Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Studying Multi-operational Production Systems with Modular Hybrid Petri Nets

G. J. Tsinarakis, K. P. Valavanis†, N. C. Tsourveloudis {E-mail: (tsinar, kimonv, nikost)@dpem.tuc.gr}
Technical University of Crete
Intelligent Systems and Robotics Laboratory
Department of Production Engineering and Management
Chania, Greece, GR – 73 100

Abstract — **Modular hybrid timed Petri Nets are used to model and analyze behavior of random topology multi-operational production systems. Each such system is decomposed into 3 fundamental modules, followed by derivation of their corresponding modular HPN models. The overall system model is obtained via synthesis of the individual modules, satisfying system constraints (routes, production rates, machine expected up- down- or idle- time and priorities). For any topology multi-operational production system, overall HTPN model nodes are calculated, followed by derivation of model invariants. Results show applicability of the proposed methodology and justify its modeling power and generality.**

I. INTRODUCTION

Hybrid PNs are a viable mathematical and graphical tool, suitable to model and study hybrid systems [4]. HPNs model sequences of phases of materials continuous flow with elementary discrete controllers; hence they may be used to study multi-operational production systems composed of machines and buffers, where machines produce multiple product types at given time periods (not in parallel) and different types of products follow their own routes through the system. Such systems are high-volume systems where machines produce a variety of product / part types according to the demand and production policies (WIP and backlog minimization). Production times are short and number of produced parts large. Therefore, ordinary PNs are inefficient as an analysis and synthesis tool [2].

HPNs are defined by combining an ordinary with a Continuous PN [3]-[6]. CPNs are derived from timed PNs by "fluidification" relaxing the condition that a marking is integer [7]. In CPNs tokens represent a real quantity of token fragments and transitions move with the velocity of token fragments from preceding places to places after [8]. State space becomes infinite allowing continuous dynamics modeling [9].

This paper draws upon previous work [10]. However, it extends its use by studying multi-operational random topology production systems.

The paper main contribution and originality is that it is the first comprehensive effort to use HTPNs for creating a well-defined framework for systematic study of generalized multi-operational production systems. Further, the modular HTPN approach is independent of the system architecture and structure and analysis and synthesis of any complex system is accomplished in terms of analysis and synthesis of the basic modules.

II. MULTI-OPERATIONAL PRODUCTION SYSTEM MODULES

Expanding the well-justified approach discussed in [13]-[15] and [10], 3 fundamental modules are derived as shown in Figure 1 (circles and rectangles represent buffers, machines). These modules, when connected, represent manufacturing networks of various layouts.

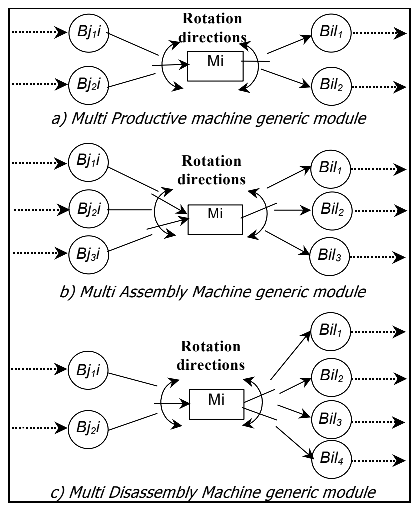

Figure 1: Generic modules of multi-operational production systems.

A generic module is a fundamental subsystem with a set of connection arcs defining its interactions with other modules and a set of discrete and continuous relations defining its internal state. Rotating arcs represent module "modification basis", demonstrating how unprocessed parts reach machines and quit when ready. Arcs indicate that machines are not dedicated and at given time periods produce different products.

Generalizations of the generic modules are obvious and correspond to (n_{pi}) input buffers- 1 machine - (n_{pi}) output buffers multi-productive machine module, (n_{Ai1}) input buffers - 1 machine - (n_{Ai2}) output buffers multi-assembly module and (n_{Di1}) input buffers - 1 machine – (n_{Di2}) output buffers multi-disassembly module.

† Now with the Dept of CSEE, University of South Florida, Tampa, FL 33620 (kvalavan@csee.usf.edu)

III. HYBRID PETRI NET MODULES

A. Hybrid Petri Net Fundamentals

A HTPN is mathematically described by ***HTPN = {P, T, I, O, h, τ, m₀}*** [13]. *P* is the set of places partitioned in subsets of continuous P_c and discrete places P_d. Respectively set of transitions *T* is partitioned in subsets T_c and T_d. *I* and *O* represent input and output incidence mappings: $I : P \times T \rightarrow R^+$ or N and $O : P \times T \rightarrow R^+$ or N. It is required for all $t \in T_c$ and for all $p \in P_d$, ***I(p,t) = O(p,t)***. $h : P \cup T \rightarrow \{D,C\}$ is hybrid function and indicates for every node if it is discrete or continuous. $\tau : T \rightarrow \mathbb{R}^+$ associates each transition with a positive real. Discrete transitions are associated with time delay d_j, while continuous associate maximal firing speeds $V_j = 1/d_j$. m_0 represents nets initial marking, that is positive integers or null for D-places while for C-places initial marking is positive real numbers or null. Marking and speed vector define the state of a *Continuous Constant Speed PN*. In *HPNs* sets *I* and *O* must meet the following criterion: if p_i and t_j are a discrete place and a continuous transition respectively, then $I(p_i,t_j) = O(p_i,t_j)$ must be verified.

A HPN marking is: $m(t) = m_0 + W \bullet (n(t) + \int_{u=0}^{t} v(u) \bullet du)$,

where the first term in the parentheses corresponds to D-transitions and the second to C-transitions. *W* is the incidence matrix, m_0 the initial marking, *n(t)* represents the number of times each D-transition has been fired between initial time and time *t* and *v(t)* is the instantaneous firing speeds associated with C-transitions at time *t*. This equation represents a trajectory in the marking space [1]-[3].

The following symbols are used to represent a *HPN*: *Continuous places* are drawn with double circles (◎), *discrete places* as simple circles (○), *continuous transitions* with double bars (▯) and *discrete transitions* as simple bars. *Immediate transitions* are black (▮) while *timed* are empty bars (▯). In discrete places, *tokens* are small black circles in places, while for continuous places only their number is shown.

HPN functioning is altered by 3 kinds of events: firing of a discrete transition, emptying of a continuous place, and marking of a continuous place which is input of a discrete transition, reaches value of the arc weight between place and transition. By locating these types of events a *HPN* remains an event driven model, although it contains continuous functioning [3], [12].

The state of a *HTPN* is defined by markings of its discrete and continuous elements and continuous part speeds. A characteristic quantity of *HTPN* functioning is an invariant behavior state (IB-state).

B. Generic Hybrid Petri net modules

HTPN models corresponding to Figure 1 modules, called *generic HTPN modules*, are shown in Figures 2-4. Tokens are for demonstration purposes. Arc weights are set 1. Models follow common principles concerning their continuous and discrete parts. Discrete parts in given time intervals redefine type of manufactured product with respect to predefined criteria, while continuous parts describe processes followed in the production phase. Discrete states are "configurations" of the process in a qualitative model, and change the control policy. Discrete parts of the product types processed in a machine are connected, to be possible to redefine the type of product manufactured. Change of produced piece may demand change of machine settings as well as of the tools used.

The transportation of pieces to the machines as well as the machine setups after change of product type are not considered negligible and are represented by timed transitions or continuous transitions of given speed.

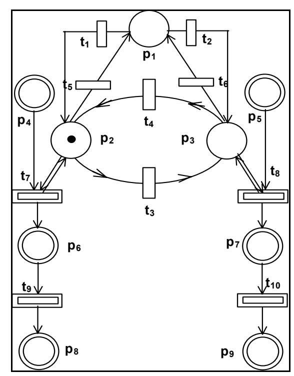

Figure 2: Generic multi-productive machine HTPN model.

In multi productive machine model, discrete part is structured so that a machine cannot produce 2 types of pieces concurrently. For this reason, on the discrete part there is 1 token and all weights are 1 to preserve tokens. In the discrete part there is a place representing machine breakdown that is connected with places representing both types of pieces produced. When machine is repaired, the piece produced is not the same as before, so the place of machine out of order leads through timed transitions to both product types. In this part of the net there is conflict that is solved with respect to strategies followed.

P_d	p_1	Machine out of order
	p_2	Machine setup for type 1 products
	p_3	Machine setup for type 2 products
P_c	p_4	Initial Buffer 1
	p_5	Initial Buffer 2
	p_6	Type 1 pieces at machine
	p_7	Type 2 pieces at machine
	p_8	Final buffer 1
	p_9	Final buffer 2

T_d	t_1	Breakdown repair & type 1 parts production
	t_2	Breakdown repair & type 2 parts production
	t_3	Change of pieces type lead to the machine from 1 to 2
	t_4	Change of pieces type lead to the machine from 2 to 1
	t_5	Breakdown while producing type 1 part
	t_6	Breakdown while producing type 2 part
T_c	t_7	Supply type 1 pieces at machine
	t_8	Supply type 2 pieces at machine
	t_9	Process type 1 pieces
	t_{10}	Process type 2 pieces

Table 1: Multi-productive machine module node explanation.

In generic multi-assembly module, sum of tokens in the discrete part is 2 (2 types of pieces form 1 final), while capacities of places corresponding to parts participation in the assembly is 1, as the assembly of 2 parts of the same type is not possible. The place representing machine breakdown has capacity 2, as both discrete tokens are led there when a breakdown occurs. Generic multi assembly model is shown in Figure 3. In this, 3 types of raw materials exist, 2 of which are used in each assembly. 3 assemblies are possible and 3 final products are obtained.

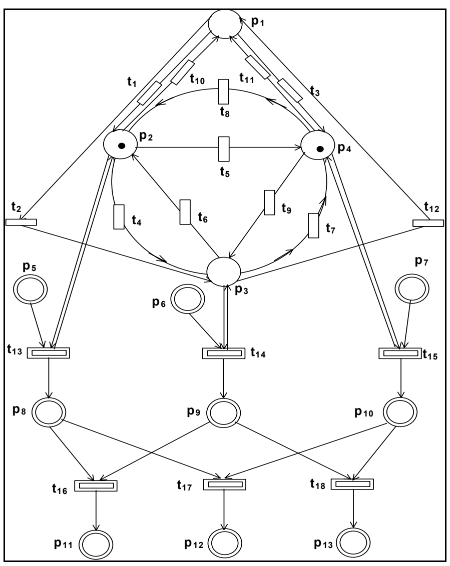

Figure 3: Generic multi- assembly machine module HTPN model.

P_d	p_1	Machine out of order
	p_2	Machine setup for type 1 products
	p_3	Machine setup for type 2 products
	p_4	Machine setup for type 3 products
P_c	p_5	Initial Buffer 1
	p_6	Initial Buffer 2
	p_7	Initial Buffer 3
	p_8	Type 1 pieces at machine
	p_9	Type 2 pieces at machine
	p_{10}	Type 3 pieces at machine
	p_{11}	Final buffer 1 (type 1-2 products)
	p_{12}	Final buffer 2 (type 1-3 products)
	p_{13}	Final buffer 3 (type 2-3 products)
T_d	t_1	Breakdown repair and type 1 pieces supply
	t_2	Breakdown repair and type 2 pieces supply
	t_3	Breakdown repair and type 3 pieces supply
	t_4	Change of pieces type activated for lead to machine from 1 to 2
	t_5	Change of pieces type activated for lead to machine from 1 to 3
	t_6	Change of pieces type activated for lead to machine from 2 to 1
	t_7	Change of pieces type activated for lead to machine from 2 to 3
	t_8	Change of pieces type activated for lead to machine from 3 to 1
	t_9	Change of pieces type activated for lead to machine from 3 to 2
	t_{10}	Machine breakdown while producing part where piece of type 1 participates
	t_{11}	Machine breakdown while producing type where piece of type 2 participates
	t_{12}	Machine breakdown while producing type where piece of type 2 participates
T_c	t_{13}	Supply type 1 pieces at machine
	t_{14}	Supply type 2 pieces at machine
	t_{15}	Supply type 3 pieces at machine
	t_{16}	Implement assembly 1-2
	t_{17}	Implement assembly 1-3
	t_{18}	Implement assembly 2-3

Table 2:Multi-assembly machine module nodes explanation.

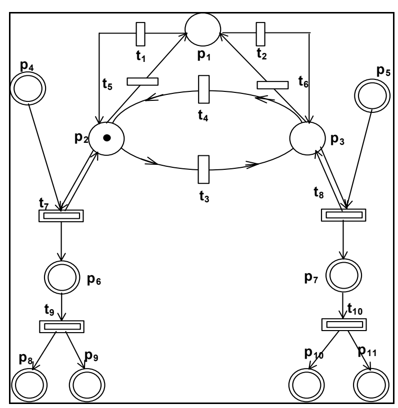

Figure 4:Generic multi-disassembly machine module HTPN model.

Generic disassembly module almost is similar with multi-productive machine model. The main difference is that each part is split into 2 new types of parts; each led to a different final buffer. In generic disassembly module, the number of tokens in the discrete part remains 1, while discrete places capacities are 1. Generic multi disassembly model is shown in Figure 4.

The meaning of multi assembly module nodes is for the majority common with the meanings of the nodes of multi-productive machine module of Figure 2. Differences and completions are shown in Table 3.

P_d	p_8	Final buffer 1-1 (1st piece type produced from disassembly 1)
	p_9	Final buffer 1-2 (2nd piece type produced from disassembly 1)
	p_{10}	Final buffer 2-1 (1st piece type produced from disassembly 2)
	p_{11}	Final buffer 2-2 (2nd piece type produced from disassembly 2)
T_c	t_9	Do disassembly 1
	t_{10}	Do disassembly 2

Table 3: Multi-disassembly machine module node explanation.

C. HTPN Modules Analysis

Considering generic *PN* modules with any finite initial marking $\boldsymbol{m_0}$, one may conclude that: *i)* In the discrete part of the generic modules there are conflicts arising from the fact that machines are not dedicated. Conflicts are solved during simulation or priorities may be defined. There is also a conflict between machine breakdown and pieces production that is solved during net's operation, as a breakdown appearance has the highest priority. *ii)* Generic modules are partially live, as in their discrete part no deadlock happens, since total number of tokens remains constant. Continuous parts are not live and remain live as long as there are pieces in initial buffers; *iii)* modules are ***k- bounded*** (absence of self-loops in combination with arc weights being 1 ensure *k-boundedness*); *iv)* multi-productive machine is conservative (multi assembly uses 2 pieces for the production of 1, multi-disassembly produces 2 pieces from 1 initial); *v)* modules are non-persistent (due to conflicts); *vi)* modules are not repetitive and not consistent since there are not repetitive transitions sequences whose firing results in the initial marking.

Upper limits of tokens found in parallel in net places are defined with respect to m_0 and places capacities.

D. Invariants calculation

All generic HTPN-models of fundamental modules have no T-invariants since no periodical functioning of the net takes place. Generic multi-productive machine model has 3 P-invariants. These are:

$\boldsymbol{m(p_1)+m(p_2)+m(p_3)=1}$

$\boldsymbol{m(p_4)+m(p_6)+m(p_8)=k_1,\ k_1= m_0(p_4)+m_0(p_6)+m_0(p_8)}$

$\boldsymbol{m(p_5)+m(p_7)+m(p_9)=k_2,\ k_2= m_0(p_5)+m_0(p_7)+m_0(p_9)}$

The first *P-invariant* shows 3 mutually exclusive machine states (machine processes type 1 parts, type 2 parts, or there is machine breakdown), while the other 2 guarantee that the sum of parts in the initial, final buffer and in the machine for each product type is constant and equal to the respective initial sum $\boldsymbol{k_i}$, i=1,2 in these buffers and in machine. The first P-invariant refers to the discrete part of the model, while the remaining 2 to the continuous.

The invariant set for multi-assembly machine module consists of the following 4 P-invariants:

$\boldsymbol{m(p_1)+m(p_2)+m(p_3)+m(p_4)=2}$ (*p_1* has capacity 2, *p_2, p_3 and p_4* have capacity 1)

$\boldsymbol{m(p_5)+m(p_8)+m(p_{11})+m(p_{12})=k_1}$

$\boldsymbol{m(p_6)+m(p_9)+m(p_{11})+m(p_{13})=k_2}$

$\boldsymbol{m(p_7)+m(p_{10})+m(p_{12})+m(p_{13})=k_3}$

$\boldsymbol{k_i}$, i=1-3 is the initial sum of tokens to the set of places that compose each P-invariant. The first *P-invariant* shows 4 mutually exclusive machine states (machine performs 1-2 assembly, 1-3 assembly, 2-3 assembly, or there is machine breakdown) while the rest 3 ensure that the sum of parts in the initial, final buffers and in machine for each product is constant and equal to the initial. The first P-invariant refers to the discrete part of the model, while the remaining 3 to the continuous.

The P-invariants of the generic multi-disassembly module *HPN* model are the following 5:

$\boldsymbol{m(p_1)+m(p_2)+m(p_3)=1}$

$\boldsymbol{m(p_4)+m(p_6)+m(p_8)=k_1,\ k_1= m_0(p_4)+m_0(p_6)+m_0(p_8)}$

$\boldsymbol{m(p_4)+m(p_6)+m(p_9)=k_2,\ k_2= m_0(p_4)+m_0(p_6)+m_0(p_9)}$

$\boldsymbol{m(p_5)+m(p_7)+m(p_{10})=k_3,\ k_3= m_0(p_5)+m_0(p_7)+m_0(p_{10})}$

$\boldsymbol{m(p_5)+m(p_7)+m(p_{11})=k_3,\ k_4= m_0(p_5)+m_0(p_7)+m_0(p_{11})}$

The first *P-invariant* is identical to the respective of multi-productive machine and shows 3 mutually exclusive machine states (machine performs type 1 or type 2 disassembly, or there is breakdown). The rest 4 P-invariants guarantee that the sum of parts in the initial, final buffer and in the machine for each product type is constant and equal to the initial sum of parts in these buffers ($\boldsymbol{k_i}$ i=1-4) and in machine. The first P-invariant refers to the discrete part of the model, while the rest to the continuous. The continuous P-invariants are by 2 relative, as only final buffer changes.

IV. HTPN MODULE SYNTHESIS

Generic modules synthesis procedure is presented through a real example shown in Figure 5. The case presented is simple but generalizations are provided.

A system consisting of 2 multi-productive machines is considered. In this, 2 products are manufactured, each receiving 2 operations, 1 in each machine. Type 1 pieces visit machine 1 and then machine 2, while type 2 parts follow the converse route. From Figure 5, it is obvious that 2 place fusions occur in combined net's model. Places p_5 and p_{18} are fused in place $p_{5\text{-}18}$ while places p_8 and p_{13} form place $p_{8\text{-}13}$. Continuous places are reduced by 2, while transitions are equal to the total of each module transitions. The combined PN input places are reduced by 2 (fused places are not external). Maximum capacities of fused places are defined with respect to the capacities of the places from which they arise. Obviously, $m_0(p_{5\text{-}18})=m_0(p_5)+m_0(p_{18})$.

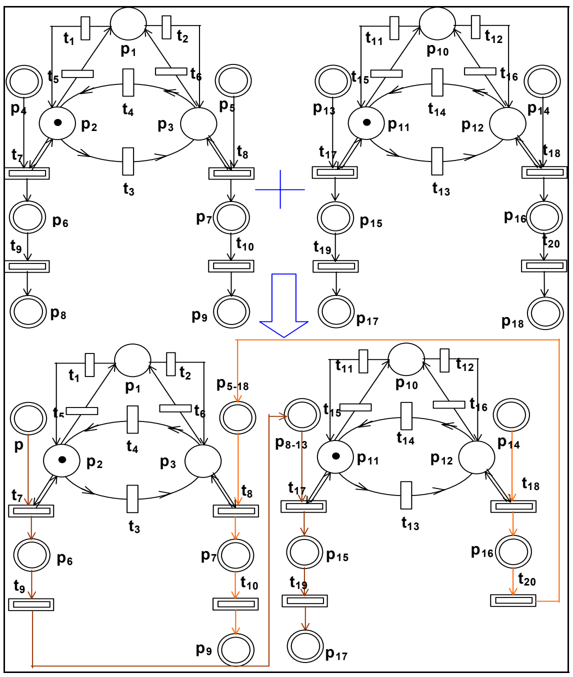

Figure 5: Generic multi-productive machine modules synthesis.

Properties of the combined HTPN are calculated. 4 P-invariants exist, 2 of which refer to the mutually exclusive machine states $\boldsymbol{(m(p_1)+m(p_2)+m(p_3)=1}$ and $\boldsymbol{m(p_{10})+m(p_{11})+m(p_{12})=1)}$. The remaining refer to the preservation of total number of parts in the HTPN and are $\boldsymbol{m(p_4)+m(p_6)+m(p_{8\text{-}13})+m(p_{15})+m(p_{17})=n_1}$, where $\mathbf{n_1}$ the initial sum of tokens in p_4, p_6, p_8, p_{13}, p_{15} and p_{17} and $\boldsymbol{m(p_{14})+m(p_{16})+m(p_{5\text{-}18})+m(p_7)+m(p_9)=n_2}$ where $\mathbf{n_2}$ the initial sum of parts in p_{14}, p_{16}, p_{18}, p_5, p_7 and p_9. Synthesis of other generic PN modules is obtained in a similar way but due to space limitations are omitted.

A. Generalizations

The number of nodes of a random topology multi-productive system model is calculated from the modules that compose it and output buffers. The latter number is used in the computation of number of fused places. Table 4 shows the complexity of generic modules and their generalization for more complicated cases. $\boldsymbol{s_j}$ refers to the number of initial pieces taking part in a assembly in a machine j to form final part and $\boldsymbol{d_k}$ refers to the number of part types produced from initial in disassembly module k. In multi-assembly and multi-disassembly it is considered that in all processes in a machine, the same numbers of initial or final parts participate. Also in generalized multi-assembly all assemblies between raw materials are considered valid.

Model	Nodes type	Generic module	Generalized module (n-components)
Multi productive machine	P (C)	6	$3*n$
	P (D)	3	$n+1$
	T (C)	4	$2*n$
	T (D)	6	$n*(n+1)$
Multi-assembly	P (C)	9	$2*n+\frac{n!}{s_j!\left(n-s_j\right)!}$
	P (D)	4	n+1
	T (C)	6	$n+\frac{n!}{s_j!\left(n-s_j\right)!}$
	T (D)	12	$n*(n+1)$
Multi disassembly	P (C)	8	$(2+d_k)*n$
	P (D)	3	$n+1$
	T (C)	4	$2*n$
	T (D)	6	$n*(n+1)$

Table 4: HTPN modules complexity for each node type.

A production system HTPN model is derived in terms of the generic HTPN modules; that is n_1 modules of multi-productive machine, n_2 modules of multi-assembly, n_3 modules of multi-disassembly, n_4 input places and n_5 output places. HTPN model consists of $10*(n_1+n_3)+18*n_2$ transitions. Total number of places when considered separately is $9*n_1+13*n_2+11*n_3$. Places fusion at connection points reduces this number by $0.5*(2*n_1+3*n_2+4*n_3–n_5)$ 0.5*(number of modules outputs – external outputs). So, total number of multi-operational production systems places is $8*n_1+9*n_3+11.5*n_2+0.5*n_5$.

Considering individual generalized HTPN modules as shown in Table 4, total transitions number is $\sum_{i=1}^{n_1} l_i*(l_i+3)+\sum_{j=1}^{n_2}\left[l_j*(l_j+2)+\frac{l_j!}{s_j!\left(l_j-s_j\right)!}\right]+\sum_{k=1}^{n_3} l_k*(l_k+3)$ where l_i refers to the number of products produced by multi-productive machine i, l_j refers to the raw materials used by multi-assembly machine j (s_j in each) and l_k to the number of part types produced by multi-disassembly machine k. Total number of places is:

$$\sum_{i=1}^{n_1}(4*l_i+1)+\sum_{j=1}^{n_2}\left[3*l_j+1+\frac{l_j!}{s_j!\left(l_j-s_j\right)!}\right]+\sum_{k=1}^{n_3}\left[l_k*(3+d_k)+1\right]$$
$$-\frac{1}{2}\left(2*\sum_{i=1}^{n_1} l_i+\sum_{j=1}^{n_2}\left(l_j+\frac{l_j!}{s_j!\left(l_j-s_j\right)!}\right)+\left(\sum_{k=1}^{n_3} l_k*(1+d_k)\right)-(n_4+n_5)\right)=$$
$$=\sum_{i=1}^{n_1}(3*l_i+1)+\sum_{j=1}^{n_2}\left(\frac{5}{2}*l_j+1+\frac{l_j!}{2*s_j!\left(l_j-s_j\right)!}\right)+\sum_{k=1}^{n_3}\left(\frac{l_k}{2}(5+d_k)+1\right)+\frac{n_4+n_5}{2}$$

V. A CASE STUDY

Production system of Figure 6 with its HTPN model of Figure 7 is presented as case study. It consists of 4 machines and 13 buffers, (3 initial, 3 final, 7 internal) and produces 3 types of final products that follow 2 independent routes (products 2 and 3 follow the same route as they are produced from the same type of piece by disassembly in M_1) and could be modeled by different systems. Each machine spends a different portion of it's time for the production of each product (each machine performs 2 processes). As it is obvious from Figure 7, 4 modules compose total production systems model (in Figure 7 arcs forming the same route have the same color). These are 2 multi-productive machines, 1 multi-assembly and 1 multi-disassembly module. In multi-assembly the 2 assemblies do not have a common part and all the other assemblies that theoretically could be performed do not have practical meaning. Parts enter system through initial buffers 1, 2 and 3 (places p_4, p_{26} and p_{27}), while parts reaching final buffers 11, 12 and 13 (places p_{10}, p_{11} and p_{40}) are final.

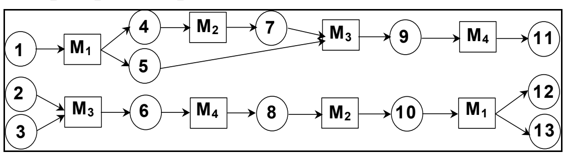

Figure 6: The multi-operational production system of case study.

Total HTPN system model consists of **33** places (sum of modules places–fused places = 40-7) and **40** transitions. Final model has 10 P-invariants, 4 referring to machines mutually exclusive states and the rest to parts preservation (these invariants are composed by individual modules invariants by replacing places taking part in fusion with the fused and merging the 2 invariants in which common place participate).

$m(p_1)+m(p_2)+m(p_3)=1$

$m(p_{12})+m(p_{13})+m(p_{14})=1$

$m(p_{21})+m(p_{22})+m(p_{23})=1$

$m(p_{32})+m(p_{33})+m(p_{34})=1$

$m(p_4)+m(p_6)+m(p_{8\text{-}24})+ m(p_{28})+ m(p_{30\text{-}36})+ m(p_{38})+ m(p_{40})=k_1$

$m(p_4)+m(p_6)+m(p_{9\text{-}15})+ m(p_{17}) + m(p_{19\text{-}25})+m(p_{28})+ m(p_{30\text{-}36})+ m(p_{38})+ m(p_{40})=k_2$

$m(p_{26})+m(p_{29})+m(p_{31\text{-}35})+ m(p_{37}) + m(p_{16\text{-}39})+m(p_{18})+ m(p_{5\text{-}20})+ m(p_7)+ m(p_{10})=k_3$

$m(p_{26})+m(p_{29})+m(p_{31\text{-}35})+ m(p_{37}) + m(p_{16\text{-}39})+m(p_{18})+ m(p_{5\text{-}20})+ m(p_7)+ m(p_{11})=k_4$

$m(p_{27})+m(p_{29})+m(p_{31\text{-}35})+ m(p_{37}) + m(p_{16\text{-}39})+m(p_{18})+ m(p_{5\text{-}20})+ m(p_7)+ m(p_{10})=k_5$

$m(p_{27})+m(p_{29})+m(p_{31\text{-}35})+ m(p_{37}) + m(p_{16\text{-}39})+m(p_{18})+ m(p_{5\text{-}20})+ m(p_7)+ m(p_{11})=k_6$

The properties of the multi-operational production system HTPN model are the same with the respective ones of fundamental modules from which it arises. So, considering HTPN with any finite initial marking, net is partially live, ***k- bounded***, not conservative, non-persistent, not repetitive and not consistent.

For initial marking $m_0=\{0, 1, 0, 20, 1, 0, 0, 2, 1, 0, 1, 0, 0, 1, 2, 0, 0, 0, 0, 1, 14, 15, 0, 0, 1, 2, 0, 1, 0, 0, 1, 0\}$, system's performance is studied through simulation (fused places are represented by the first component of their name e.g. place's $p_{5\text{-}20}$ marking is the 5^{th}). Simulations are performed using Visual Object Net [16] and help in optimization of system's performance through minimizing machines idleness, maximizing total throughput and optimizing buffer capacities.

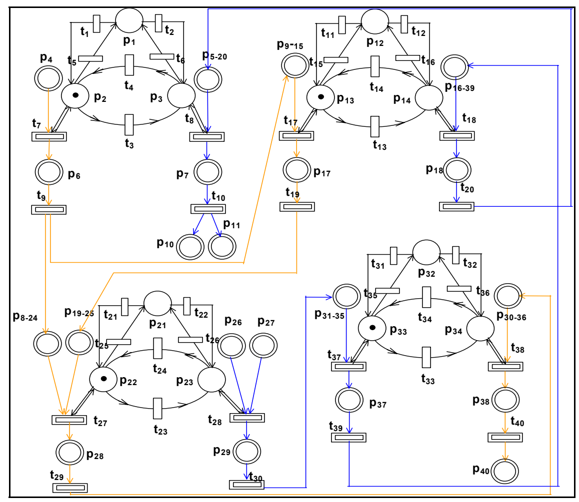

Figure 7: Total System HTPN model.

Except the initial marking, the firing speeds of nets continuous and discrete transitions are defined. All continuous transitions speeds are considered constant, while the delays associated to some discrete transitions change according to nets state and also priorities are defined (especially for conflicts).

Continuous transitions firing speeds: $\{t_7, t_8, t_9, t_{10}, t_{17}, t_{18}, t_{19}, t_{20}, t_{27}, t_{28}, t_{29}, t_{30}, t_{37}, t_{38}, t_{39}, t_{40}\}=\{2, 3, 2.5, 1, 3, 2, 1.5, 1.75, 4, 3, 1.67, 2.5, 2, 3, 3, 1.5\}$. Discrete transitions that refer to machine breakdown or repair have constant delays $\{t_1, t_2, t_5, t_6, t_{11}, t_{12}, t_{15}, t_{16}, t_{21}, t_{22}, t_{25}, t_{26}, t_{31}, t_{32}, t_{35}, t_{36}\}=\{5, 7, 6, 8, 6, 8, 5, 6, 5, 7, 7, 6, 4, 6, 5, 8\}$, while delays of transitions referring to change of produced part type are not constant and are functions of the number of tokens in the input buffers places to such transitions (they are described from $a*p_i$, where a is a constant real number between 0 and 3).

In this case the simulation is terminated after 40.3 time units. The final marking of the net by the end of simulation is $m_f=\{0, 0, 1, 0, 0, 0, 0, 0, 0, 19, 20, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 24\}$ which means that 19, 20 and 24 final products of the 3 types have been produced. Figure 8 shows the levels of the 7 internal net buffers during the simulation.

From Figure 8 it is obvious that none internal buffer has concurrently more than 14 pieces ($p_{8\text{-}24}$, $p_{16\text{-}39}$ and $p_{19\text{-}25}$ reach 12 pieces while $p_{5\text{-}20}$ and $p_{9\text{-}15}$ have at maximum 7 pieces). This is utilized in buffer capacities optimization or for non-effective net components finding. Another interesting point is what is the result from changes of characteristic nets quantities (machine speeds, machine breakdowns) in overall net's behavior.

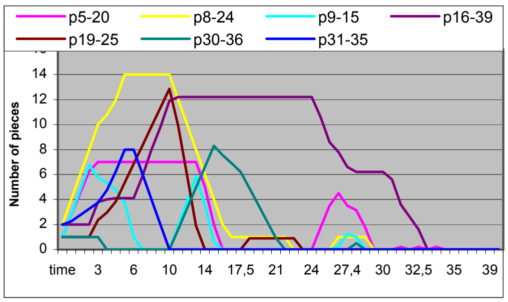

Figure 8: Internal buffer levels during simulation.

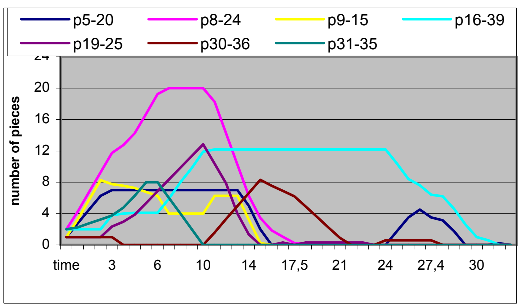

Figure 9: Buffer levels after change of speeds of t_7 and t_{10}.

By doubling the speed of t_{10} (by using a machine with increased productivity) and by repeating the simulation with all other parameters constant, the same final products are produced in 34.7 time units. There is a reduction of the total time about 15%. The buffer levels do not change from the respective of Figure 8, since only the speed of last machine (for type 2 and 3 products) has been changed. If the speed of t_7 changes from 2 to 3, simulation is terminated after 32.4 time units (extra reduction 7%). New buffer levels during simulation are shown on Figure 9. It is obvious that speeds increase results in a big increase of capacities especially for buffer $p_{8\text{-}24}$. This process may be repeated until the optimization of nets characteristics.

VI. CONCLUSIONS

Modular HTPNs have been used for modelling, analysis and synthesis of random topology multi-operational production networks. 3 generic modules have been considered, their corresponding generic HTPN modules have been derived and their properties have been extracted. Generalizations have been provided and expressions for the number of system's discrete and continuous nodes for random configuration systems have been calculated. P-invariants provide further insight to production systems study and behavior. Simulation results demonstrate the effectiveness of the proposed method in optimizing such systems and finding non-effective components.

VII. ACKNOWLEDGMENTS

This work has been partially supported by a grant from the GSRT / EU number 2000ΣE 01330005.

VIII. REFERENCES

[1] David R., Alla H., On hybrid Petri Nets, *Discrete Event Dynamic Systems: Theory and Applications*, vol. 11, pp. 9 – 40, 2001, Kluwer Academic Publishers.

[2] Allam M., Alla H., Modeling and Simulation of an Electronic Component Manufacturing System Using Hybrid Petri Nets, *IEEE Transactions on Semiconductor Manufacturing*, Vol. 11, No 3., pp. 374-383, August 1998.

[3] David R., Alla H., *Petri Nets & Grafcet – Tools for modeling discrete event systems*, Prentice Hall, 1992

[4] J. Le Bail, H. Alla and R. David, "Hybrid Petri Nets", in *PRoc. Eur. Control Conference*, Grenoble, July 1991, pp. 1472-1477

[5] Alla. H., David R., A modelling and analysis tool for discrete events systems: continuous Petri net, *Performance Evaluation*, vol. 33, pp. 175-199, 1998.

[6] Zerhouni N., Ferney M., El Moudni A., Transient analysis of manufacturing systems using continuous Petri Nets, *Mathematics and Computers in Simulation*, vol. 39, pp. 635-639, 1995.

[7] Di Febbraro A., Giua A., Menga G., Guest Editorial for the Special Issue on Hybrid Petri Nets, *Discrete Event Dynamic Systems*, Vol 11, No. 1&2, 2001.

[8] Duridanova V., Hummel T., Modelling of Embedded Mechatronic Systems Using HPNs, *IASTED International Conference on Modeling, Identification and Control,* 2001, Innsbruck, IASTED / ACTA Press 2001, vol. 1, pp. 521 –526.

[9] Drath R., Engmann U., Schwuchow, Hybrid Aspects of Modeling Manufacturing Systems using Modified PNs, *5th Workshop on Intelligent Manufacturing Systems*, Brazil, 1999.

[10] G. J. Tsinarakis, K. P. Valavanis, N. C. Tsourveloudis, Modular Petri Net Based Modeling, Analysis and Synthesis of Dedicated Production Systems, accepted *ICRA 2003*, Taiwan.

[11] Christensen S., Petrucci L., Modular Analysis of Petri Nets, *The Computer Journal*, Vol. 43, No. 3, pp. 224 – 242, 2000.

[12] Sava A. T., Alla H., Combining Hybrid Petri Nets and Hybrid Automata, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 5, pp. 670 – 678, October 2001.

[13] Tsourveloudis N., Dretoulakis E., Ioannidis S., Fuzzy work-in-process inventory control of unreliable manufacturing systems, *Information Sciences*, Vol. 27, pp. 69 – 83, 2000.

[14] Ioannidis S., Tsourveloudis N., Valavanis K., Fuzzy Supervisory Control of Manufacturing Systems, Accepted to *IEEE Transactions on Robotics and Automation*.

[15] S. Ioannidis, N. C. Tsourveloudis and K. P. Valavanis, Supervisory Control of Multiple-Part-Type Production Networks, *10th IEEE Mediterranean Conference on Control and Automation*, Lisbon, Portugal, July 2002.

[16] Drath R., *Visual Object Net ++*, Visual Object oriented Petri Net based Engineering Tool, Version 2.7a, Paramsoft Software Development, http://www.paramsoft.de/, 2002.