Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Modular Petri Net Based Modeling, Analysis and Synthesis of Dedicated Production Systems

G. J. Tsinarakis, K. P. Valavanis, N. C. Tsourveloudis
Technical University of Crete
Intelligent Systems and Robotics Laboratory
Dept. of Production Engineering and Management
Chania, Greece GR-73100

***Abstract* – Ordinary t-timed Petri Nets are used for modeling, analysis and synthesis of random topology production systems and networks. Each production system is first decomposed into production line (transfer chain), assembly, disassembly and parallel machines modules and then their corresponding modular Petri Net models are derived. The overall system PN model is obtained via synthesis of the generic modules satisfying system constraints. P- and T- invariants are calculated and given a random topology production system, the total number of the system PN model nodes (places, transitions) is calculated from the corresponding generic PN modules. Results show the applicability of the proposed methodology.**

I. INTRODUCTION

Petri Nets and their modifications are widely used to study DEDS, production systems and networks. A Petri Net (PN) is defined as the five-tuple: $PN=\{P, T, I, O, m_0\}$, where $P=\{p_1, p_2 \dots p_n\}$ is a finite set of places, $T=\{t_1, t_2 \dots t_m\}$ is a finite set of transitions, $P \cup T=V$, where V is the set of vertices and $P \cap T = \emptyset$. $I: (P \times T) \rightarrow N$ is an input function and $O: (P \times T) \rightarrow N$ an output function with N a set of non–negative integers, and m_0 the PN initial marking. PN structural and behavioral properties capture precedence relations and structural interactions between system components [1]-[16].

In this paper, (which is the natural outgrowth of previous reported research [15], [17]) t-timed ordinary modular PNs are utilized for modeling, analysis, synthesis and simulation of random topology dedicated production systems, in which machines fail and are repaired randomly. Four generic PN modules, corresponding to the *production line* (or *transfer chain*), *assembly*, *disassembly* and *parallel machines* modules are derived. The overall system PN model is obtained via synthesis of the component models considering simultaneously overall system constraints, and the Martinez-Silva algorithm [19] is used to calculate P- and T- invariants. Given a random topology production system, the total number of the system PN model nodes (places, transitions) is calculated from the corresponding generic PN modules.

Paper contributions include: i) modular PN based approach is independent of the system architecture and structure, ii) the model construction method may be applied to any configuration DEDS, and iii) analysis and synthesis of any complex system is accomplished in terms of analysis and synthesis of the 4 basic PN modules.

Section 2 presents the generic modules; Section 3 derives their PN models; Section 4 presents the overall system PN model; Section 5 presents simulation results, while Section 6 concludes the paper.

II. PRODUCTION SYSTEM GENERIC MODULES

A production system may be viewed as a network of machines/workstations and buffers. Random machine breakdowns disturb the production process and starvation or blocking may occur affecting the downstream and upstream buffer levels. Events that may occur in a production network include changes in buffer states (full or empty) and changes in machine states (up or down). When a machine breaks down preceding machines remain operating until one of their downstream buffers is filled. Similarly, succeeding machines continue processing until their upstream buffers are empty.

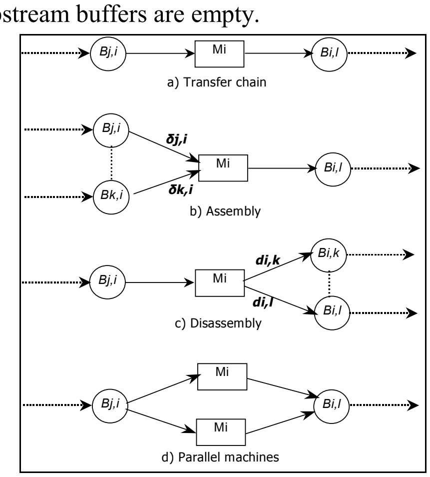

Fig.1 Fundamental, generic, modules

The production floor modeling approach introduced and explained in [15], [17] is extended so that every production system or network is decomposed into the line (chain), assembly, disassembly and parallel machines module, the simplest version of which is shown in Figure 1 (circles and rectangles represent buffers and machines; notation is straightforward). These modules, if connected to each other may represent manufacturing networks of various layouts.

Generalizations of the four generic modules are obvious; transfer chains may contain n machines ($n+1$ buffers), assembly (disassembly) modules may have n input (output) buffers and parallel machine module may have n machines.

III. PETRI NET MODELS OF GENERIC MODULES

The four basic PN models corresponding to the four generic modules of Figure 1 (called *generic PNs* from now on) are shown in Figures 2-5. Timed transitions are presented as white rectangles, while immediate transitions as black rectangles. All transition input and output arc weights are equal to 1. Table 1 explains the meaning of each place and transition. Places $p_0 - p_5$ and transitions $t_1 - t_4$ have the same meaning in all four generic PNs. Transitions correspond to system activities resulting in state changes, while places correspond to resource (machine, parts) availability or state (machine up, down, working, free). Table 2 shows PN module complexity for the general case of n machines in each module.

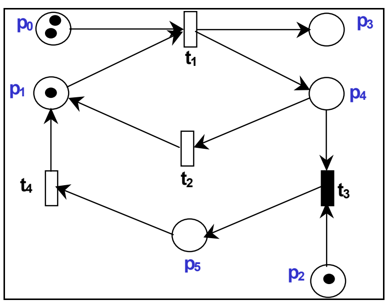

Fig.2 Production line (chain) PN module

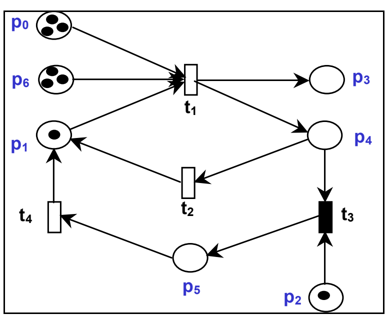

Fig.3 Assembly PN module

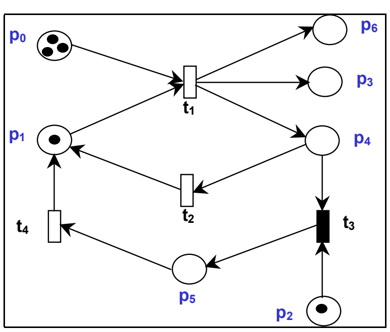

Fig. 4 Disassembly PN module

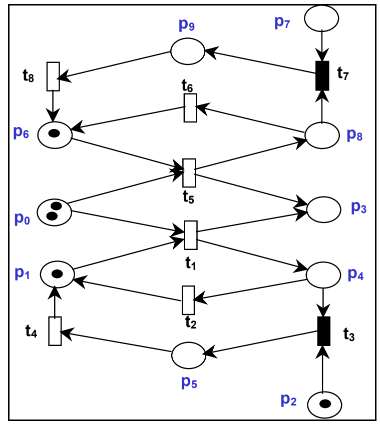

Fig. 5 Parallel machines PN module

The generic PN modules have been derived based on the following mostly realistic assumptions: i) buffers have finite capacities, ii) machines operate at a given speed that may change at specific moments according to events taking place in the system, iii) setup times and transportation times of pieces through the production system are negligible compared to production times, iv) machine breakdowns may happen infinitely often, but only after the completion of a production cycle. Tokens shown in the generic PN modules are for demonstration purposes only: the token in p_1 indicates that a machine is free and operates (next piece production may begin), but when one production cycle terminates, the next cycle starts either immediately (through t_2) or after the appearance of a machine breakdown and its repair (path p_2, t_3, p_5, t_4).

A. Discussion

Considering the four generic PN modules as shown in Figure 1 (not for simulation purposes) with any finite initial marking m_0, several observations are made: a) As long as there is part availability in the input buffer(s), all four generic PNs after the completion of one production cycle return to the state of starting a new cycle; b) the parts number in the initial buffer(s) defines the exact number of production cycles; c) all modules are *k-bounded*; d) modules are non-conservative (transition t_3 consumes two tokens and produces one, in assembly and disassembly module transition t_1 is also non conservative, in parallel machines module t_7 is non conservative); e) modules are non-persistent (firing of t_3 may disable t_2); f) modules are not repetitive and not consistent.

For the transfer chain, the upper limit for k is defined as $min\ \{max\ \{C_0,\ C_3\},\ (m_0(p_0)+m_0(p_3))\}$, where C_i is the maximum capacity of p_i. Maximum number of tokens in a place is the minimum of maximum capacity of two buffers and the sum of the initial tokens in these places.

For the assembly module, the upper limit for k is different since there are at two input buffers, defined as $min\ \{max\ \{C_0,\ C_3,\ C_6\},\ (m_0(p_3) + max\{\ m_0(p_0),\ m_0(p_6)\})\}$.

For the disassembly assembly module, the upper limit for k is calculated considering two output buffers as $min\{max\{C_0, C_3, C_6\}, (m_0(p_0) + max\{m_0(p_3), m_0(p_6)\})\}$.

B. Invariants

The Martinez-Silva algorithm [19] is used to calculate the minimal P- and T- invariants (all other invariants are linear combinations of minimal). P-invariants are nonzero nonnegative integer solutions $\boldsymbol{X}$ of the equation $\boldsymbol{X^T A = 0}$ that also satisfy $\boldsymbol{X^T m = X^T m_0}$ where $\boldsymbol{X}$ is an $\boldsymbol{n_p}$-element vector, $\boldsymbol{m_0}$ the initial marking of the net and $\boldsymbol{m}$ a marking of the reachability set of $\boldsymbol{m_0}$, $\boldsymbol{R(m_0)}$. *T-invariants* are the nonzero nonnegative integer solutions $\boldsymbol{Y}$ of the matrix equation $\boldsymbol{AY = 0}$ where $\boldsymbol{Y}$ is $\boldsymbol{n_t}$-element vector. There are $\boldsymbol{(n_p-r)}$ basic *P-invariants* and $\boldsymbol{(n_t-r)}$ *T-invariants*, where $\boldsymbol{r=rank(A)}$. *P-invariants* express a notion of token conservation in sets of places for all reachable markings without enumeration of the reachability set. *T- invariants* describe a transition firing sequence $\boldsymbol{s}$, such that $\boldsymbol{m_j \rightarrow m_j}$.

Node	Model	Meaning
P_0	Common	Parts available in initial buffer
p_1	Common	Machine available to process part
p_2	Common	Machine breakdown
p_3	Common	Parts in final buffer
p_4	Common	Machine finished process of a part
p_5	Common	Machine out of order
p_6	Assembly	B type parts available in corresponding initial buffer
	Disassembly	B type parts in the corresponding final buffer
	Parallel machines	Second Machine (M_2) available to process part
p_7	Parallel machines	Machine M_2 breakdown
p_8	Parallel machines	Machine finished process of a part
p_9	Parallel machines	Machine M_2 out of order
t_1	Common	Machine processing (producing) part
t_2	Common	Empty machine's signal return
t_3	Common	Machine breaks down
t_4	Common	Machine has been repaired and is available to produce again
t_5	Parallel machines	Machine M_2 is processing (producing) part
t_6	Parallel machines	Empty machine's signal return for M_2
t_7	Parallel machines	Machine M_2 breaks down
t_8	Parallel machines	Machine M_2 has been repaired and is available to produce again

Table 1 Basic modules node (P and T) explanation

Model	Nodes type	Basic (generic) model	Generalized model (n components)
Transfer chain	P	6	5 * n + 1
	T	4	4 * n
Assembly	P	7	n + 5
	T	4	4
Disassembly	P	7	n + 5
	T	4	4
Parallel machines	P	10	2 + 4 * n
	T	8	4 * n

Table 2 Complexity of generalized PN modules for n machines

The transfer chain module has 2 P-invariants and no T-invariant. The P-invariants are *{1 0 0 1 0 0}* and *{0 1 0 0 1 1}* resulting in $m(p_0) + m(p_3) = n_1$ and $m(p_1) + m(p_4) + m(p_5) = 1$. The first P-invariant guarantees that the sum of parts in the initial and in the final buffer is constant and equal to the initial sum of parts in these buffers n_1, while the second shows 3 mutually exclusive machine states (machine ready to process part, empty or machine breakdown). The two above P-invariants are common for all four modules; however the other modules have one more. The third P-invariant of the assembly module is $m(p_6)+m(p_3)=n_2$ and refers to the sum of tokens of the second initial buffer and final buffer that is equal to the initial sum of parts in these two buffers. The third P-invariant of the disassembly module is $m(p_0)+m(p_6)=n_2$ and refers to the sum of tokens of the initial buffer and the second final buffer. The third P-invariant of the two parallel machines module is $m(p_6)+m(p_8)+m(p_9)=1$ and refers to the mutually exclusive states of the second machine M_2, same with the ones of the first machine.

IV. PN MODULE SYNTHESIS

The synthesis procedure of the simplest two transfer chains is shown in Figure 6. Generalizations are provided.

Observing Figure 6, it is obvious that places p_3 and p_6 are fused in place $p_{3\text{-}6}$. The total number of places is reduced by one, while transitions are equal to the total of each module transitions. The combined PN input places are reduced by one ($p_{3\text{-}6}$ is an internal place, not input buffer any more). The maximum capacity of $p_{3\text{-}6}$ may be defined as $C_{3\text{-}6} = min\{C_3, C_6\}$ or $C_{3\text{-}6} = max\{C_3, C_6\}$ or with any number in between (based on system constraints). Obviously, $m_0(p_{3\text{-}6})=m_0(p_3)+m_0(p_6)$.

The combined PN properties may be detected accordingly by simulation and use of appropriate tools. There exist three P-invariants (two are identical with the individual module P-invariants). Two refer to the mutually exclusive states of the combined PN given by equations $m(p_1)+m(p_4)+m(p_5)=1$ and $m(p_7)+m(p_{10})+m(p_{11})=1$. The third refers to the preservation of the total number of parts in the PN and is given by $m(p_0) + m(p_{3\text{-}6}) + m(p_9) = n_1$, where n_1 is the initial sum of parts (tokens) in the three places. Synthesis of other generic PN modules is obtained in a similar way but due to space limitations are omitted.

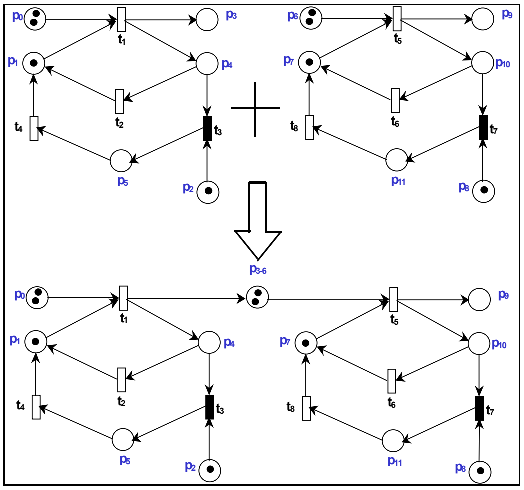

Fig. 6 Synthesis of two generic transfer chain modules

A. Generalizations

It is possible to calculate the number of nodes of a random topology production system PN model from the corresponding PN modules that compose it and from the number of external parts entering the system. The latter number is necessary to compute the number of fused places in the individual modules connection points.

Consider first that a production system combined PN model is derived in terms of the four generic Petri Net modules; that is n_1 modules of transfer chain, n_2 modules of two-piece assembly, n_3 modules of two-piece disassembly, n_4 modules of two-parallel machines, and that there are n_5 (external, non-fused) input places. The combined PN model consists of $4*(n_1+n_2+n_3)+8*n_4$ transitions. The total number of generic PN modules places when considered separately is $6*n_1+7*(n_2+n_3)+10*n_4$. Fusion of places at connection points reduces the number of places by $n_1+n_3+n_4+2*n_2-n_5$ (number of places that are not external inputs). Thus, the total number of the combined net places is $5*(n_1+n_2)+6*n_3+9*n_4+n_5$.

Considering next individual PN modules as shown in Table 2, the total number of transitions is given by $4*(n_2+n_3)+\sum_{i=1}^{n_1} 4*l_i+\sum_{j=1}^{n_4} 4*l_j$, where l_i and l_j are the transfer chain and parallel machine module components. The total number of places is calculated as

$$\sum_{i_1=1}^{n_1}(5*l_{i_1}+1)+\sum_{i_2=1}^{n_2}(l_{i_2}+5)+\sum_{i_3=1}^{n_3}(l_{i_3}+5)+$$

$$+\sum_{i_4=1}^{n_4}(4*l_{i_4}+2)-(n_1+n_3+n_4+\sum_{i_2=1}^{n_2} l_{i_2}-n_5)=$$

$$=\sum_{i_1=1}^{n_1}5*l_{i_1}+\sum_{i_3=1}^{n_3} l_{i_3}+\sum_{i_4=1}^{n_4}4*l_{i_4}+5*n_2+4*n_3+n_4+n_5,$$

where l_{i1}, l_{i2}, l_{i3}, l_{i4} show the number of machines (chain), number of input and output buffers in the assembly and disassembly modules, and the number of parallel machines in the parallel machine module, respectively.

V. A CASE STUDY

The production system of Figure 7 with its PN model shown in Figure 8 is used as a case study [15], [17]. The system is composed of 2 transfer chains, 2 assembly modules and 1 disassembly module. Parts enter the system through initial buffer (Module 1), while parts reaching the final buffer after machine M_5 are final parts ready to be removed from the system. PN consists of 28 places and 20 transitions. 5 transitions are immediate corresponding to potential machine breakdowns. There are 2 external input places p_0 and p_{25}. Parts reaching p_{29} are finished parts.

From Figure 9, it is obvious that internal buffer $p_{9\text{-}19}$ is full of parts for a large percentage of the system function, potentially resulting in frequent machine (M_4) blockage, while other buffers (like $p_{22\text{-}32}$) do not even reach half of their capacity. Changing buffer capacities, for example by reducing arbitrarily the capacity of buffer $p_{22\text{-}32}$ to 3 (from 8) and by repeating the simulation with the rest of the parameter values the same, the simulation is terminated after 547 steps with total duration 228 time units. Figure 10 shows the internal buffers levels during this simulation.

Next reduction of the buffer $p_{16\text{-}26}$ capacity from 8 to 5 is tried. Simulation is terminated after 553 steps and the total time is 229 time units. The mean production time after these two changes is 6.94 time units. Figure 11 shows the results. This process may be repeated (trial and error) until all buffers work at their capacities.

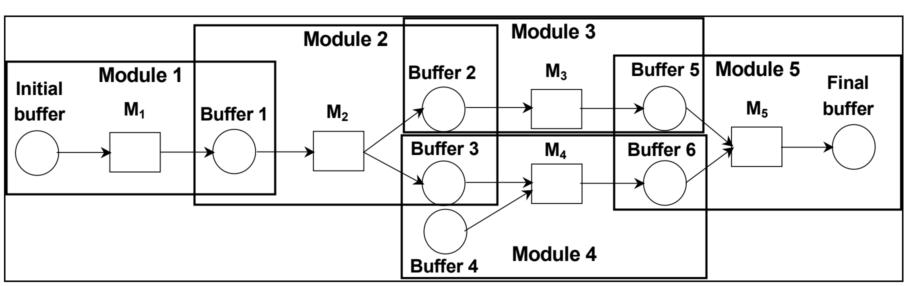

Fig. 7 A Production System and its module decomposition

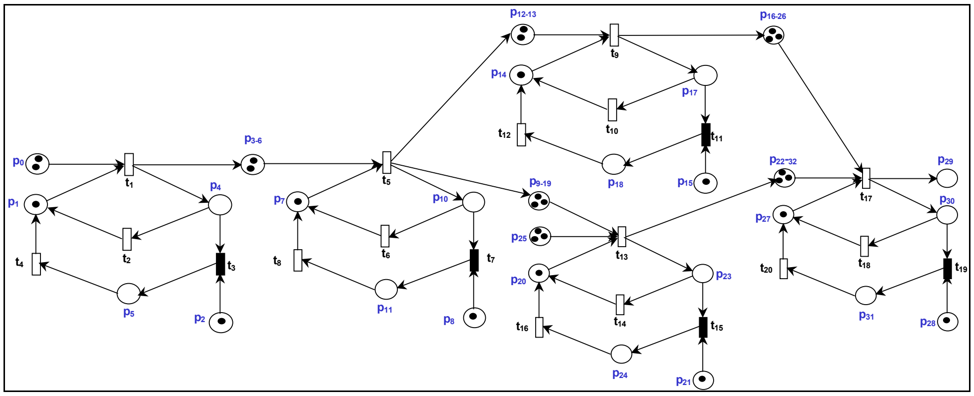

Fig.8 Overall system Petri Net model

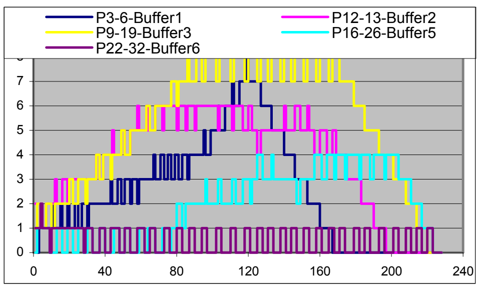

Fig. 9 Internal buffer levels during simulation

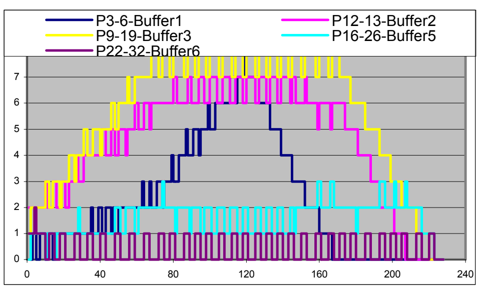

Fig. 10 New buffer levels-reduced capacity of p_{22-32}

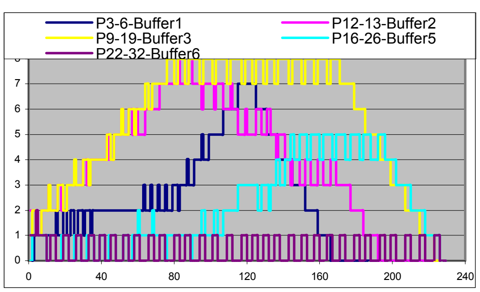

Fig. 11 New buffer levels-reduced capacity of p_{16-26}

System performance may also be improved by changing machine times, reducing idle periods, etc. By reducing the production time of machine M_4 by 1 time unit, the simulation is completed after 475 steps and the total time needed to produce the 33 pieces is 196 time units. The mean production time is 5.94 time units, 1 time unit less than before (15% performance improvement). The new buffer levels are shown in Figure 12.

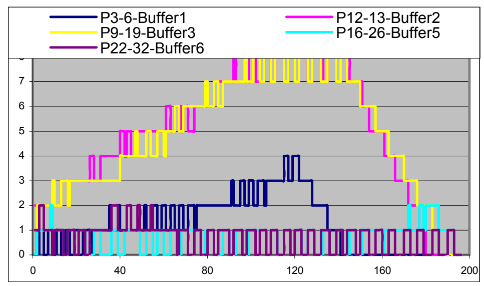

Fig. 12 Buffer levels after changing M_4 production rate.

Further system performance improvement may be obtained by studying machine breakdown behaviour. Reducing the frequency of machine breakdown appearance (with preventive machine maintenance) will increase system operational periods. Let's consider that the mean time of the exponential breakdown appearance for machine M_4 is 6 time units instead of 3. In this case the simulation is completed in 459 steps with total duration of 189 time units and mean production time of 5.73 time units, 0.21 time units less than before. Figure 13 shows the corresponding buffer levels during simulation.

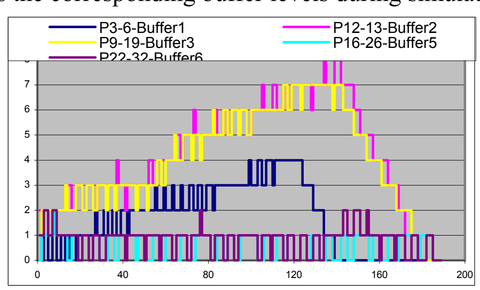

Fig. 13 New buffer levels after changing M_4 breakdowns mean time

Additionally, for this simulation, the corresponding machine operation rates are approximately 63.5% for M_1, 49.2% for M_2, 52.4% for M_3, 67.7% for M_4 and 70% for M_3. This shows that M_2 and M_3 are idle for longer time periods in comparison with the other machines (almost for half of the operational time of the system) and so they may be used for other activities as well. Figure 14 shows how production has changed as function of the changes made.

An interesting point that concerns production systems and is obvious also by simulations, is that the behaviour of the net is heavily determined by the slowest machine, as other machines of the net are obligated to follow its production rhythms through the appearance and spread of blockages and starvation phenomena.

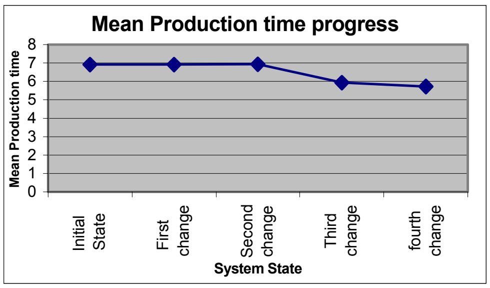

Fig. 14 Progress of the mean production cycle time in relation with the described system changes

VI. CONCLUSIONS

Modular PNs have been used for modelling, analysis and synthesis of random topology production networks. Four generic modules have been considered and their corresponding generic PN modules have been derived. Generalizations have been provided and expressions for the number of system nodes for random configuration systems have been calculated. P-invariants provide further insight to production systems study. Simulation results demonstrate the effectiveness of the proposed method.

VII. ACKNOWLEDGEMENTS

This work has been partially supported by a research grant from the GSRT/EU, EPAN M.4.3 Greece-Slovenia. The authors wish to thank Henryk Anschuetz for the provision of the full version of HPSIM Petri net simulator.

V. REFERENCES

[1] Balduzzi F., Giua A., Menga G., First – Order Hybrid Petri Nets: A model for Optimization and Control, *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 4, August 2000.

[2] Domg M., Chen F. F., Process modelling and analysis of manufacturing supply chain networks using object-oriented Petri nets, *Robotics and Computer Integrated Manufacturing* 17, pp. 121 – 129, 2001.

[3] Frey G., Assembly line sequencing based on Petri-net T-invariants, *Control Engineering Practice* 8, pp. 63-69, 2000.

[4] Jehng Wern-Kueir, Petri net models applied to analyze automatic sequential pressing systems, *Journal of Materials Processing Technology* 120, pp. 115 – 125, 2002.

[5] Jeng M.D., Xie X., Modeling and Analysis of Semiconductor Manufacturing Systems With Degraded Behavior Using Petri Nets and Siphons, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 5, October 2001.

[6] Valavanis K., On the Hierarchical Modeling Analysis and Simulation of Flexible Manufacturing Systems with Extended Petri Nets, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 1, February 1990.

[7] Desrochers A., Al – Jaar R., *Applications of Petri Nets in Manufacturing Systems –Modeling, Control and Performance Analysis*, IEEE Press, 1995.

[8] Moody J., Antsaklis P., *Supervisory Control of Discrete Event Systems Using Petri Nets*, Kluwer Academic Publications, 1998.

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

[10] Proth J. M., Xiaolan X., *Petri Nets, A Tool for Design and Management of Manufacturing Systems*, John Wiley & Sons, 1996.

[11] Zhou M.C., DiCesare F., *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Kluwer Academic Publishers, 1993.

[12] Murata T., Petri Nets: Properties, Analysis and Applications, *Proceedings IEEE*, Vol. 77, No. 4, pp. 541 – 580, April 1989.

[13] Gu T., Bahri P., A survey of Petri net applications in batch processes, *Computers in Industry* 47, pp. 99 – 111,2002.

[14] Valette R., Chezalviel – Pradin B., Girault F., *An Introduction to Petri net theory, Fuzziness in Petri Nets*, Eds. J. Cardoso, H. Carmago, *Studies in Fuzziness and Soft Computing*, Vol. 22, pp. 3 – 24, Physica Verlag 1999.

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

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

[17] Ioannidis S., Tsourveloudis N., Valavanis K., Fuzzy Supervisory Control of Manufacturing Systems, *IEEE Transactions on Robotics and Automation* (Revised – Under review).

[18] Govil M., Fu M., Queueing Theory in Manufacturing: A Survey, *Journal of Manufacturing Systems*, Vol. 18, No. 3, 1999.

[19] Martinez J., Silva M., A Simple and fast algorithm to obtain all invariants of generalized Petri net, *Application and Theory of Petri Nets,* Vol. 52, pp. 301 – 311, New York: Springer-Verlag, 1982.

[20] Anschuetz Henryk, HPSIM, http://home.t-online.de/home/henryk.a/petrinet/e/hpsim_e.htm.