requirement $P1$, and the FCPN $N_2(m_2^0)$ meets requirement $P2$ and $P3$. Specifically, requirement $P3$ is enforced by $p_{10} \in t_{in} \cap t_{out}$. Since $(t_{in})_{N_2} = \emptyset$, the PN $\hat{N}_2(\tilde{m}_2^0)$, shown in figure 3.2(c), is also an FCPN.

The FCPN $\hat{N}_2(\tilde{m}_2^0)$ can be made live by the (minimally restrictive) supervisory policy $\hat{P}_2$ that control-disables $t_{11}$ when $p_9$ has the only token in the place-set $\{\pi_0, p_6, p_7, p_8, p_9, p_{11}\}$. This supervisory policy does not control-disable the non-choice transition $t_{in}$ for any reachable marking in $\hat{N}_2(\tilde{m}_2^0)$ (cf. [7]).

A supervisory policy $\hat{P}_1$ that makes sure the current marking of $N_1(m_1^0)$ does not leave the right-closed ] set of markings whose minimal elements are $\{(1 0 0 0 0)^T, (0 0 0 1 1)^T\}$ enforces liveness in $N_1(m_1^0)$ (cf. [3]).

From theorem 3.2.1 we know there is a supervisory policy that enforces liveness in the PN $N_3(m_3^0)$ shown in figure 3.2(d). This supervisory policy can be implemented in a distributed fashion. That is, the decision of control-disabling $t_{11}$ can be made using just the token loads of the place-set $\{\pi_0, p_6, p_7, p_8, p_9, p_{11}\}$, where the token load of (the fictitious place) $\pi_0$ is unity only if the number of occurrences of $t_{in}$ equals that of $t_{out}$ in the past transition firings. The transition $t_{in}$ is control-enabled only when there is at least one token in $p_4$ and $p_5$. That is, these decisions are made just as they were for the constituent FCPNs $\hat{N}_2(\tilde{m}_2^0)$ and $N_1(m_1^0)$. 
Figure 3.2: (a) An FCPN $N_1(m_1^0)$, that meets requirement $P1$. (b) An FCPN $N_2(m_2^0)$ that meets requirements $P2$ and $P3$. (c) $\tilde{N}_2(\tilde{m}_2^0)$, which is also an FCPN, and (d) The FCPN $N_3(m_3^0)$ obtained by the refinement process of section 2.1.
In this thesis we used the refinement procedure of Suzuki and Murata [1], where two PNs $N_1(m_1^0)$ and $N_2(m_2^0)$ are combined in a specific manner to obtain a larger PN $N_3(m_3^0)$. We introduced a restriction ($P1$) on $N_1(m_1^0)$, and two restrictions ($P2$ and $P3$) on $N_2(m_2^0)$. We showed that these restrictions are decidable, and following the construction of Suzuki and Murata [1], we converted the PN $N_2(m_2^0)$ to another PN $\tilde{N}_2(m_2^0)$, by the addition of an extra place and two additional arcs. We showed that there is a supervisory policy that enforces liveness in $N_3(m_3^0)$ if and only if there are similar policies for $N_1(m_1^0)$ and $\tilde{N}_2(m_2^0)$. We showed this result implies the liveness enforcing supervisory policy for $N_3(m_3^0)$, when it exists, can be implemented in a distributed fashion.

4.1 Future work

Future work will involve working on the user interface for petri nets. The first phase would involve creating a petri net, converting non free-choice petri net to a free-choice petri net and attaining the reachability graph and the minimal elements for the nets. The GUI will be developed using Java programming. The next phase will involve extending the GUI to obtain step-wise refinement and abstraction as stated in the Suzuki and Murata [1] paper. The GUI can thus be used to control large petri nets by refining them into smaller ones and finding the supervisory policies for these nets. We aim to develop a software package with complete functionality from producing the right closed set to attaining the reachability graph and finding a supervisory policy for a petri net (large or small). Eventually this can be applied to an operating system thus ensuring that the operating system is always live. Hence, we can overcome the current problems that operating systems face.
where the system reaches a livelock and the user is forced to reboot the system. The future work will thus also involve testing the software package on a complex operating system.
REFERENCES


