Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Modular Hybrid Petri Nets for Studying Multi-operational Production Systems Where Parts Follow Multiple Alternative Processes *

G. J. Tsinarakis
Intelligent Systems and Robotics Laboratory
Department of Production Engineering and Management
Technical University of Crete
Chania, Greece, GR – 73 100
tsinar@dpem.tuc.gr

K. P. Valavanis
Department of Computer Science and Engineering
University of South Florida
Center for Robot-Assisted Search and Rescue
4201 E. Fowler Avenue, Tampa, FL 33620-5399
kvalavan@csee.usf.edu

Abstract **- This paper presents a Modular Hybrid Timed Petri Net (MHTPN) methodology used to study multi-operational production systems in which parts are used for the production of different product types. This research generalizes work presented in [1] that studied production systems where all parts of the same type were used for the production of a type of final products following a common route. 3 fundamental system modules are derived; their MHTPN models are defined, followed by modules synthesis to obtain the overall system Petri net. For any topology, system's nodes and invariants are calculated. Applicability of the method is illustrated through an analytical example.**

Index Terms - Multi-operational production systems, hybrid systems, modular timed hybrid PNs, complexity analysis.

I. INTRODUCTION

Analysis and optimization issues of realistic scale discrete event dynamic systems (DEDS) are in the majority of cases computationally intractable. To overcome such issues, fluid models that are continuous dynamics approximations of DEDS have been developed and applied [2] [3]. In such a system, called a hybrid system [4], [6], the behavior of interest is determined by interacting continuous and discrete dynamics; both behaviors are complementary and essential to deriving the system model, called hybrid model.

Hybrid Petri Nets (HPNs) are suitable to model and study hybrid systems [5]. HPNs do have all the advantages of Petri Net models [6]; they do not require exhaustive enumeration of all reachable states and they may describe systems with infinite state space [8]. They allow modular system representation, leading to reduction of overall model's complexity [7].

HPNs are defined by combining an ordinary PN with a continuous PN (CPN) [9]. In CPNs tokens represent a real quantity of token fragments. Hybrid timed PNs (HTPNs) consisting of a timed PN and a Constant Speed Continuous PN (CCPN) comprise the most popular tool for applications, since they may be used for system simulation and performance evaluation. CCPNs stem from timed PNs with maximal speed functioning [6]. The state of a HTPN is defined by its discrete and continuous elements markings and continuous part speeds.

This paper studies multi-operational production systems in which incoming or in-process parts are used for the production of different product types - that is parts may be used in alternative processes. Three fundamental modules are derived and their corresponding MHTPN models are defined. The overall system Petri net model is obtained through synthesis of these modules. For any possible system topology, the number of system nodes and invariants is calculated. An analytical example is used to illustrate the applicability of the proposed method.

The main contribution of this paper is that it generalizes completely the discussed MHTPN methodology making it applicable to all types of multi-operational production systems regardless of structure, architecture and complexity. Further, analysis and synthesis of any such system is derived in terms of analysis and synthesis of three basic modules followed by node complexity and P-invariant calculation.

II. MULTI-OPERATIONAL PRODUCTION SYSTEMS GENERALIZED MODULES.

Three fundamental modules, generalizations of those presented in [11][13], are derived for any multi-operational production. They are illustrated in Figure 1; circles represent buffers and rectangles represent machines. These modules appropriately connected together model production systems of random topology and complexity. Bold arcs represent parts transfer from buffers to machines and/or vice versa.

In the generalized multi-productive machine module there is one input and one output bold arcs. Machine M_i receives parts from one of the n_{pi} upstream buffers Bj_ni and after processing them sends products to the corresponding downstream buffer Bil_n. Multi-productive machine module performs n_{pi} different types of processes, where $n_{p_i} \geq 2$.

Generalized multi-assembly module refers a n_{Ai1} input buffers-one machine-n_{Ai2} output buffers module that performs n_{Ai2} types of s_i parts assemblies. All assemblies consist of the same number of initial parts and $n_{A_{i1}} \geq 3$, since each assembly is performed at least between two parts, and

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

machine performs multiple processes. There are s_i input and ***one*** output bold arcs.

In the generalized multi-disassembly, machine receives parts from 1 of the n_{Di1} input buffers ($n_{D_{i1}} \geq 2$), separates them and sends them to d_i from n_{Di2} output buffers, $n_{D_{i_2}} = \sum_{i_2=1}^{n_{D_{i_1}}} d_{ii_2}$.

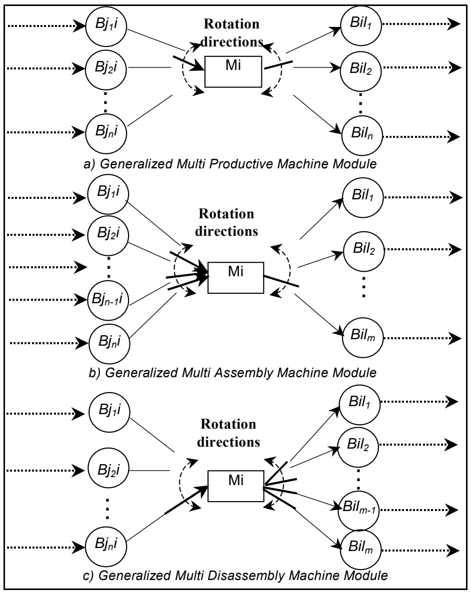

Fig. 1 Multi-operational production systems generalized modules

III. Hybrid Petri net Fundamentals

HTPNs are defined in [5], [9]. A generalized timed HPN is described by the 7-tuple $\boldsymbol{H = \{P, T, I, O, h, \tau, m_0\}}$ [8]. P is a finite set of places partitioned in subsets of continuous P_c and discrete places P_d such that $P_c \bigcup P_d = P$ and $P_c \bigcap P_d = \varnothing$. T is a set of transitions, partitioned in subsets T_c and T_d. $T_c \bigcup T_d = T$, $T_c \bigcap T_d = \varnothing$ and $P \bigcap T = \varnothing$. 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 $t_j \in T_c$ and for all $p_i \in P_d$, $I(p_i,t_j)=O(p_i,t_j)$. This states that an arc joining a ***C-*** transition to a ***D-***place demands the existence of the reverse arc ensuring marking of ***D-***places to remain integer. $h: P \bigcup T \rightarrow \{D,C\}$ is hybrid function that indicates for each place or transition if it is ***D-*** or ***C-***. $\tau: T \rightarrow R^+$ associates each transition with a positive real. ***D-***transitions are associated with time delay d_j, while ***C-***transitions with maximal firing speed $V_j = 1/d_j$. m_0 represents nets initial token distribution (marking), which is positive integers or 0 for ***D-***places and reals or 0 for ***C-***places.

An HTPN marking m at time t is deduced from initial m_0 due to a firing sequence of transitions S, using the relation $m(t) = m_0 + W * (n(t) + \int_{u=0}^{t} v(u) * du)$, where W is the incidence matrix, $n(t)$ the vector of ***D-***transition firings between initial time and t, and $v(t)$ the instantaneous firing speeds associated with ***C-***transitions at time t. The first term in the parentheses corresponds to ***D-***transitions and the second to ***C-***transitions.

If there is a conflict between a ***D-*** and a ***C-*** transition, ***D-***transition has priority over the ***C-***. In case of conflict between continuous transitions with a common empty ***C-*** input place, any solution satisfying *"sum of instantaneous firing speeds of transitions feeding the place minus the sum of instantaneous firing speeds of transitions emptying the place is equal to 0"* is admissible. When common input place is ***D-*** and contains a token, any solution such that $\sum_{j=1}^{\alpha} \frac{v_j}{V_j} = 1$ is admissible.

A vector X of R^{n_p} is a ***P-***invariant if $X^T * W = 0$. A vector Y of R^{n_t} is a ***T-***invariant if $W * Y = 0$. ***P-***invariants describe token conservation in sets of places for all reachable markings. ***T-*** invariants represent a necessary condition for a periodical functioning of a HTPN. Most HTPN properties are the same with the respective OPN properties, adapted to accommodate continuous variables features.

In HTPNs ***C-***places are represented as double circles (◎), ***D-***places as circles (○), ***C-***transitions as double boxes (▯) and ***D-***transitions as boxes. Black boxes represent *immediate* transitions (▮) and white *timed* ones (▯). In ***D-***places *tokens* are small black dots, while for ***C-***places the real number of tokens in each place is shown.

The HTPN behavior remains event driven, although it contains a continuous functioning [9]. HTPN functioning is changes by the occurrence of three kinds of events. *i)* ***D-****transition* firing; *ii)* Marking of a ***C-****place* becomes 0; *iii)* Marking of a ***C-****place* that is input to a ***D-****transition* reaches the weight of the arc linking the place to the transition.

IV. Generalized HTPN Modules

Generalized MHTPN modules of Figure 1 are shown in Figures 2-4. Tokens are for demonstration purposes and all arc weights are 1. Models follow common principles governing their *C-* and *D-* parts. *C-places* describe resource availability; *D-places* correspond to discrete system states; *D-transitions* describe state changes, *C-transitions* correspond to continuous events speeds (machine operations). *D-places* representing product types processed in a machine are connected in order to be possible to redefine the type of manufactured product. Models *D*-parts comprise a type of "controller" of the process performed in a machine, while *C*-parts describe production processes (parts transport from buffers to machines, process in machine). Change of the part type processed in a machine may demand change of machine settings andas of the used tools.

Modules are derived based on realistic assumptions: *i)* all buffers are finite and each one hosts one part type, *ii)* machines operate at given speeds periodically redefined according to the net status, *iii)* machine breakdowns happen randomly, *iv)* the process performed in a machine changes regularly according to specific criteria after the selection of machine setup, *v)* breakdowns interrupt immediately the performed process.

Multi productive machine module describes a machine performing n_{Pi} types of processes, each corresponding to a different incoming part type. Net's discrete part consists of $n_{Pi}+1$ places, one for each process type and one for machine breakdown (p_1). It has one token, defining the machine status and is constructed so that it is not possible for machine to produce concurrently multiple product types from different raw materials. p_1 is connected through timed transitions to other ***D***-places both sides, as breakdowns happen in whatever performed process and after repair of machine, the performed process is not the same as before. After the breakdown repair there is a structural conflict as all part types may theoretically be produced, which is resolved with respect to system quantitative features (e.g. parts in the buffers) and priorities.

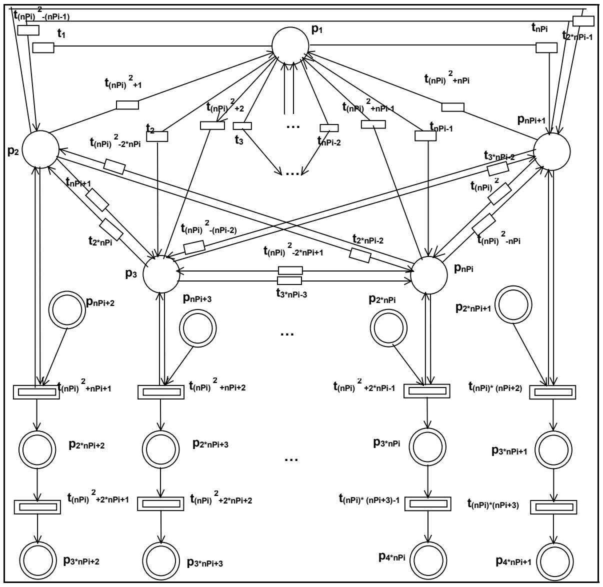

Fig. 2 Generalized multi-productive machine HTPN model.

TABLE I
MULTI-PRODUCTIVE MACHINE NODES EXPLANATION

P_d	p_1	Machine out of order
	$p_2 - p_{n_{P_i}+1}$	Machine setup for type $1 - n_{p_i}$ parts
P_c	$p_{n_{P_i}+2} - p_{2n_{P_i}+1}$	Buffers of type $1 - n_{p_i}$ initial parts
	$p_{2n_{P_i}+2} - p_{3n_{P_i}+1}$	Type $1 - n_{p_i}$ parts at machine
	$p_{3n_{P_i}+2} - p_{4n_{P_i}+1}$	Final buffers for type $1 - n_{p_i}$ products
T_d	$t_1 - t_{n_{P_i}}$	Breakdown repair and type $1 - n_{p_i}$ products production
	$t_{n_{P_i}+1} - t_{(n_{P_i})^2}$	Change of produced product type (all combinations)
	$t_{(n_{P_i})^2+1} - t_{(n_{P_i})^2+n_{P_i}}$	Machine breakdown while processing $1 - n_{p_i}$ type parts
T_c	$t_{(n_{P_i})^2+n_{P_i}+1} - t_{n_{P_i}(n_{P_i}+2)}$	Supply at machine type $1 - n_{p_i}$ parts
	$t_{(n_{P_i})^2+2n_{P_i}+1} - t_{n_{P_i}(n_{P_i}+3)}$	Process type $1 - n_{p_i}$ parts respectively

The multi-assembly machine module describes a machine that performs different types of s_i-part assemblies with $s_i \geq 2$ (all assemblies in a module consist of the same number of parts). n_{Ai1} initial part types are available ($n_{A_{i1}} \geq 3$). Not all initial part combinations form a valid assembly. For this reason, part types resulting in non-valid assemblies are never supplied together in a machine. Transitions representing non-valid assemblies implementation have maximal firing speed 0, meaning that no token flow can happen in this part of the model. Final buffers of non-valid assemblies are excluded, as no products are found there. Generalized multi-assembly machine module has $n_{A_{i2}} = \frac{n_{A_{i1}}!}{s_i!\left(n_{A_{i1}} - s_i\right)!} - q_i$, final buffers where q_i refers to the number of non-valid initial product combinations. All *D*-place capacities representing machine setups are 1, since multiple parts of the same type do not participate in an assembly. p_1 representing machine breakdown has capacity s_i since all *D*-part tokens are led there in case of breakdown. The sum of tokens in the discrete part remains s_i during operation.

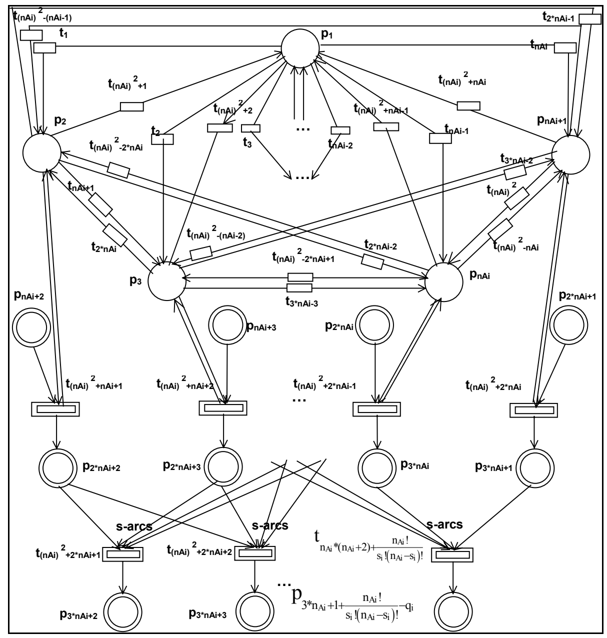

Fig. 3 Generalized multi-assembly machine HTPN module.

TABLE II
MULTI-ASSEMBLY MACHINE MODULE NODES EXPLANATION

P_d	p_1	Machine out of order
	$p_2 - p_{n_{A_{i1}}+1}$	Machine setup for type $1 - n_{A_{i1}}$ parts respectively
P_c	$p_{n_{A_{i1}}+2} - p_{2n_{A_{i1}}+1}$	Buffers of type $1 - n_{A_{i1}}$ initial parts respectively
	$p_{2n_{A_{i1}}+2} - p_{3n_{A_{i1}}+1}$	Type $1 - n_{A_{i1}}$ parts at machine
	$p_{3n_{A_{i1}}+2} - p_{3n_{Ai}+1+n_{A_{i2}}}$	Final buffers for $1 - n_{A_{i2}}$ products respectively
T_d	$t_1 - t_{n_{A_{i1}}}$	Breakdown repair & type $1 - n_{A_{i1}}$ parts supply
	$t_{n_{A_{i1}}+1} - t_{(n_{A_{i1}})^2}$	Change of part type activated for lead to machine (all possible combinations)
	$t_{(n_{A_{i1}})^2+1} - t_{(n_{A_{i1}})^2+n_{A_{i1}}}$	Machine breakdown while producing a product where $1 - n_{A_{i1}}$ part participates
T_c	$t_{(n_{A_{i1}})^2+n_{A_{i1}}+1} - t_{n_{A_{i1}}(n_{A_{i1}+2})}$	Supply at machine type $1 - n_{A_{i1}}$ parts respectively
	$t_{(n_{A_{i1}})^2+2n_{A_{i1}}+1} - t_{n_{A_{i1}}*(n_{A_{i1}}+2)+\frac{n_{A_{i1}}!}{s_i!(n_{A_{i1}}-s_i)!}}$	Process type $1 - n_{A_{i1}}$ parts respectively

The generalized multi-disassembly module is similar to the corresponding generalized multi-productive machine module. The main difference is that each of the n_{Di1} processes performed in the machine ($n_{D_{i1}} \geq 2$), produces d_{ij} different product types ($d_{ij} \geq 2, j = 1,...,n_{D_{i1}}$). For each product a different output buffer exists. The number of modules output buffers is $n_{D_{i2}} = \sum_{i_2=1}^{n_{D_{i1}}} d_{ii_2}$. The discrete part of the module has 1 token and all the discrete place capacities are equal to 1.

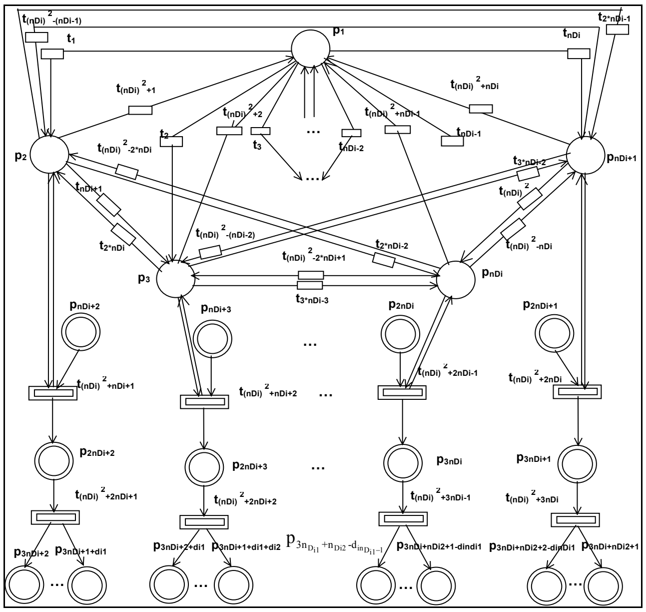

Fig. 4 Generalized multi-disassembly machine HTPN module

TABLE III
MULTI-DISASSEMBLY MODULE NODES EXPLANATION

P_d	p_1	Machine out of order
	$p_2 - p_{n_{D_{i1}}+1}$	Machine setup for type $1 - n_{D_{i1}}$ parts
P_c	$p_{n_{D_{i1}}+2} - p_{2n_{D_{i1}}+1}$	Buffers of type $1 - n_{D_{i1}}$ initial parts
	$p_{2n_{D_{i1}}+2} - p_{3n_{D_{i1}}+1}$	Type $1 - n_{D_{i1}}$ parts at machine
	$p_{3n_{D_{i1}}+2} - p_{3n_{D_{i1}}+n_{D_{i2}}+1}$	Final buffers for type $1 - n_{D_{i2}}$ products
T_d	$t_1 - t_{n_{D_{i1}}}$	Breakdown repair & type $1 - n_{D_{i1}}$ disassembly performed
	$t_{n_{D_{i1}}+1} - t_{(n_{D_{i1}})^2}$	Change of part type lead to machine (all combinations)
	$t_{(n_{D_{i1}})^2+1} - t_{(n_{D_{i1}})^2+n_{D_{i1}}}$	Machine breakdown while processing $1 - n_{D_{i1}}$ type parts
T_c	$t_{(n_{D_{i1}})^2+n_{D_{i1}}+1} - t_{n_{D_{i1}}(n_{D_{i1}}+2)}$	Supply type $1 - n_{D_{i1}}$ parts at machine
	$t_{(n_{D_{i1}})^2+2n_{D_{i1}}+1} - t_{n_{D_{i1}}(n_{D_{i1}}+3)}$	Implement disassembly $1 - n_{D_{i1}}$

V. GENERALIZED HTPN MODULES ANALYSIS

A. Properties Analysis

Considering HTPN modules with finite initial marking m_0, the following conclusions are stated: *i)* There exist conflicts in models *D*-parts since machines are multi-productive. Conflicts are solved by assigning priorities. In conflict between machine breakdown and process, breakdown has the highest priority. *ii)* As long as there are parts in the input buffers, operations keep on until a breakdown occurs. *iii)* HTPN models are partially live. No deadlock occurs in their *D*-part. Modules *C*-parts remain live as long as parts in initial buffers exist. Parts in initial buffers define the duration that *C*-part remains live. So HTPN modules are partially live; *iv)* modules are *k- bounded* (no self-loops and arc weights being 1 ensure that); *v)* Multi-productive machine is conservative. Multi-assembly uses s_i parts for one final product, while multi-disassembly produces d_{ij} products from 1 initial part; *vi)* Modules are non-persistent; *vii)* Token preservation and machine mutually exclusive states are described by P-invariants; *viii)* Modules are not repetitive and not consistent. Thus, no T-invariants exist.

Upper limit of tokens found in *C*- places is defined with respect to initial markings $m_0(p)$ and place capacities C_i.

B. P-invariants Calculation

Generalized multi-productive machine module has $(n_{Pi}+1)$ *P-invariants*. The first one refers to the mutually exclusive machine states. The rest refer to the preservation of tokens parts within the system, where k_i, $i=1,...,n_{Pi}$, is the initial sum of tokens in the respective set of places. The *P-invariants* are:

$$m(p_1)+m(p_2)+\ldots+m(p_{nPi})+m(p_{nPi+1})=1$$
$$m(p_{nPi+2})+m(p_{2nPi+2})+m(p_{3nPi+2})=k_1$$
$$\vdots$$
$$m(p_{2nPi})+m(p_{3nPi})+m(p_{4nPi})=k_{nPi-1}$$
$$m(p_{2nPi+1})+m(p_{3nPi+1})+m(p_{4nP+1})=k_{nPi}$$

Generalized multi-assembly machine module has $(n_{Ai}+1)$ *P-invariants*. One refers to mutually exclusive states and the other n_{Ai} to token preservation within the system. k_i, $i=1,...,n_{Ai}$ is the initial sum of tokens in the respective places set. Each of the n_{Ai} invariants consists of a place representing initial buffer i, a place representing type i parts entering the machine and places representing all buffers where final products composed of i parts may be found. Maximum number of such places is $c_{1i} = \frac{n_{A_i}!}{s!(n_{A_i}-s_i)!} * \frac{s_i}{n_{A_i}} = \frac{(n_{A_i}-1)!}{(s_i-1)!(n_{A_i}-s_i)!}$, where $\frac{n_{Ai}!}{s_i!(n_{Ai}-s_i)!} = c_{2i}$ are all possible combinations, multiplied by s_i parts participating in an assembly, divided by n_{Ai} raw materials and the respective minimum number is the same reduced by q_i.

Letting $m(p_{3n_{Ai}+2}) = A_1,...,m(p_{3n_{Ai}+1+c_{2i}}) = A_{c_{2i}}$ $\quad q_i = 0$, the *P-invariants* of multi-assembly machine module are:

$$m(p_1)+m(p_2)+\ldots+m(p_{nAi})+m(p_{nAi+1})=s_i$$
$$m(p_{nA_i+2})+m(p_{2nA_{i1}+2})+A_1+A_2+\ldots+A_{c_{1i}-1}+A_{c_{1i}}=k_1$$
$$m(p_{nA_i+3})+m(p_{2nA_i+3})+A_2+A_3+\ldots+A_{c_{1i}+1}=k_2$$
$$\vdots$$
$$m(p_{2nA_i+1})+m(p_{3nA_i+1})+A_{c_{2i}-c_{1i}}+\ldots+A_{c_{2i}-1}+A_{c_{2i}}=k_{nA_i}$$

Generalized multi-disassembly has $(n_{D_{i2}}+1)$ *P-invariants*. The first refers to mutually exclusive machine states and the rest to token preservation within the system with k_i $i=1,...,n_{Di2}$ the initial sum of tokens in the respective places set. Each d_{ij} *P–invariants* refer to the same initial buffer and part types in machine with different only the final buffer. *P-invariants* are:

$$m(p_1)+m(p_2)+\ldots+m(p_{nD1i})+m(p_{nDi1+1})=1$$

$$m(p_{nDi1+2})+m(p_{2nDi1+2})+m(p_{3nDi1+2})=k_1$$

$$\vdots$$

$$m(p_{nDi1+2})+m(p_{2nDi1+2})+m(p_{3nDi1+1+di1})=k_{di1}$$

$$\vdots$$

$$m(p_{2nDi1+1})+m(p_{3nDi1+1})+m(p_{3nDi1+nDi2+2-dinDi1})=k_{nDi2-dinDi1+1}$$

$$\vdots$$

$$m(p_{2nDi1+1})+m(p_{3nDi1+1})+m(p_{3nDi1+nDi2})=k_{nDi2}$$

VI. MHTPN MODULE CONNECTION

As stated, the proposed method is suitable for studying multi-operational production systems where initial or in process part types are not "dedicated". Such parts can be used for the production of different products that follow a partially or totally different route in the system (visit different machines or receive processes in a different sequence). This has major impact in calculating net's nodes and P-invariants complexity.

TABLE IV
MULTI-DISASSEMBLY MODULE NODES EXPLANATION

Model	Nodes type	Nodes number
Multi-productive machine	DP	$n_{Pi}+1$
	DT	$n_{Pi}(n_{Pi}+1)$
	CP	$3(n_{Pi})$
	CT	$2(n_{Pi})$
Multi-Assembly	DP	$n_{Ai1}+1$
	DT	$n_{Ai1}(n_{Ai1}+1)$
	CP	$2n_{Ai1}+n_{Ai2}$
	CT	$n_{Ai1}+\dfrac{n_{Ai1}!}{(n_{Ai1}-s_i)!\,s_i!}$
Multi-Disassembly	DP	$n_{Di1}+1$
	DT	$n_{Di1}(n_{Di1}+1)$
	CP	$2n_{Di1}+n_{Di2}$
	CT	$2n_{Di1}$

An example of a multi-productive machine performing 3 processes types is considered before generalizations. One raw materials type exist, that all receive the first process and then either the second or the third one. Alternative routes for 2[nd] and 3[rd] process are shown with blue and red arcs.

Total net properties are the same with the respective of the HTPN modules properties. It's places number is reduced in comparison to the sum of modules places due to places fusion (-2 from the expected). HTPN model has 2 P-invariants from which one refers to mutually exclusive states ($\boldsymbol{m(p_1)+m(p_2)+m(p_3)+m(p_4)=1}$) and the other to tokens preservation ($\boldsymbol{m(p_5)+m(p_6)+m(p_9)+m(p_7)+m(p_{10})+m(p_8)+m(p_{11})=k_1}$).

Nodes complexity of a system build from HTPN modules is derived in terms of their components. A system composed of $\boldsymbol{n_1}$ multi-productive machines, $\boldsymbol{n_2}$ multi-assemblies, $\boldsymbol{n_3}$ multi-disassemblies and $\boldsymbol{n_4}$ input places, has $\boldsymbol{n_T}$ transitions:

$$n_T=\sum_{i=1}^{n_1} n_{P_i}(n_{P_i}+3)+\sum_{j=1}^{n_2}\left[n_{A_{j1}}(n_{A_{j1}}+2)+\frac{n_{A_{j1}}!}{s_j!\left(n_{A_{j1}}-s_j\right)!}\right]+\sum_{k=1}^{n_3} n_{D_k}(n_{D_k}+3).$$

Total places number is reduced in comparison to the sum of individual modules places, as places fusion takes place in modules connection points. In these, 2 or more places (representing buffers) are fused and form a new place (such is $\boldsymbol{p_9}$ in the test case. $\boldsymbol{p_9}$ is produced by fusion of 3 places, representing the 1[st] process output buffer and the input buffers of 2[nd] and 3[rd] process. Sum of individual modules places is: $\sum_{i=1}^{n_1}(4n_{P_i}+1)+\sum_{j=1}^{n_2}\left[3n_{A_{j1}}+1+n_{A_{j2}}\right]+\sum_{k=1}^{n_3}\left[3n_{D_{i1}}+n_{D_{i2}}+1\right]$. Fusion reduces this number by $\sum_{i=1}^{n_1}n_{P_i}+\sum_{j=1}^{n_2}n_{A_{j1}}+\sum_{k=1}^{n_3}n_{D_{k1}}-n_4$. So, multi-operational production systems have $\boldsymbol{n_P}$ places:

$$n_P=\sum_{i=1}^{n_1}(3n_{P_i}+1)+\sum_{j=1}^{n_2}\left[2n_{A_{j1}}+n_{A_{j2}}+1\right]+\sum_{k=1}^{n_3}\left[2n_{D_{i1}}+n_{D_{i2}}+1\right]+n_4.$$

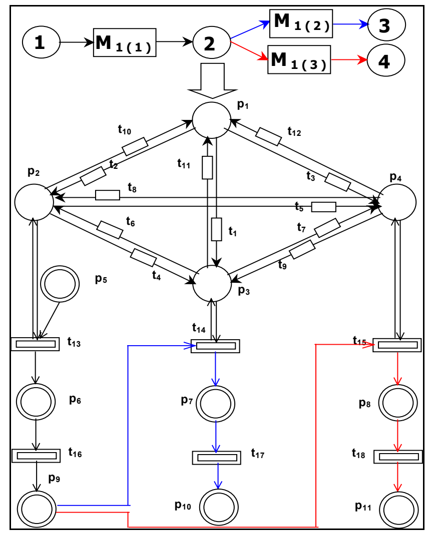

Fig. 5 HTPN model of system where products follow alternative routes.

Considering a multi-operational production system, its *P*-invariants number can theoretically be calculated as function of modules forming total net features, combined with net's topology. As it is obvious from the test case, *P*-invariants are classified in two main types: one referring to the mutually exclusive machine states and the other to parts preservation within the system. $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$ respectively represent the two invariant types. So, overall *P*-invariants number is $\boldsymbol{d_1+d_2}$. $\boldsymbol{d_1}$ is equal to net's machines (as each machines mutually exclusive states are presented in the same invariant) and is: $\boldsymbol{d_1=n_1+n_2+n_3}$.

$\boldsymbol{d_2}$ computation is complicated and not exact in all cases. A general equation for the upper limit of $\boldsymbol{d_2}$, valid for all possible cases is calculated. For specific multi-operational production systems subclasses $\boldsymbol{d_2}$ can be derived accurately.

Disassembly processes are the majors responsible for the generation of multiple invariants, as from one input several outputs (also *P*-invariants) are produced. Using a machine for multiple processes does not affect token preservation, since at each time period at most one part type is processed. Parts participating in alternative processes affect *P*-invariants number. Especially when disassemblies follow alternative processes, all possible buffer combinations of the alternative processes must be considered.

First of all, the general equation for $\boldsymbol{d_2}$ holding in all cases as upper limit is calculated. So $\boldsymbol{d_2}$ is:

$$d_2 \leq \sum_{i=1}^{n_4}\left[\prod_{j=1}^{a_1}\left[n_{D_{i,j,1}} - \sum_{i_1=1}^{o_1} n_{A_{1,i_1}} + \sum_{i_1=1}^{o_1} n_{A_{1,i_1}} * E_{1,i_1}\right]\right], \text{ where}$$

$$E_{1,i_1} = \prod_{j_2=1}^{a_{2,i_1}}\left[n_{D_{i_1,j_2,2}} - \sum_{i_{2,i_1}=1}^{o_{2,1}} n_{A_{2,i_{2,i_1}}} + \sum_{i_{2,i_1}=1}^{o_{2,1}} n_{A_{2,i_{2,i_1}}} * E_{2,i_{2,i_1}}\right]$$

Factors $E_{2,i_{2,i_1}}$ and respective factors of next level are provided by similar equations. In d_2 calculation, for each of the n_4 initial part types, processes are sequentially considered. Initially all alternative processes are considered ($\prod_{j=1}^{a_1}$). In net, for each part type, two sequential disassemblies define a level, where the first defines level's starting point and the second ending point, being at the same time next levels starting point. Then, number of first disassembly received ($n_{Di,j,,1}$) products is considered and from this the number of first level disassembly products participating in assemblies of the same level is subtracted ($\sum_{i_1=1}^{o_1} n_{A_{1,i_1}}$). If in a level some parts do not participate in an assembly but receive process in multi-productive machine, they are considered (in the equation) as participating in 1-part assemblies. In the calculated result, for every first level assembly, the product of number of input to assembly parts ($n_{A1,,j1}$) and respective factor $E_{1,j1}$, $j_1=1,\ldots o_1$ is added. $E_{1,,i1}$ refers to the already described part of the equation, adapted to second level features *(product of 2nd level alternative processes disassembly - 2nd level disassembly products participating in 2nd level assemblies + product of number of input to 2nd level assembly parts ($n_{A2,k}$) and the respective factor $E_{2,i2i1}$. $E_{2,j}$ is received in a respective way for 3rd level quantities and this process is repeated for as many levels as defined by each products process sequence disassemblies)*. If a part does not participate in a disassembly, $n_{Di,1}$ is 1. In d_2 equation, the equality is valid for production systems in which no assembly takes place. If assemblies exist, $<$ holds as some invariants are calculated multiple times in the equation. In a multi-operational production system with no alternative processes systems invariants number is:

$$d_2 = \sum_{i=1}^{n_4}\left[n_{D_{i,1}} - \sum_{i_1=1}^{o_1} n_{A_{1,i_1}} + \sum_{i_1=1}^{o_1} n_{A_{1,i_1}} * E_{1,i_1}\right], \text{ where}$$

$$E_{1,i_1} = \left[n_{D_{i,2,i_1}} - \sum_{i_{2,i_1}=1}^{o_{2,1}} n_{A_{2,i_{2,i_1}}} + n_{A_{2,i_{2,i_1}}} * E_{2,i_{2,i_1}}\right]$$

In fact, this equation is the already described general equation with absence of ($\prod$) due to alternative routes.

VII. Method Application

Practical value of the method is better understood through its application in a test case production system presented in Figure 6. Systems model consists of 4 machines and 17 buffers, 3 initial, 5 final and 9 internal. 5 products that follow independent routes through the system are produced (products 2 & 3 are produced by disassembly in M_1, similar for 4 & 5). Each machine performs at minimum 2 and at maximum 3 types of processes. Blocks of the same color represent all the processes performed in a machine. 4 appropriately connected modules form system's MHTPN model. These are a multi-disassembly that performs 3 types of processes, 2 multi-productive machine modules each performing 3 process types and a multi-assembly that performs 2 types of assembly. Parts reaching buffer 6 may follow 2 alternative routes that lead to the production of different products (pairs of products 2 & 3 and 4 & 5 respectively). For each part type, the arcs forming it's route into the system have the same color through the net with only exception the alternative processes, where arcs leading to buffer 6 are blue, while the ones leaving from there are red and pink. In multi-assembly the 2 assemblies do not have common parts and every other assembly that theoretically could be performed does not have practical meaning. Buffers 1, 2 and 3 (places p_5, p_{20} and p_{21}) are system entrances, while buffers 13 - 17 (places p_{11}-p_{14} and p_{45}) contain products.

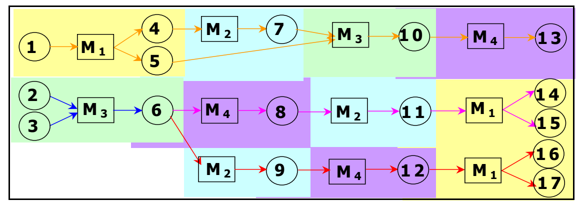

Fig. 6 Case study multi-operational production system.

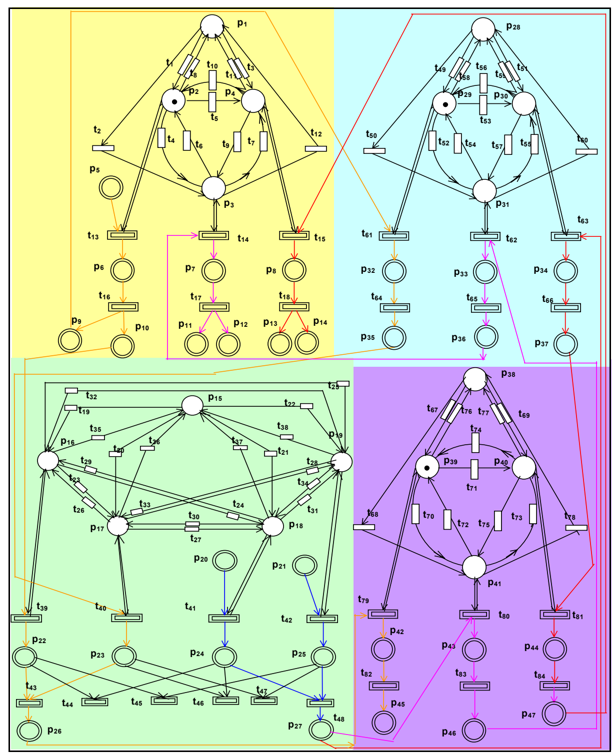

Fig. 7 Overall system MHTPN model.

Overall systems MHTPN model consists of *47* places and *84* transitions and has *14* P-invariants. 4 refer to mutually exclusive states: $m(p_1)+m(p_2)+m(p_3)+m(p_4)=1$, $m(p_{15})+m(p_{16})+m(p_{17})+m(p_{18})+m(p_{19})=2$, $m(p_{28})+m(p_{29})+m(p_{30})+m(p_{31})=1$ and $m(p_{38})+m(p_{39})+m(p_{40})+m(p_{41})= 1$. 10 P-invariants refer to parts preservation: $m(p_5)+m(p_6)+m(p_9)+m(p_{32})+m(p_{35})+m(p_{23})+m(p_{26})+m(p_{42})+m(p_{45})=k_1$, $m(p_5)+m(p_6)+m(p_{10})+m(p_{22})+m(p_{26})+m(p_{42})+m(p_{45})=k_2$, $m(p_{20})+m(p_{24})+m(p_{27})+m(p_{43})+m(p_{46})+m(p_{33})+m(p_{36})+m(p_7)+m(p_{11})=k_3$, $m(p_{20})+m(p_{24})+$

$m(p_{27})+m(p_{43})+m(p_{46})+m(p_{33})+m(p_{36})+m(p_7)+m(p_{12})=k_4$, $m(p_{20})+m(p_{24})+m(p_{27})+m(p_{34})+m(p_{37})+m(p_{44})+m(p_{47})+m(p_8)+m(p_{13})=k_5$, $m(p_{20})+m(p_{24})+m(p_{27})+m(p_{34})+m(p_{37})+m(p_{44})+m(p_{47})+m(p_8)+m(p_{14})=k_6$, $m(p_{21})+m(p_{25})+m(p_{27})+m(p_{43})+m(p_{46})+m(p_{33})+m(p_{36})+m(p_7)+m(p_{11})=k_7$, $m(p_{21})+m(p_{25})+m(p_{27})+m(p_{43})+m(p_{46})+m(p_{33})+m(p_{36})+m(p_7)+m(p_{12})=k_8$, $m(p_{21})+m(p_{25})+m(p_{27})+m(p_{34})+m(p_{37})+m(p_{44})+m(p_{47})+m(p_8)+m(p_{13})=k_9$, $m(p_{21})+m(p_{25})+m(p_{27})+m(p_{34})+m(p_{37})+m(p_{44})+m(p_{47})+m(p_8)+m(p_{14})=k_{10}$, where k_i, $i=1,10$ is the initial sum of tokens in the respective set of places.

Overall model's properties arise from the respective properties of the fundamental modules from which it consists. So, for any finite $\boldsymbol{m_0}$, net is *partially live, k- bounded, not conservative, non-persistent, not repetitive and not consistent.*

Net's initial marking has: $m(p_5)=21$, $mp(_{20})=20$, $m(p_{21})=22$, 1 token in the discrete part of each multi-productive machine and multi-disassembly module and 2 tokens in the discrete part of multi-assembly. Simulations are performed with Visual Object Net [14] and are used to optimize systems performance through optimization of measures as total production times, etc.

C-transitions firing speeds and *D*-transitions delays are defined. *D*-transitions: breakdown repair: t_1-$t_3=4$, t_{19}-$t_{22}=3$, t_{49}-$t_{51}=5$ and t_{67}-$t_{69}=2$; breakdown: t_{10}-$t_{12}=3$, t_{35}-$t_{38}=5$, t_{58}-$t_{60}=3$ and t_{58}-$t_{60}=4$; change of produced product: functions of tokens in buffers – omitted due to space limitations. *C*-transitions: $t_{13}=4$, $t_{14}= t_{15}=3$, $t_{16}= t_{17}=2$, $t_{18}=3$, t_{39}-$t_{42}=2$, $t_{43}=4$, t_{44}- $t_{47}=0$, $t_{48}=2$, $t_{61}=4$, $t_{62}= t_{65}=3$, $t_{66}=2$, $t_{79}=2$, t_{80}-$t_{83}=3$, and $t_{84}=2$.

With the parameters as described, simulation is terminated after *78* time units. By the end of the simulation final buffers p_{11} and p_{12} contain *7* products, p_{13} and p_{14} *13* products and p_{45} *21* products. Internal buffer levels are presented in Figure 8.

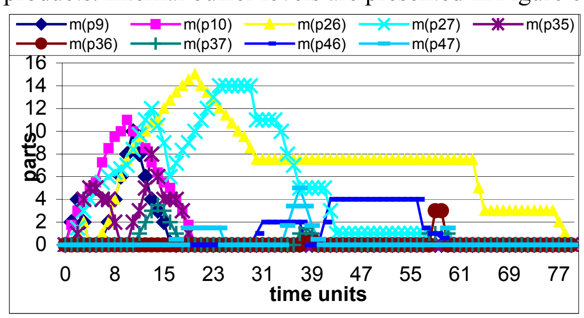

Fig. 8 Internal buffer levels during initial simulation.

From Figure 8 it is obvious that the maximum number of parts found in a buffer is 15 instantly found in $\boldsymbol{p_{26}}$, while in $\boldsymbol{p_{27}}$ for a small time period 14 parts are found and in $\boldsymbol{p_{10}}$ 11. These results can be used for buffer capacities optimization. Another fact is the impact of changing net's operational features in its overall behavior. By doubling the speed of $\boldsymbol{t_{79}}$ (from 2 to 4) and by keeping all other net parameters constant, simulation is terminated after 62 time units (reduction of net's operational time 20%). Buffer levels during second simulation are shown in Figure 9. From this it is obvious that none buffer has more than 13 pieces ($\boldsymbol{p_{26}}$) concurrently. This process may be continued until the optimization of all desirable net features.

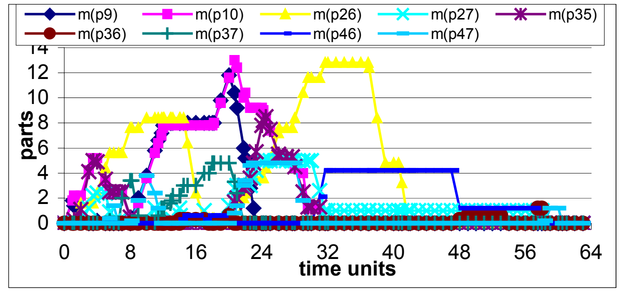

Fig. 9 Internal buffer levels during second simulation.

VII. Conclusions

A framework for study of multi-operational production systems with use of MHTPNs is proposed. 3 fundamental modules and their corresponding HTPN models are considered and their properties are studied. Complicated systems models are built from these basic blocks. Expressions for overall system's places and transitions numbers as well as *P*-invariants are calculated. Simulations ensure the applicability of the method in optimizing systems behavior and quantities.

References

[1] Tsinarakis G. J., Valavanis K. P., Tsourveloudis N. C., "Studying Multi-operational Production Systems with Modular Hybrid Petri Nets", IEEE Int. Conf. on Intelligent Robots and Systems - IROS 2003.

[2] Balduzzi F., Giua A., Seatzu C., "Modeling and Simulation of Manufacturing Systems with First-Order Hybrid Petri Nets", Int. Journal of Production Research, Special Issue on Modeling, Specification and Analysis of Manufacturing Systems, vol. 39, no. 2, 2001.

[3] Pettersson S., Lennartson B., "Hybrid Modelling focused on Hybrid Petri Nets", Proceedings 2nd European Workshop on Real-time and Hybrid Systems, Grenoble, France, pp. 303 – 309, 1995.

[4] Antsaklis P., "Scanning the Issue / Technology", Proceedings of the IEEE, Special Issue on Hybrid Systems: Theory and Applications, A brief introduction to the Theory and Applications of Hybrid Systems, vol. 88, no. 7, pp. 879-887, July 2000.

[5] David R., Alla H., "On Hybrid Petri Nets", Discrete Event Dynamic Systems: Theory and Applications, vol. 11, pp. 9 – 40, 2001.

[6] Sava A. T., Alla H., "Combining Hybrid Petri Nets and Hybrid Automata", IEEE Trans.actions on Robotics and Automation, vol. 17, no. 5, pp. 670 – 678, October 2001.

[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] Allam M., Alla H., "Modeling and simulation of an electronic component manufacturing system using Hybrid Petri nets", IEEE Trans. on Semiconductor Manufacturing, vol. 11, no 3., pp. 374-383, 1998.

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

[10] Tsinarakis G., Valavanis K., Tsourveloudis N., "Modular Petri Net Based Modeling, Analysis and Synthesis of Dedicated Production Systems", IEEE Int. Conf. on Robotics and Automation - ICRA 2003.

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

[12] Ioannidis S., Tsourveloudis N., Valavanis K., "Fuzzy supervisory control of manufacturing systems", Accepted to IEEE Trans. on Robotics and Automation.

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

[14] Drath R., *Visual Object Net ++*, Version 2.7a, Paramsoft Software Development, http://www.paramsoft.de/, 2002