Proceedings of the 2005 IEEE
International Conference on Automation Science and Engineering
Edmonton, Canada, August 1 & 2, 2005

Studying Multi-assembly Machine Production Systems with Hybrid Timed Petri Nets

G. J Tsinarakis[1], N. C. Tsourveloudis [1], and K. P. Valavanis [2],
[1] *Department of Production Engineering and Management, Technical University of Crete, GR – 73 100, Chania, Greece*
[2]*Center for Robot-Assisted Search and Rescue, Department of Computer Science and Engineering, University of South Florida, 4201 E. Fowler Avenue Tampa, FL 33620-5399, USA*

Abstract— **This paper generalizes the use of the Hybrid Timed Petri Net method introduced in [1] to study multi-operational production systems in which incoming or in-process parts are used for different products production. In particular, the proposed multi-assembly machine module is generalized to describe a machine performing multiple assemblies consisting of different initial parts numbers. The applicability of the method is illustrated through an analytical example.**

I. INTRODUCTION

THE complexity growth of industrial systems requires accurate and complete models reflecting realistic studies and performance evaluation. Flaws in the modeling process contribute significantly to the development time and cost, affecting operational efficiency, too [2].

Discrete Event System models are described at an abstraction level where the time base is continuous, but during a time-span a finite number of events causing change of systems state occur. Between events, system's state is constant. Initially formalisms as state automata, ordinary Petri nets, finite state machines and event graphs were introduced, in which the order in which events occurred and not the explicit time was represented [3], [4]. Simulation, analysis and optimization of realistic scale Discrete Event Dynamic Systems (DEDS), whose reachable states number is typically large, requires large amount of computational efforts as problems become computationally and analytically intractable. Inclusion of time causes an increase in model's complexity, leading to reachable state space explosion [4].

To overcome such issues, fluid models that are continuous approximations of DES have been developed, where, model execution leads to continuously changing variables. Fluid models advantages are: enhanced computational efficiency, state space dimension reduction and that no significant errors are introduced in performance analysis via simulation. Continuous design parameters make possible the use of gradient information for sensitivity analysis and speed up optimization [5]. Composition of a continuous time system and a DES called hybrid, has shown to be more efficient [6].

In a hybrid system the behavior of interest is determined by interacting continuous and discrete dynamics. Both behaviors are complementary and essential to derive overall systems model, called hybrid model [7], [8]. Hybrid models are used when it is important to represent the evolutions of continuous variables, as well as the functional sequences of phases of the discrete part [10]. Hybrid systems are common in manufacturing, control, batch and chemical processes and transportation and embedded systems [7], [8].

Hybrid Petri Nets (HPNs) provide a unified environment for modeling, formal analysis, design and study of systems [11]. HPN models are used as a visual-communication aid, enabling to set-up mathematical models governing the behavior of systems and with tokens addition they allow simulating system's activities [12]. HPNs inherit all the advantages of OPN models, such as the ability to capture behaviors including concurrency, synchronization, mutual exclusion and precedence relations [8]. In addition, they do not require exhaustive enumeration of reachable states; they can finitely describe systems with infinite state space and allow modular system representation [13], [14].

Although, Petri nets are not a new modeling and analysis tool, they have been successfully used for the representation of complex and ill-defined notions, such as, routing and operation flexibilities [21], which are important for managing the change in today's distributed virtual enterprises.

This paper generalizes work presented in [1]. In this, a framework for study of multi-operational production systems with HTPNs is proposed. Multi-operational production systems are met as multi-class systems in [15] and are modeled using FOHPNs while in [16] machines of this type are referred as multi-product machines and are studied using semi-Markov models. Representative applications are met in automotive manufacturing companies, mainly to metal stamping and forming activities [16].

In [1] three basic modules (multi-productive machine, multi assembly and multi disassembly) and their HTPN models are introduced and studied. Multi-assembly module performs assemblies consisting of the same number of parts. Complicated systems models are built from the basic blocks and are analyzed according to the properties and features of the fundamental modules.

The main contribution of this paper is that it generalizes research presented in [1], [17]; it extends its use in all multi-

This work was partially supported from "Herakleitos" fellowships for research from the Technical University of Crete EPEAEK II 88727.

operational production systems types regardless of structure, architecture and complexity. The proposed method may be used in numerous modeling, performance evaluation, analysis and design problems and is of general use as it does not refer to a specific problem, but can be adapted to the features of any system following certain well-defined steps.

II. MULTI -ASSEMBLY PRODUCTION SYSTEM MODULE

Assembly receives more and more attention in industry, as it can count for a great deal of hidden cost, such as scrap and rework. Assembly includes basic physical actions, logistics, inspection, constraints on the generation of assembly plans, measurement of parts geometric features, transportation, inventory and a variety of support and life cycle activities.

An assembly model must be capable of capturing a diverse set of information needed to describe the entities and activities associated with assemblies, so that product designers, logistic systems, supplier relations and field support can have access to these information [18].

In this paper a generalized multi-assembly machine module is presented. The ($\boldsymbol{n_{Ai1}}$) input buffers-one machine-($\boldsymbol{n_{Ai2}}$) output buffers multi-assembly module obtains two or more parts from upstream buffers, assembles them to form a product and sends it to the respective downstream buffer $\boldsymbol{B_{ilm}}$. $\boldsymbol{n_{Ai1}}$ types of initial parts (raw materials) are present, and are assembled to different final products. Each part may participate in multiple different assembly types. At each time instant at most one process type is performed.

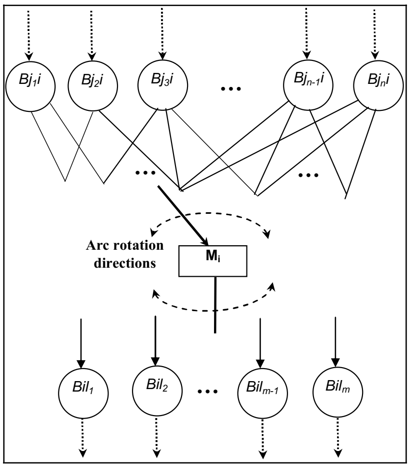

Fig. 1: Multi-assembly machine fundamental module

Multi-assembly machine module is shown in Fig. 1 (circles and rectangles represent buffers and machines). A module is a fundamental subsystem with a set of input and output arcs defining interactions with other modules and a set of internal discrete and continuous relations that define module's internal state. Bold rotating arcs demonstrate how unprocessed parts enter machines and how they are removed when processed. They rotate to connect to the appropriate fixed input–output buffer combinations. Rotating arcs indicate that machines are not dedicated but at given time periods produce different product types. Multi-assembly machine module when appropriately connected to multi-productive machine and multi-disassembly modules can be used to represent manufacturing networks of various layouts.

III. HYBRID PETRI NETS

The first step on HPNs definition was the introduction of Continuous Petri nets (CPNs) by Alla and David. CPNs, [19], arise from the respective timed PNs by "fluidification" of their integer marking vectors. Combination of an OPN and a CPN, results in a HPN. Analytical description of the main concepts regarding HPNs can be found in [19]-[20].

A generalized marked timed HPN is described by the 7-tuple $\boldsymbol{H = \{P, T, I, O, h, \tau, m_0\}}$. P is a non-empty set of places partitioned in subsets of continuous P_c and discrete places P_d such that $\boldsymbol{P_c \cup P_d = P}$, $\boldsymbol{P_c \cap P_d = \varnothing}$. T is the finite set of transitions, partitioned in subsets T_c and T_d. $\boldsymbol{T_c \cup T_d = T}$, $\boldsymbol{T_c \cap T_d = \varnothing}$. I and O, $\boldsymbol{I:\begin{cases}P_d \times T \rightarrow \mathbb{N}\\ P_c \times T \rightarrow \mathbb{R}_0^+\end{cases}}$, $\boldsymbol{O:\begin{cases}P_d \times T \rightarrow \mathbb{N}\\ P_c \times T \rightarrow \mathbb{R}_0^+\end{cases}}$ are the pre- and post- incidence mappings specifying arcs. The set of arcs A is partitioned into subsets of standard and inhibitor arcs. An inhibitor arc of weight r from a place p_i to a transition t_j allows the firing of t_j only if the marking of p_i is less than r. For all $\boldsymbol{t_j \in T_c}$, $\boldsymbol{p_i \in P_d}$ connected with standard arcs, $\boldsymbol{I(p_i,t_j) = O(p_i,t_j)}$ must be verified. This states that an arc joining a C-transition to a D-place demands the existence of the reciprocal arc ensuring marking of D-places to remain integer. $\boldsymbol{\tau: T \rightarrow \mathbb{R}^+}$ 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 initial token distribution, which are positive integers or 0 for D-places and real numbers or 0 for C-places. Input and output places of D-transitions can be continuous or discrete.

The marking at time t reached from m_0 after firing a transitions sequence s, is $\boldsymbol{m(t) = m_0 + W*(n(t) + \int_{u=0}^{t} v(u)*du)}$, where W is the incidence matrix, $n(t)$ is the vector of discrete transition firings between initial time and time t, and $v(t)$ is 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

Priorities are defined between continuous and discrete transitions for conflict cases. If there is a conflict between a D- and a C-transition, D-transition has priority. In case of conflict between continuous transitions with a common empty continuous 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 discrete and contains a token, any solution such that $\boldsymbol{\sum_{j=1}^{a} \frac{v_j}{V_j} = 1}$ is admissible.

A vector $\boldsymbol{X}$ of $\boldsymbol{R^{n_p}}$ is a $\boldsymbol{P}$-invariant if $\boldsymbol{X^T * W = 0}$. A vector $\boldsymbol{Y}$ of $\boldsymbol{R^{n_t}}$ is a $\boldsymbol{T}$-invariant if $\boldsymbol{W * Y = 0}$ [20]. $\boldsymbol{P}$-invariants express a notion of token conservation in sets of places for all reachable markings. T- invariants are a necessary condition for a periodical functioning of a HTPN.

The visual disjunction of HTPN components is necessary. So, in a HTPN C- places are drawn with double circles (◎), D-places as simple circles (○), C-transitions are represented with double bars (▯) and D-transitions as simple bars. Immediate transitions as black bars (▮) while timed as empty bars (▯). In discrete places, tokens are shown as small black circles, while for continuous places only the number of tokens "residing" in each place is shown. Standard arcs are drawn as usual ($\rightarrow$), while inhibitor arcs are represented by arcs whose end is marked with a small circle (——○).

The HTPN behavior remains event driven, although it contains a continuous functioning. HTPN functioning is changes by the occurrence of 3 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. Multi-Assembly HTPN Model

A. *Module fundamentals*

The following assumptions stand for multi-assembly module: *i)* buffers have finite capacities and are dedicated (one product type is found in each buffer), *ii)* machines operate at given speeds, redefined with respect to the events happening in the system, *iii)* machine breakdowns happen infinitely often, *iv)* machines change the type of process performed at certain time instants according to specific criteria after the selection of appropriate machine setup.

The parts transfer to machines and machine setup after change of a produced part type events are represented by timed transitions or continuous transitions with a given speed. When breakdowns occur, there is immediate interruption of the process implemented in the machine.

All transition input and output arc weights are equal to 1 except the ones leading to redefinition of the performed process. C-places describe resource availability; D-places correspond to system states; D-transitions describe state changes, C-transitions correspond to the speed of continuous events. The discrete part periodically redefines the product type manufactured with respect to predefined criteria (e.g. WIP minimization), while the continuous part describes processes that transform raw materials to final products.

B. *Hybrid Petri net model*

Multi-assembly machine module performs assemblies consisting of combinations of different numbers of initial parts, from 2 to n_{Ai1}= initial parts number. Not all theoretical possible combinations of raw materials lead to products that have practical meaning. Parts resulting in non-valid combinations are never supplied concurrently to a machine. Multiple parts of a type do not participate in an assembly.

The discrete part of the model of Fig. 2 is constructed so that it is not possible for a machine to perform in parallel multiple processes (this practically means that no multiple C-transitions representing different assembly types may fire in parallel). D-place capacities are set one, since there is one token in net's discrete part defining at each time instant it's state, and net's structure ensures that it remains 1-bounded.

Place p_1 represents machine operational and ready to produce (waiting to define next process) and is connected with places p_3-p_{nAi1+2} representing setup for different assembly types. p_2 represents machine out of order and all net's D-places representing process types are connected to it. When machine is repaired after a breakdown, the type of assembly performed is not the same as before; hence p_1 is connected through timed transitions to all places p_3-p_{nAi1+2}. At different times, $n_{A_{i2}} = \sum_{s_i=2}^{n_{A_{i1}}} \frac{n_{A_{i1}}!}{s_i!\left(n_{A_{i1}}-s_i\right)!} - q_i$ assembly types are performed ($q_i \geq 0$ is the non-valid assemblies number). Each part type participates at most in $c_1 = \sum_{s_i=2}^{n_{A_{i1}}} \frac{(n_{A_{i1}}-1)!}{(s_i-1)!\left(n_{A_{i1}}-s_i\right)!}$ assembly types. After the repair of a breakdown, there is a structural conflict as all process types can be theoretically performed, but in fact, only one type is performed at each time instant. This conflict is resolved with respect to system quantities (e.g. number of tokens in input buffers) and assigned priorities.

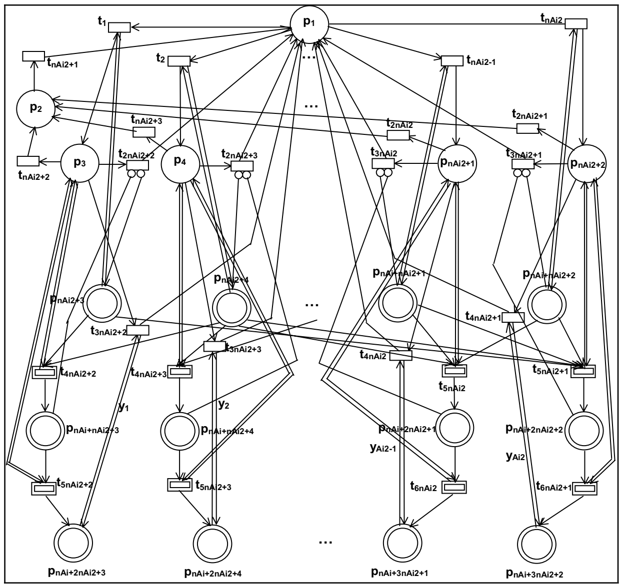

Fig. 2: Multi-assembly machine module HTPN model.

When an initial buffer becomes empty and no parts of this type are available at the machine, or a final buffer reaches it's maximum capacity, the process performed has to change in order not to lose time and production capabilities. The first event happens through transitions $t_{2*nAi2+2}$ - $t_{3*nAi2+1}$. These are connected with places representing process types and through inhibitor arcs with initial buffers and places representing parts at machines, so that as soon as all parts of a type have been processed, no firing of these transitions can take place, and lead token to p_1 to redefine the performed process. The second event takes place through the firing of $t_{3*nAi2+2}$–$t_{4*nAi2+1}$ that are connected with places representing process types and with pairs of arcs of weight y_j with final buffers (y_i=the maximum capacity of the respective final buffer). $t_{4*nA2i+2}$ – $t_{5*nAi2+1}$ are connected with initial buffers and with pairs of arcs with process types places in order to represent parts supply at the machine. Finally, pairs of arcs connect places representing process types with transitions describing process performance, since process of a type in a machine can not continue when the net state has changed. The meanings of model nodes are presented in Table I.

TABLE I
FUNDAMENTAL HTPN MODULES NODES (P AND T) MEANINGS

Node type	Node	Meaning
P_d	p_1	Machine operational and ready to process
	p_2	Machine out of order
	$p_3 - p_{n_{Ai2}+2}$	Machine setup for type $1-n_{A_{i2}}$ parts
P_c	$p_{n_{Ai2}+3} - p_{n_{A_i}+n_{Ai2}+2}$	Initial parts buffers
	$p_{n_{A_i}+n_{Ai2}+3} - p_{n_{Ai}+2n_{Ai2}+2}$	Type $1-n_{A_{i2}}$ assembly completed
	$p_{n_{A_i}+2n_{Ai2}+3} - p_{n_{Ai}+3n_{A_{i2}}+2}$	Final product buffers
T_d	$t_1 - t_{n_{Ai2}}$	Set machine for $1-n_{A_{i2}}$ type process
	$t_{n_{Ai2}+1}$	Breakdown repair
	$t_{nA_{i2}+2} - t_{2nA_{i2}+1}$	Machine breakdown while performing n_i type assembly
	$t_{2nA_{i2}+2} - t_{3nA_{i2}+1}$	Finish of n_i type raw materials -change assembly type
	$t_{3nA_{i2}+2} - t_{4nA_{i2}+1}$	Final buffer for type n_i products is full - change process type
T_c	$t_{4nA_{i2}+2} - t_{5nA_{i2}+1}$	Perform assembly $1-n_{A_{i2}}$
	$t_{5nA_{i2}+2} - t_{6nA_{i2}+1}$	Move product from machine to output buffer

Considering the multi-assembly machine HTPN model of Fig. 2 with any finite $\boldsymbol{m_0}$, one may observe that: *i)* Conflicts exist between parts process and machine breakdown and also between different assembly types when a machine is ready to process. In the first case, breakdown has the highest priority. In the second, decision is made according to system status. *ii)* As long as there is part availability in the input buffers, operations continue until a breakdown occurs. *iii*) Assembly module is not generally live. Initial parts in buffers define the duration that it remains live. No deadlock occurs as long as there is part availability (*D*-part reaches steady state -machine ready to process- when raw materials process has finished). *iv)* Multi-assembly HTPN model is **k**-bounded. Absence of self-loops in combination with the fact that it is totally covered by P-invariants, ensuring token preservation, guarantee it. *v)* Multi-assembly HTPN model is not conservative since it uses multiple parts for a product. *vi)* Module is non-persistent due to the existence of conflict transitions in module's *D*-part. There is additional conflict in the *C*-part, since initial parts may participate in processes leading to different final products. *vii)* Token preservation and the machine mutually exclusive states are described by the respective P-invariants. *viii)* Module is not repetitive and not consistent - no repetitive sequences of transitions whose firing results in m_0 or in periodic appearance of a restricted number of markings exist. Thus, there are no T-invariants.

C. Calculation of multi-assembly HTPN model nodes and P-invariants complexities

The multi-assembly machine module has $(\boldsymbol{n_{Ai1}+1})$ P-invariants. One refers to the discrete part of the net describing the mutually exclusive machine states ($\boldsymbol{n_{Ai2}}$ states represent machine setups for different assembly types; one refers to machine breakdown and one to machine being operational and ready to produce). The other *P-invariants* refer to the continuous part of the net, describing tokens preservation for each initial part type within the system. In these $\boldsymbol{k_i}$, $\boldsymbol{i}=1,\ldots,\boldsymbol{n_{Ai1}}$ is the initial sum of tokens in the respective set of places. Each such invariant consists of two places representing initial buffer and $\boldsymbol{i}$ type of parts entering the machine, and by $\boldsymbol{c_{2i}}$ places representing final buffers in which products whose the initial part is component, with $c_{2i} \le c_1 = \sum_{s_i=2}^{nA_{i1}} \frac{(n_{A_{i1}}-1)!}{(s_i-1)!\left(n_{A_{i1}}-s_i\right)!}$. The equality stands when all assemblies in which the initial part $\boldsymbol{i}$ participates are valid. Letting $m(p_{n_{Ai1}+2n_{Ai2}+3}) = A_1,\ldots,\; m(p_{n_{Ai1}+3n_{Ai2}+2}) = A_{n_{Ai2}}$, the *P-invariants* of multi-assembly machine module are:

$$m(p_1)+m(p_2)+\ldots+m(p_{nAi2+1})+m(p_{nAi2+2})=1$$

$$m(p_{nA_{i2}+3})+m(p_{nA_{i1}+nA_{i2}+3})+A_1+A_2+\ldots+A_{c_{21}-1}+A_{c_{21}}=k_1$$

$$m(p_{nA_{i2}+4})+m(p_{nA_{i1}+nA_{i2}+4})+A_1+A_2+A_4+\ldots+A_{c_{22}}+A_{c_{22}+1}=k_2$$

$$\vdots$$

$$m(p_{nA_{i1}+nA_{i2}+2})+m(p_{nA_{i1}+2nA_{i2}+2})+A_{n_{Ai2}-c_{2n_{Ai1}}}+\ldots+$$
$$+A_{n_{Ai2}-1}+A_{n_{Ai2}}=k_{n_{Ai1}}$$

The nodes complexities of multi-assembly machine HTPN model are presented in Table II.

TABLE II:
MULTI ASSEMBLY MACHINE HTPN MODEL'S NODES COMPLEXITIES.

	Node type	***Nodes number***
Multi-Assembly HTPN model	*DP*	$n_{Ai2}+2$
	DT	$4n_{Ai2}+1$
	CP	$n_{Ai1}+2n_{Ai2}$
	CT	$2n_{Ai2}$

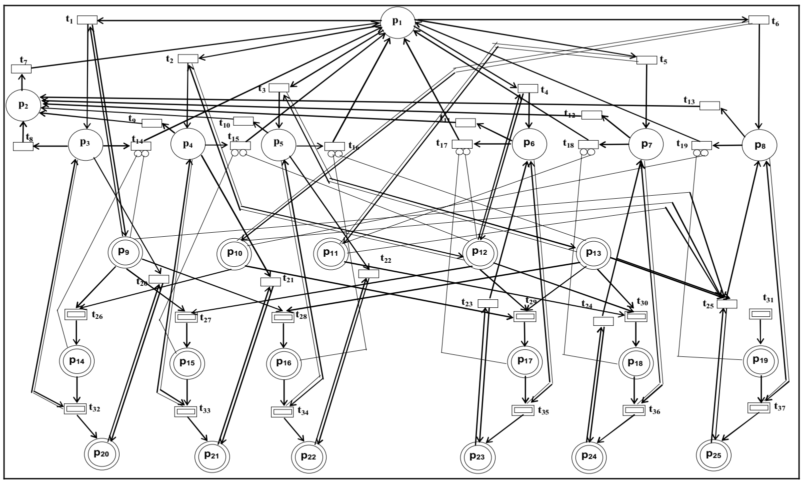

Fig. 3: Example of multi-assembly machine production system HTPN model.

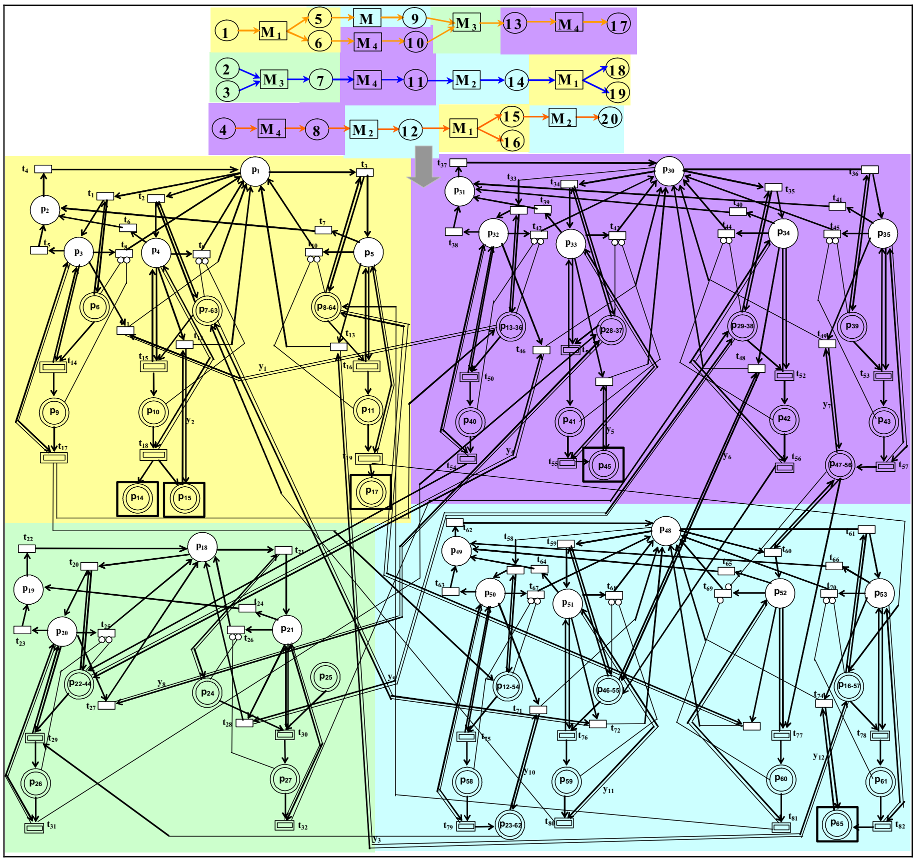

Fig. 4: Multi-operational production system and it's HTPN model.

D. *An Example*

A multi-assembly machine production system, in which a machine uses five types of raw materials A-E to produce different final product types, is considered. From the twenty five theoretically possible raw materials combinations, the following eight lead to production of valid products: *1) AB, 2) AC, 3) AD, 4) AE, 5) BCD, 6) BDE, 7) CDE* and *8) ABCE.* The HTPN model of this system is presented in Fig. 3. Models nodes and P-invariants are consistent with the ones calculated from the equations introduced here.

V. Method Application And Simulation Results

The practical value of the method is illustrated through its application to the production system of Fig. 4. The HTPN model of the overall system is constructed following the synthesis procedure introduced in [1].

The case study production system is composed of 4 machines and 20 buffers (4 input, 5 output and 11 internal). Five types of final product, stored in output buffers 16 – 20, are produced. Each machine allocates a percentage of its operational time for each product. Initial parts in buffers 2 and 3 follow the same route; two final products result from each disassembly operation in M_1. Each machine performs 2-4 types of processes in different time intervals. In total, 12 types of processes are performed in the system.

Parts enter system through buffers 1-4 (places p_6, p_{24}, p_{25}, p_{39}). Final products are found in output buffers 16-20 (places p_{14}, p_{15}, p_{17}, p_{45}, p_{65}). All other buffers contain in-process parts. Overall system HTPN model consists of 54 places. There exist 12 P-invariants, 4 referring to mutually exclusive machine states and 8 to part preservation in sets of places.

For any finite marking m_0, the HTPN model of the overall system is not generally live, is k-bounded, not conservative, non-persistent, not repetitive and not consistent. These properties arise from fundamental modules.

In m_0 there are 100 parts in each of the places p_6, p_{24}, p_{25} and p_{39}. Initially all machines are operational and ready to produce, meaning, that places p_1, p_{18}, p_{30} and p_{48} contain one token each in m_0. The initial marking of all other places is zero. Net's operational phase is completed when places representing final buffers p_{14}, p_{15}, p_{17} p_{45} and p_{65} contain 100 tokens each. In final net state m_f places p_1, p_{18}, p_{30} and p_{48} contain one token each (machines are operational and ready to produce but do not have raw materials and remain idle).

System's performance is studied through simulations performed using Visual Object Net [9]. Firing speeds of continuous transitions representing parts process are: $\{t_{14}, t_{15}, t_{16}, t_{29}, t_{30}, t_{50}, t_{51}, t_{52}, t_{53}, t_{75}, t_{76}, t_{77}, t_{78}\} = \{1.5, 1.3, 1.2, 1, 1.3, 1.9, 1.1, 2, 1.5, 0.8, 1.5, 1.3, 2\}$ parts/time unit. Firing speeds of transitions representing processed parts removal are $\{t_{17}, t_{18}, t_{19}, t_{31}, t_{32}, t_{54}, t_{55}, t_{56}, t_{57}, t_{79}, t_{80}, t_{81}, t_{82}\} = \{2, 2, 2.5, 2.5, 2.5, 2, 2, 2, 2, 1.5, 1.5, 1.5, 1.5\}$ parts / time unit.

D-transitions that refer to machine repair follow normal distributions (normally distributed numbers between zero and *x* defined as (*rnd(x)*), are generated describing the duration of breakdown repairs). So, $\{t_1, t_{22}, t_{37}, t_{62}\}= \{rnd(5), rnd(6), rnd(5), rnd(7)\}$. The respective stands for transitions referring to breakdown while performing each assembly type. Durations between breakdowns are equal for all part types. These are: $\{t_5, t_6, t_7, t_{23}, t_{24}, t_{38}, t_{39}, t_{40}, t_{41}, t_{63}, t_{64}, t_{65}, t_{66}\}=\{rnd(20), rnd(20), rnd(20), rnd(22), rnd(22), rnd(20), rnd(20), rnd(20), rnd(20), rnd(18), rnd(18), rnd(18), rnd(18)\}$. All other *D*-transitions occurrence is immediate when necessary preconditions are satisfied.

With these values, simulation is terminated after 465.4 time units. Then the firing speed of t_{75} is increased from 0.8 to 1.5 parts/time unit. Change of a transition firing speed represents changes in the operational speed of a machine. In this case, simulation is completed in 419.9 time units - a reduction of production time by 11%. Then the speed of t_{29} is doubled (from 1 to 2 parts/time unit) and simulation is completed after 382.9 time units, meaning that there is an extra reduction of the simulation time by 8,6%. The final modification step concerns changing the time between M_2 breakdowns. Breakdown appearances of machine 2 are normally distributed in (0, 30). Then simulation is completed in 358.7 time units, resulting in an additional reduction of production time almost 6.32%. The overall reduction of production time with all the described changes is almost 23%. Simulations may be continued until the optimization of a given objective (simulation time minimization, throughput maximization etc.). Overview of the calculated quantitative parameters for the simulations is presented in Table II.

TABLE II:
CALCULATION OF SYSTEM'S QUANTITATIVE FEATURES.

PARAMETER	**SIMULATION**			
	Initial	**Second**	**Third**	**Fourth**
Duration (time units)				
In time units	465.4	419.8	382.9	358.7
Mean production time (time units)				
Type 1 products	4.423	3.488	3.618	3.587
Type 2 products	4.654	3.972	3.828	3.352
Type 3 products	4.654	3.972	3.828	3.352
Type 4 products	3.1	4.198	3.797	2.737
Type 5 products	2.8	4.189	3.741	2.66
% of the simulation time that buffer i is full, i = 5 – 15				
Buffer 5 (p_{12-54})	59.1	26.6	35.28	41.65
Buffer 6 (p_{13-36})	6.27	22.75	22.51	33.43
Buffer 7 (p_{29-38})	34.33	50.69	41.68	52.05
Buffer 8 (p_{47-56})	44.97	51.11	35.75	15.86
Buffer 9 (p_{23-62})	3.1	12.34	9.01	22.14
Buffer 10 (p_{22-44})	45.94	13.84	16.95	21.27
Buffer 11 (p_{46-55})	43.88	40.64	5.51	19.24
Buffer 12 (p_{8-64})	4.34	15.97	0.08	6.97
Buffer 13 (p_{28-37})	1.12	3.76	15.8	20.23
Buffer 14 (p_{7-63})	0.13	9.72	2.27	1
Buffer 15 (p_{16-57})	15.94	25.87	11.12	13.02

VI. CONCLUSIONS

HTPNs have been used for modeling and study of multi-operational production systems. Multi-assembly module is generalized to represent assemblies consisting of different parts numbers. The proposed methodology is complete, covering the aspects of system composition, constraint satisfaction, complexity analysis and performance evaluation with minimum assumptions regardless of system topology.

REFERENCES

[1] Tsinarakis G. J., Valavanis K. P., "Modular Hybrid Petri Nets for Studying Multi-operational Production Systems Where Parts Follow Multiple Alternative Processes", in Proc. *IEEE Int. Conf. on Robotics and Automation*– New Orleans 2004, pp. 547 – 553.

[2] Zurawski R., Zhou M. C., "Petri nets and Industrial Applications: A Tutorial", *IEEE Trans. on Industrial Electronics*, vol. 41, no. 6, 1994.

[3] Vangheluwe H., "Discrete Event Modelling and Simulation", *COMP522, Modelling and Simulation, Mc Gill's Computational Science and Engineering programme*, 2002.

[4] Beek van D., Rooda J., Muyzenberg van den M., "Specification of combined continuous-time / discrete-event models", in *Modelling and Simulation* – in Proc. of 1996 European Simulation Multiconference, Budapest, pp. 219 –224.

[5] 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 Manuf. Systems, vol. 39, no. 2, 2001.

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

[7] Antsaklis P., "Scanning the Issue / Technology", in Proc. 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.

[8] 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.

[9] Drath R., Visual Object Net software package, v.2.07 a 2002. Available: http://www.ParamSoft.de/

[10] Champagnat R., Esteban P., Pingaud H., Valette R., "Petri net based modeling of hybrid systems", *Computers in Industry*, vol. 36, pp. 139 –146, 1998.

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

[12] Furcas R., Giua A., Piccaluga A., Seatzu C., "Modeling Production Systems with Inventory Using Hybrid Petri Nets", in Proc. *IEEE Int. Conf. on Control Applications*, Mexico City, 2001, pp. 434 - 440.

[13] 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.

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

[15] 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, pp. 255 – 282, 2001.

[16] Hodgson T., Ge G., King R., Said H., "Dynamic lot size/sequencing policies in a multi-product single-machine system", *IIE Transactions*, vol. 29, pp. 127 – 137, 1997.

[17] Tsinarakis G. J., Valavanis K. P., Tsourveloudis N. C., "Studying Multi-operational Production Systems with Modular Hybrid Petri Nets", In Proc. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* – Las Vegas – Nevada, 2003, pp. 3028 -3034.

[18] Whitney D., "The Potential for assembly modeling in product development and manufacturing", MIT, 1996.

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

[20] David R., Alla H., *Discrete, Continuous and Hybrid Petri Nets*, Springer - Verlag, 2005.

[21] Tsinarakis G. J., Tsourveloudis N. C., Valavanis K. P., "Petri Net Modeling of Routing and Operation Flexibility in Production Systems", accepted for presentation in *IEEE 13th Mediterranean Conference on Control and Automation,* Limassol, Cyprus, 2005.