



**HAL**  
open science

# NESTED COLORED TIMED PETRI NETS FOR PRODUCTION CONFIGURATION OF PRODUCT FAMILIES

Linda Zhang, Brian Rodrigues

► **To cite this version:**

Linda Zhang, Brian Rodrigues. NESTED COLORED TIMED PETRI NETS FOR PRODUCTION CONFIGURATION OF PRODUCT FAMILIES. International Journal of Production Research, 2009, 48 (06), pp.1805-1833. 10.1080/00207540802585329 . hal-00548945

**HAL Id: hal-00548945**

**<https://hal.science/hal-00548945>**

Submitted on 21 Dec 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**NESTED COLORED TIMED PETRI NETS FOR PRODUCTION CONFIGURATION OF PRODUCT FAMILIES**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2008-IJPR-0502.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	15-Oct-2008
Complete List of Authors:	Zhang, Linda; University of Groningen, Operations Rodrigues, Brian; Singapore Management University, Lee Kong Chian School of Management
Keywords:	PRODUCTION MODELLING, PETRI NETS
Keywords (user):	PRODUCTION MODELLING, PETRI NETS



# NESTED COLORED TIMED PETRI NETS FOR PRODUCTION CONFIGURATION OF PRODUCT FAMILIES

Lianfeng (Linda) Zhang<sup>\*†</sup>, Brian Rodrigues<sup>‡</sup>

<sup>†</sup>Department of Operations, University of Groningen,  
Landleven 5, 9747 AD Groningen, The Netherlands

<sup>‡</sup>Lee Kong Chian School of Business, Singapore Management University  
Stamford Road 50, 178899, Singapore

**Abstract:** Production configuration is as an effective technique to deal with product variety while maintaining production stability and efficiency. It involves a diverse set of process elements (e.g., machines, operations), a high variety of component parts and assemblies and many constraints arising from product and process variety. Production configuration entails the selection and subsequent arrangement of process elements into complete production processes and the final evaluation of configured multiple alternatives. To better understand production configuration and its implementation, we study the underlying logic for configuring production processes using a dynamic modeling and visualization approach. This is accomplished by developing a new formalism of nested colored timed Petri nets (PNs). In view of the inherent modeling difficulties, in the formalism three types of nets - process nets, assembly nets and manufacturing nets - together with a nested net system are defined. Using an industrial example of vibration motors, we show how the proposed formalism can be applied to specify production processes at different levels of abstraction to achieve production configuration.

**Keywords:** Process platform, production configuration, Petri nets, modeling.

---

\* Corresponding author. Email: [L.Zhang@rug.nl](mailto:L.Zhang@rug.nl)

## 1. INTRODUCTION

The importance of planning production processes has been well recognized (Huang et al., 2004). As a vital link between design and production, planning determines how well a product can be fulfilled in terms of cost, lead time and quality (Martinez et al., 2000). Decision making in planning production processes is complex since it involves multiple, often conflicting, production performance metrics, a variety of operations types, operations precedence and alternative manufacturing resources of same kinds (Chan et al., 2001). Production processes of products consist of both manufacturing processes of component parts and assembly processes of component assemblies. Thus, production process planning provides inputs (e.g., parts, assemblies, manufacturing resources) to downstream computer-aided process planning and assembly planning for specifying detailed process parameters such as cutting speed, feed rate and depth of cut (Turner et al., 1992). It also provides inputs for scheduling needs (Martinez et al., 2000).

Manufacturers at large have adopted the strategy of developing product families in attempting to stay competitive through satisfying diverse individual customer requirements while maintaining low costs (Sanderson and Uzumeri, 1996). Successful product family development relies on achieving efficiencies in both designing and producing product families. Some researchers (e.g., Fixson (2005), Huang et al. (2005), Lu and Botha (2006), to name but a few) have pointed out that planning of production processes for product families can help achieve production stability, and eventually mass production efficiency. Production processes planning for product families is more challenging and complex when compared with that for individual products (Martinez et al., 2000). The reasons are threefold. First, the optimal production processes of individual products may not contribute to that of the product family as a whole due to the limited number of manufacturing resources (Zhang, 2007). Second, the solution space of production processes of product families is combinatorial in nature due to the large number of individual products, process types, operations types, and process elements (Martinez et al., 2000). Last, production processes planning for product families involves specific technical and managerial challenges, which are inadequately addressed in the literature (Lu and Botha, 2006).

1  
2  
3  
4 Among the authors reporting production process planning for product families, some  
5 approach this topic from a strategic level and, their solutions provide companies with  
6 managerial guidelines for deciding on production/manufacturing structures and investment  
7 choices (Blackhurst et al., 2005; Garg, 1999; Huang et al., 2005; Petersen et al., 2005; Raman  
8 and Chhajed, 1995). Others address planning manufacturing processes for part families  
9 (Schierholt, 2001; Williams et al., 2007) and planning assembly sequences for assembly  
10 families (De Lit et al., 2003; Gupta and Krishnan, 1998; Martinez et al., 2000). This suggests  
11 that few researchers deal with planning production processes for product families that  
12 involves both parts and assemblies at an operational level. In response to the lack of research,  
13 configuring production processes from a process platform in relation to a product family (i.e.,  
14 production configuration) has been proposed as an effective means of achieving product  
15 family production stability and efficiency (Jiao et al., 2007; Zhang, 2007).  
16  
17

18 A process platform organizes all data pertaining to a product family and the  
19 corresponding process family as an integrated common process structure. Thus, production  
20 configuration provides a generic umbrella to capture and utilize commonality, within which  
21 each new product fulfillment is instantiated and extended, thereby anchoring planning to the  
22 integrated common process structure. Within a process platform, elements such as processes,  
23 operations, machines, tools, fixtures and setups are selected and arranged to form optimal  
24 production processes for individual products by considering the cohort of the product family.  
25 Production configuration involves product and process variety exhibited by large volumes of  
26 different types of product and process family data, and many constraints, which, for example,  
27 describe the connections between different types of data. In addition, configuration needs to  
28 deal with the many process variations resulting from design changes among members in the  
29 product family. Lastly, in accordance with the many assemblies and parts at different levels of  
30 product hierarchies, production configuration concerns granularity issues (i.e., configuring  
31 processes for product items at different levels of abstraction). To help manufacturers better  
32 understand production configuration, this study addresses the underlying reasoning by  
33 modeling the process of configuring production processes for a product family from the  
34 corresponding process platform.  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Due to their executability, graphical representation and mathematical support, Petri nets

1  
2  
3  
4 (PNs) have been well recognized as a powerful modeling, simulation and evaluation tool for  
5 complex flows and processes (Peterson, 1981). As a graphical tool, PNs serve as a visual  
6 modeling technique and as a communication aid for describing models. As a mathematical  
7 tool, PNs can be exploited to perform qualitative and quantitative analysis of systems being  
8 modeled. A number of extensions have been made to classic PNs in order to satisfy different  
9 modeling requirements. Among all, colored PNs (CPNs; Jensen, 1995), timed PNs (TPNs;  
10 Ramachandani, 1974) and nested PNs (NPNs; Lomazova, 2000; Lopez-Mellado and  
11 Almeyda-Canepa, 2003) are of particular interest in this study. CPNs are able to provide a  
12 concise, flexible and manageable representation of large systems by attaching a variety of  
13 colors to tokens. By including timing, TPNs can capture systems physical behaviors through  
14 assuming specific durations for various activities. By considering PNs as tokens, NPNs  
15 model large and complex system through refinement and abstraction of PNs.  
16  
17

18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
Towards this end, in this paper, we apply PN techniques, specifically CPNs, TPNs and  
NPNs, to model production configuration. Accordingly, a new formalism of NCTPNs (nested  
colored timed PNs) is developed by integrating the basic principles of CPNs, TPNs and  
NPNs. In accordance with the different types of processes in high variety production and the  
inherent similarities among them, in the proposed formalism several nets are defined based  
on a basic PN structure. Product and process variety and configuration constraints are  
handled by attaching various data regarding product items and process elements to tokens,  
resulting in colored tokens. A reconfiguration mechanism, consisting of inhibitor arcs,  
reconfigurable transitions and conceptual machine places, is incorporated into the formalism  
to cope with process variations. Timing is introduced to facilitate the selection of proper  
machines, operations and processes. Moreover, a multilevel net system is further developed  
based on net nesting to tackle explicitly granularity issue in production configuration.

The rest of the paper is structured as follows: Section 2 provides a review of work  
relevant to PNs modeling applications. The definitions in the proposed formalism are  
presented in Section 3, following which the multilevel system of NCTPNs is introduced in  
Section 4. Also presented is the evolution of the net system through defining several rules. In  
the industrial example in Section 5, the formalism is applied to model production  
configuration of vibration motors for mobile phones. Also discussed are system analysis and

1  
2  
3  
4 application results. Further discussions along with some managerial implications are provided  
5  
6 in Section 6.

## 7 8 9 **2. RELATED WORK**

10  
11 To capture the hierarchical nature of assembly planning, [Thomas et al. \(1996\)](#) propose a  
12 hierarchical PN framework, where each control plan corresponding to an assembly robot  
13 operation is viewed as a lower level representation of assembly tasks at the higher level.  
14  
15 [Adamous et al. \(1998\)](#) introduce a hierarchical model of flexible assembly system control  
16 using object PNs, which is built on hierarchical decomposition of the system through  
17 coordination among lower level decomposed system elements. [Lomazova \(2000\)](#) discusses  
18 two level NPNs for modeling multi-agent distributed system and the extension to multi-level  
19 NPNs. She proves that NPNs are able to maintain the properties of classic PNs. To  
20 accommodate qualitative modeling of multiple mobile robot systems, [Lopez-Mellado and](#)  
21 [Almeyda-Canepa \(2003\)](#) develop a three level scheme of PNs, in which tokens themselves  
22 can be defined as PNs as well. In software system modeling, [Hiraishi \(2000\)](#) presents a  
23 PN-based model, which adopts the classic PNs to analyze and design multi agent systems. He  
24 represents agents using tokens, which themselves are PNs.

25  
26  
27 [Li and Lara-Rosano \(1999\)](#) develop a hybrid object-oriented colored PNs (OOCPNs) by  
28 enhancing OOCPNs with time delay and firing speed. The hybrid OOCPNs are used to model  
29 formally batch-based electronic component manufacturing systems. Recognizing the  
30 importance of solving deadlocks in flexible manufacturing systems, [Abdallaht et al. \(2002\)](#)  
31 put forward a deadlock free scheduling algorithm using TPNs in order to minimize mean  
32 flow time. [Reddy et al. \(1993\)](#) present an algorithm for qualitative and quantitative analysis  
33 of TPN models in manufacturing systems with focus on real time control and performance  
34 evaluation. [Liu et al. \(2002\)](#) propose a workflow modeling language-based CPNs called  
35 WFCP-nets (workflow based on CPNs) and apply it to the product development workflow.  
36  
37 [Chin et al. \(2006\)](#) put forward methodologies based on integrated definition language and  
38 CPNs for modeling and simulating complicated manufacturing processes. Researchers, e.g.,  
39 [Dotoli and Fanti \(2004\)](#), [Nandula and Dutta \(2000\)](#), [Jiang et al. \(2000\)](#), [Ravi Raju and](#)  
40 [Krishnaiah Chetty \(1993\)](#), to name but a few, also apply CPNs and TPNs to manufacturing

1  
2  
3  
4 systems modeling, analysis and control. An observation on available PN models is that most  
5  
6 researchers have adopted the basic ideas of different extensions of classic PNs and further  
7  
8 extended them to accommodate the different modeling requirements of their own problem  
9  
10 domains. Similarly, in this work, we integrate the advantages of NPNs, TPNs and CPNs to  
11  
12 accommodate the difficulties in modeling production configuration.

### 13 14 15 **3. DEFINITIONS OF NESTED COLORED TIMED PETRI NETS**

16  
17 Production configuration entails a process of configuring a complete production process  
18  
19 for a product along with the product hierarchy (i.e., at different levels of abstraction). Being  
20  
21 located at the highest level, the more abstract production process involving the product's  
22  
23 immediate child items is configured first. Subsequently, the assembly processes and/or  
24  
25 manufacturing processes for producing these immediate child items are configured at the  
26  
27 second level, and so on. The manufacturing processes for parts at the lowest level of each  
28  
29 branch of the hierarchy are last to be configured.

30  
31 In each such process, be it a manufacturing process for a part, an assembly process for an  
32  
33 assembly or the abstract production process for the final product, a number of alternative  
34  
35 machines (including the corresponding tools, fixtures, etc) are able to complete same tasks.  
36  
37 While these machines work on same input items for same tasks, they execute different  
38  
39 operations and, in most cases, incur different cycle times. In practice, it is a common solution  
40  
41 that buffers are adopted to keep raw materials, parts, assemblies and final products.  
42  
43 Furthermore, a number of different types of WIPs (work in processes) are involved in a  
44  
45 process. Unlike a part or assembly that can be located in the BOM (bill of materials) of a  
46  
47 product, a WIP is an intermediate pseudo item formed by sibling parts and/or assemblies and  
48  
49 is not specified in the BOM.

50  
51 Bearing in mind the similarities embedded in processes (e.g., adoption of buffers and  
52  
53 alternative machines, involvement of WIPs), we define a basic PN structure first. Accordingly,  
54  
55 the basic PN structure reflects the generalized common process elements, which are assumed  
56  
57 by different types of nets defined in the proposed formalism.

#### 58 59 **3.1 Basic Net Structure**

60  
*Definition 1:* A basic PN structure is a directed bipartite graph  $G = (P, T, h, A)$ , where  
 $P = P^B \cup P^I \cup P^R \cup P^{CR}$  is a finite set of places with four disjoint subsets:  $P^B$   
representing buffers (be it for raw materials, parts, assemblies, or final products),  $P^I$  items



ready to be processed (be it a raw material, part, assembly, or WIP),  $P^R$  machines, and  $P^{CR}$  the conceptual machines in accordance with alternative machines that can complete same tasks by performing different operations;

$T = T^L \cup T^R \cup T^T$ ,  $P \cap T = \emptyset$  is a finite set of transitions with three disjoint subsets:  $T^L$  denoting a set of logical transitions,  $T^R$  a set of reconfigurable transitions, and  $T^T$  a set of timed transitions;

$h \subseteq P^{CR} \times T^R$  is a finite set of inhibitor arcs that connect a conceptual machine place to a reconfigurable transition and assumes two values: 1 and 0. When  $h(p, t) = 1, \forall p \in P^{CR}, t \in T^R$ , there is a token in the conceptual machine place; the associated reconfigurable transition is disabled and cannot fire. When  $h(p, t) = 0$ , no token resides in the conceptual machine place; the associated reconfigurable transition can fire if it is enabled; and

$A \subseteq (P \times T \cup T \times P) \cap \bar{h}$  is a finite set of arcs excluding inhibitor arcs that connect places/transitions to transitions/places.

Attempting to capture and model multiple alternative machines in relation to same tasks,  $P^{CR}$  is defined in the formalism to represent the corresponding conceptual machines in addition to  $P^R$ , which is defined to denote specific machines. The common practice suggests that only one machine works on same items at one time. Thus, in conjunction with  $P^{CR}$ ,  $h$  and  $T^R$  are introduced to model this situation, where multiple machines can perform same tasks and only one is used eventually. The firing of  $T^R$  leads to the reconfiguration of proper machines. In this way,  $P^{CR}$ ,  $T^R$  and  $h$  can address process variations in system models without rebuilding new ones when machines are added and/or removed.

$T^L$  is defined to capture the logic of system running. Their firing indicates the satisfaction of preconditions of operations, among which the typical one is the presence of material items and machines to be used.  $T^T$  is defined to represent operations, which take certain time durations to complete. Accordingly, the firing of timed transitions incurs time delays. Both logical and reconfigurable transitions are untimed. Their firing is instantaneous and takes 0 time delay.

Figure 1 shows a basic PN structure and the corresponding graphical formalism. Based on the basic PN structure, three types of PNs, namely manufacturing nets (*MNets*), assembly nets (*ANets*) and a process net (*PNet*) are defined to address the granularity issue in production configuration. These nets are defined as a type with a marking, with each type



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

buffer are defined according to the corresponding parts to be produced.

Machines have two statuses - busy and idle. While tokens in idle machine places are defined to represent machines, tokens in busy machine places are defined according to items to be produced. When there is a token residing in the conceptual machine place, the conceptual machine is instantiated to the specific machine represented by the token. A part buffer place contains the set of colored tokens representing the corresponding part family. Following production practice, we define colored tokens of logical and reconfigurable transitions to be the same as those of output places and, similarly, colored tokens of timed transitions as those of input places.

Tielemans (1995) points out that cycle times are a useful performance measure of production. Thus, in this study we adopt cycle times to accommodate selection of proper processes from configured alternatives. Accordingly, time delays,  $\tau$ , are defined in the formalism to represent operations cycle times. Time delays can be obtained from a process platform of a process family in relation to a product family (Zhang, 2007). See (Bowden, 2000) for a comprehensive review of time representations in PN models.

To cope with the difficulties in modeling diverse cycle times associated with multiple machines and same tasks, an arc expression function,  $E$ , is introduced in the modeling formalism. It defines both timed arc expressions and untimed arc expressions. For every arc,  $E$  relates transition colors with multisets of place colors establishing pre- and post-conditions for transition firing. A timed arc expression is a set of antecedent-consequent statements with *XOR* relationships. Each antecedent contains a set of colored tokens with *AND* relationships. The occurrence of each such colored token may not be 1, and by default, the occurrence of 1 is omitted. The consequent includes a colored token to be generated in the busy machine place together with a time delay representing the corresponding operation cycle time. Untimed arc expressions are defined to specify (1) the input tokens for firing transitions; and (2) output tokens after firing transitions.

Figure 2 shows an example of an *MNet*. In this net, the set of places is  $\{p_1, \dots, p_7\}$ ; the set of transitions is  $\{t_1, \dots, t_4\}$ ; the set of inhibitor arcs is  $\{(p_7, t_3), (p_7, t_4)\}$ ; the set of arcs is  $\{(p_1, t_1), (p_2, t_1), \dots, (t_7, p_1)\}$ ; and the set of color sets is  $\{\{a_1, a_2\}, \{b_1, b_2\}, \{A_1, A_2, A_3\}, \{m_1, m_2\}\}$ . Table 1 shows the places, represented system elements and assigned colored tokens.













$$cp = (\Sigma^M, \varepsilon).$$

For example, in the *MNet* in Figure 2, the marking of the three input places of transition,  $t_1$ , is  $\mu(p_1) = \{a_1\}$ ,  $\mu(p_2) = \{b_1\}$  and  $\mu(p_7) = \{m_1\}$ , respectively. In addition,  $E((p_1, t_1), (A_1, \varepsilon)) \subseteq \mu(p_1) = \{a_1\}$ ,  $E((p_2, t_1), (A_1, \varepsilon)) \subseteq \mu(p_2) = \{b_1\}$  and  $E((p_7, t_1), (A_1, \varepsilon)) \subseteq \mu(p_7) = \{m_1\}$ . Thus,  $t_1$  meets the above enabling condition and fires autonomously.

*Rule 2:* A transition,  $t$ , of an *MNet* is enabled and fires simultaneously with an output transition of the place, where the *MNet* is nested, of the higher level *ANet* or *PNet* if  $\forall p, E((p, t), cp) \subseteq \mu(p)$  and  $cp = (\Sigma^M, \Sigma^M)$ .

For example, in Figure 3, the *MNet* is nested in place  $p_9$  of the *ANet* and  $t_6$  is the output transition of  $p_9$ . In the *MNet*, the marking of the input place,  $p_3$ , of the transition,  $t_2$ , is  $\mu(p_3) = \{A_1\}$ . Further,  $E((p_3, t_2), (A_1, A_1)) \subseteq \mu(p_3) = \{A_1\}$ . Thus,  $t_2$  fulfills the simultaneous enabling and firing condition with respect to the transition color pair,  $(A_1, A_1)$ .

#### 4.2.2 Transition enabling in *ANets*

*Rule 3:* A transition,  $t$ , of an *ANet* is enabled with respect to a color triple,  $ct$ , and fires autonomously without additional conditions if  $\forall p, E((p, t), ct) \subseteq \mu(p)$  and  $ct = (\varepsilon, \Sigma^A, \varepsilon)$ .

In Figure 3, if two more tokens,  $B_1$  and  $A_1$ , are added into the two input places,  $p_{10}$  and  $p_{11}$ , of transition,  $t_7$ , of the *ANet*, respectively,  $\mu(p_{10}) = \{B_1\}$ ,  $\mu(p_{11}) = \{A_1\}$  and  $\mu(p_{12}) = \{m_3\}$ . Moreover,  $E((p_{10}, t_7), (\varepsilon, X_1, \varepsilon)) \subseteq \mu(p_{10}) = \{B_1\}$ ,  $E((p_{11}, t_7), (\varepsilon, X_1, \varepsilon)) \subseteq \mu(p_{11}) = \{A_1\}$  and  $E((p_{12}, t_7), (\varepsilon, X_1, \varepsilon)) \subseteq \mu(p_{12}) = \{m_3\}$ . Accordingly,  $t_7$  satisfies the enabling rule and fires autonomously.

*Rule 4:* A transition,  $t$ , of an *ANet* is enabled and fires simultaneously with a transition bearing a  $cp = (\Sigma^A, \Sigma^A)$  or  $ct = (\varepsilon, \Sigma^A, \Sigma^A)$  of the lower level *MNet* or *ANet* that nests in an input place of  $t$ , if  $\forall p, E((p, t), ct) \subseteq \mu(p)$  and  $ct = (\Sigma^A, \Sigma^A, \varepsilon)$ .

In Figure 3,  $t_6$  of the *ANet* meets the firing rule and fires simultaneously with  $t_2$  of the nested *MNet*.

*Rule 5:* A transition,  $t$ , of an *ANet* is enabled and fires simultaneously with an output transition of a place, where the *ANet* is nested, of the higher level *ANet* or *PNet*, if  $\forall p, E((p, t), ct) \subseteq \mu(p)$  and  $ct = (\varepsilon, \Sigma^A, \Sigma^A)$ .

In Figure 3, if a token,  $X_1$ , is added into the input place,  $p_{13}$ , of transition,  $t_8$ , of the  $ANet$ ,  $\mu(p_{13}) = \{X_1\}$  and  $E((p_{13}, t_8), (\varepsilon, X_1, X_1)) \subseteq \mu(p_{13}) = \{X_1\}$ . Thus,  $t_8$  with respect to  $(\varepsilon, X_1, X_1)$  meets the enabling condition and fires simultaneously with other transitions of the host nets.

#### 4.2.3 Transition enabling in $PNet$

*Rule 6:* A transition,  $t$ , of a  $PNet$  is enabled with respect to a color pair,  $cp$ , and fires autonomously without additional conditions if  $\forall p, E((p, t), cp) \subseteq \mu(p)$  and  $cp = (\varepsilon, \Sigma^P)$ .

In the  $PNet$  in Figure 4, if two tokens  $Z_1$  and  $XY_1$  are added into the two input places  $p_{24}$  and  $p_{25}$  of transition  $t_{16}$  of the  $PNet$ , respectively,  $t_{16}$  meets the above enabling condition and fires autonomously. The same can be applied to other transitions, e.g.,  $t_{11}$ ,  $t_{17}$ .

*Rule 7:* A transition,  $t$ , of a  $PNet$  is enabled and fires simultaneously with a transition bearing a  $cp = (\Sigma^M, \Sigma^M)$  or  $ct = (\varepsilon, \Sigma^A, \Sigma^A)$  of the lower level  $MNet$  or  $ANet$  that nests in an input place of  $t$ , if  $\forall p, E((p, t), cp) \subseteq \mu(p)$  and  $cp = (\Sigma^P, \Sigma^P)$ .

In the  $PNet$  in Figure 4, a color pair assigned to  $t_{10}$  is  $(X_1, X_1)$ . Suppose the  $ANet$  in Figure 3 is nested in  $p_{16}$  of the  $PNet$ . If  $t_8$  in the  $ANet$  in Figure 3 is enabled,  $t_{10}$  and  $t_8$  fire simultaneously.

#### 4.2.4 Transition firing

While the firing of transitions of any net does not provoke the transfer of nested nets, that is, the nested nets remain in the same places before and after simultaneous transition firing, tokens, other than the nested nets, are created and removed as follows:

*Rule 8:* The firing of an enabled transition,  $t$ , in nets at any arbitrary level modifies markings by 1) generating tokens in the output places as specified by  $E((t, p), cp)$  or  $E((t, p), ct)$ ; and 2) removing tokens from the input places as specified by  $E((p, t), cp)$  or  $E((p, t), ct)$ .

For instance, in Figure 3, the firing of  $t_6$  generates a token (either  $A_1$ ,  $A_2$  or  $A_3$ ) in  $P_{11}$ . While the  $MNet$  remains in  $P_9$  after firing  $t_2$ , it evolves into a new state, i.e., a token is generated in  $P_4$  and a token is removed from the input place  $P_5$ .















1  
2  
3  
4 interactive picture of higher level more abstract processes and lower level more detailed  
5  
6 processes.

7  
8 2) The nested *ANets* and *MNets* in the proposed formalism are independent of one  
9  
10 another. There is no direct connection between two nets that are nested in different places  
11  
12 of nets at a same level of the configuration hierarchy. As a result, one nested net can be  
13  
14 changed without causing any structural changes to other nets. Hence, it is easy for  
15  
16 managers to maintain and modify the production configuration models based on the  
17  
18 proposed NCTPNs.

19  
20 3) As a graphical modeling tool, the NCTPNs provide managers with a visualization  
21  
22 of the entire process of production configuration. Transaction synchronization is used to  
23  
24 capture the interactions between lower level nested nets and higher level host nets. Thus,  
25  
26 the impact on the relevant higher level nets caused by the changes of lower level nets, be  
27  
28 it a change of net structures or of process elements configured, can be intuitively captured.  
29  
30 This, in turn, offers an easier understanding of the impacts caused by configuration  
31  
32 changes on production configuration to managers.

33  
34 4) NCTPNs can be used to facilitate the development of a production rule-based  
35  
36 computational system which implements production configuration. There are some  
37  
38 correspondence between NCTPNs and rule-based systems. The tokens carrying important  
39  
40 information pertaining to the represented objects are equivalent to facts in a rule-based  
41  
42 system. Transaction firing rules relate to rules in a rule-based system. Transition firing  
43  
44 corresponds to rule execution in a computational system. Therefore, the analogy between  
45  
46 the NCTPNs and rule-based systems can facilitate the development of a computational  
47  
48 system to implement production configuration.

49  
50 The scope of this study is limited to the development of a proper formalism to model  
51  
52 production configuration, and does not extend to the computational implementation of  
53  
54 production configuration based on the proposed model. In addition, this study does not  
55  
56 address the impact of the different combinations of production objectives and transition firing  
57  
58 rules on the final configuration results in that different companies may adopt different  
59  
60 combinations according to their own production characteristics. Accordingly, future research  
may be extended to address them. And more efforts might be made to develop a rule-based

1  
2  
3  
4 computational system to implement production configuration based on the NCTPNs.  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

For Peer Review Only

## REFERENCES

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Abdallaht, I.B., Elmaraghyj, H.A. and Elmekawyi, T., 2002, Deadlock-free scheduling in flexible manufacturing systems using Petri nets, *International Journal of Production Research*, **40**(12), pp. 2733-2756.

Adamous, M., Zerhouni, S.N. and Bourjault, A., 1998, Hierarchical modeling and control of flexible assembly systems using object-oriented Petri nets, *International Journal of Computer Integrated Manufacturing*, **11**(1), pp. 18–33.

Blackhurst, J., Wu, T. and O’Grady, P., 2005, PCDM: A decision support modeling methodology for supply chain, product and process design decisions, *Journal of Operations Management*, **23**, pp. 325-343.

Bowden, F.D.J., 2000, A brief survey and synthesis of the roles of time in Petri nets, *Mathematical and Computer Modeling*, **31**, pp. 55-68.

Chan, F.T.S., Zhang, J. and Li, P., 2001, Modeling of integrated, distributed and cooperative process planning system using an agent-based approach, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, **215**, pp. 1437-1451.

Chin, K.S., Zu, X., Mok, C.K. and Tam, H.Y., 2006, Integrated definition language 0 (IDEF) and colored Petri nets (CPN) modeling and simulation tool: A study on mould-making processes, *International Journal of Production Research*, **44**(15), pp. 3179-3205.

De Lit, P., Delchambre, A. and Henrioud, J.M., 2003, An integrated approach for product family and assembly system design, *IEEE Transactions on Robotics and Automation*, **19**, pp. 324–333.

Dotoli, M. and Fanti, M.P., 2004, Colored timed Petri net model for real-time control of automated guided vehicle systems, *International Journal of Production Research*, **42**(9), pp. 1787-1814.

Fixson, S.K., 2005, Product architecture assessment: A tool to link product, process, and supply chain design decisions, *Journal of Operations Management*, **23**, pp. 345-369.

Garg, A., 1999, An application of designing products and processes for supply chain management, *IIE Transactions*, **31**, pp. 417-429.

Gupta, S. and Krishnan, V., 1998, Product family-based assembly sequence design methodology, *IIE Transactions*, **30**, pp. 933–945.

Hiraishi, K., 2000, A Petri-net-based model for the mathematic analysis of multi-agent systems, *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC’2000)*, Oct. 2000, Nashville, TN, 4, pp. 3009–3014.

Huang, G.Q., Zhang, X.Y. and Liang, L., 2005, Towards integrated optimal configuration of platform products, manufacturing processes, and supply chains, *Journal of Operations Management*, **23**, pp. 267-290.

Huang, S.H., Liu, Q. and Musa, R., 2004, Tolerance-based process plan evaluation using Monte Carlo simulation, *International Journal of Production Research*, **42**, pp. 4871-4891.

Jensen, K., 1995, *Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Vol. 2, New York: Springer.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Jiang, Z., Zuo, M.J., Fung, R.Y.K. and Tu, P.Y., 2000, Temporized colored Petri nets with changeable structure (CPN-CS) for performance modeling of dynamic production systems, *International Journal of Production Research*, **38**, pp. 1917–1945.

Jiang, Z., Zuo, M.J., Tu, P.Y. and Fung, R.Y.K., 1999, Object-oriented Petri nets with changeable structure (OPNs-CS) for production system modeling, *International Journal of Advanced Manufacturing Technology*, **15**, pp. 445-458.

Jiao, J., Zhang, L. and Pokharel, S., 2007, Process platform planning for variety coordination from design to production in mass customization manufacturing, *IEEE Transactions on Engineering Management*, **54**, pp. 112-129.

Kaighobadi, M. and Venkatesh, K., 1994, Flexible manufacturing systems: An overview, *International Journal of Operations & Production Management*, **14**(4), pp. 25-49.

Li, X.O. and Lara-Rosano, F., 1999, Modeling an electronic component manufacturing system using object oriented colored Petri nets, *Proceedings of the 6th IEEE International Conference on Electronics, Circuits, and Systems*, Pafos, Cyprus.

Liu, D.S., Wang, J.M., Chan, S.C.F., Sun, J.G. and Zhang, L., 2002, Modeling workflow processes with colored Petri nets, *Computers in Industry*, **49**(3), pp. 267-281.

Lomazova, I.A., 2000, Nested Petri nets: A formalism for specification and verification of multi-agent distributed systems, *Fundamenta Informaticae*, **43**, pp.195-214.

Lopez-Mellado, E. and Almeyda-Canepa, H., 2003, A three-level net formalism for the modeling of multiple mobile robot systems, *International Journal of Computer Integrated Manufacturing*, **18**(2-3), pp. 122–136.

Lu, Q. and Botha, B., 2006, Process development: A theoretical framework, *International Journal of Production Research*, **44**(15), pp. 2977-2996.

Martinez, M.T., Favrel, J. and Ghodous, P., 2000, Product family manufacturing plan generation and classification, *Concurrent Engineering: Research and Applications*, **8**, pp. 12-22.

Nandula, M. and Dutta, S.P., 2000, Performance evaluation of an auction-based manufacturing system using colored Petri nets, *International Journal of Production Research*, **38**, pp. 2155–2171.

Peterson, J.L., 1981, *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood Cliffs, N.J.

Petersen, K.J., Handfield, R.B. and Ragatz, G.L., 2005, Supplier integration into new product development: Coordinating product, process and supply chain design, *Journal of Operations Management*, **23**, pp. 371-388.

Ramachandani, C., 1974, Analysis of asynchronous concurrent systems by timed Petri nets. Technical Report MAC TR 120, MIT, Cambridge.

Raman, N. and Chhajer, D., 1995, Simultaneous determination of product attributes and prices, and production processes in product-line design, *Journal of Operations Management*, **12**, pp. 187-204.

Ravi Raju, K. and Krishnaiah Chetty, O.V., 1993, Design and evaluation of automated guided vehicle systems for flexible manufacturing systems: An extended timed Petri net-based approach, *International Journal of Production Research*, **31**(5), pp. 1069-1096.

Reddy, G.B., Murty, S.S.N. and Ghosh, K., 1993, Timed Petri net: An expeditious tool for modeling and analysis of manufacturing systems, *Mathematical and Computer Modeling*, **18**(9), pp. 17-30.

1  
2  
3 Sanderson, S.W. and Uzumeri, A.V., 1996, *Managing Product Families*. Irwin Press.

4 Schierholt, K., 2001, Process configuration: Combining the principles of product  
5 configuration and process planning, *Artificial Intelligence for Engineering Design, Analysis*  
6 *and Manufacturing*, **15**, pp. 411-424.

7  
8 Thomas, J.P., Nissanke, N. and Baker, K.D., 1996, A hierarchical Petri net framework for  
9 the representation and analysis of assembly, *IEEE Transactions of Robotics and Automation*,  
10 **12**(2), pp. 268–279.

11  
12 Tielemans, P.F.J., 1995, Lead Time Performance in Manufacturing Systems. PhD  
13 Dissertation, Rotterdam School of Management, Eburon, Delft, The Netherlands.

14  
15 Turner, W.C., Mize, J.H., Caser, K.E. and Nazemetz, J.W., 1992, *Introduction to*  
16 *Industrial and Systems Engineering*, 3rd Edition. Prentice Hall.

17  
18 Wang, L.C. and Wu, S.Y., 1998, Modeling with colored timed object-oriented Petri nets for  
19 automated manufacturing systems, *Computers & Industrial Engineering*, **34**(2), pp. 463-480.

20  
21 Williams, C.B., Allen, J.K., Rosen, D.W. and Mistree, F., 2007, Designing platforms for  
22 customizable products and processes in markets of non-uniform demand, *Concurrent*  
23 *Engineering: Research and Applications*, **15**, pp. 201-216.

24  
25 Zhang, L., 2007, Process Platform-based Production Configuration for Mass  
26 Customization. PhD Dissertation, Division of Systems and Engineering Management,  
27 Nanyang Technological University, Singapore.

## LIST OF TABLES

Table 1: Places, system elements and colored tokens in the net in Figure 2

Places	System Elements	Colored Tokens
$p_1$	A raw material buffer	$\alpha(p_1) = \{a_1, a_2\}$
$p_2$	A raw material buffer	$\alpha(p_2) = \{b_1, b_2\}$
$p_3$	A busy machine	$\alpha(p_3) = \{A_1, A_2, A_3\}$
$p_4$	A part buffer	$\alpha(p_4) = \{A_1, A_2, A_3\}$
$p_5$	An idle machine	$\alpha(p_5) = \{m_1\}$
$p_6$	An idle machine	$\alpha(p_6) = \{m_2\}$
$p_7$	A conceptual machine	$\alpha(p_7) = \{m_1, m_2\}$

Table 2: Places, system elements and colored tokens in the net in Figure 3

Places	System Elements	Nets/Colored Tokens
$p_8$	An input part buffer	An <i>MNet</i> for producing part family A
$p_9$	An input part buffer	An <i>MNet</i> for producing part family B
$p_{10}$	Parts ready to be processed	$\alpha(p_{10}) = \{B_1, B_2, B_3\}$
$p_{11}$	Parts ready to be processed	$\alpha(p_{11}) = \{A_1, A_2, A_3\}$
$p_{12}$	An idle machine	$\alpha(p_{12}) = \{m_3\}$
$p_{13}$	A busy machine	$\alpha(p_{13}) = \{X_1, X_2, X_3\}$
$p_{14}$	An assembly buffer	$\alpha(p_{14}) = \{X_1, X_2, X_3\}$

Table 3: Places, system elements and the colored tokens/nets in the net in Figure 4

Places	System Elements	Nets/Colored Tokens
$p_{15}$	An input assembly buffer	An <i>ANet</i> for producing assembly family Y
$p_{16}$	An input assembly buffer	An <i>ANet</i> for producing assembly family X
$p_{17}$	Assemblies ready to be processed	$\alpha(p_{17}) = \{Y_1, Y_2\}$
$p_{18}$	Assemblies ready to be processed	$\alpha(p_{18}) = \{X_1, X_2, X_3\}$
$p_{19}$	A conceptual machine	$\alpha(p_{19}) = \{m_4, m_5\}$
$p_{20}$	An idle machine	$\alpha(p_{20}) = \{m_4\}$
$p_{21}$	An idle machine	$\alpha(p_{21}) = \{m_5\}$
$p_{22}$	A busy machine	$\alpha(p_{22}) = \{XY_1, XY_2, XY_3\}$
$p_{23}$	An input part buffer	An <i>MNet</i> for producing part family Z
$p_{24}$	Parts ready to be processed	$\alpha(p_{24}) = \{Z_1, Z_2\}$
$p_{25}$	WIPs formed by two assembly families	$\alpha(p_{25}) = \{XY_1, XY_2, XY_3\}$
$p_{26}$	A busy machine	$\alpha(p_{26}) = \{FP_1, FP_2, FP_3\}$
$p_{27}$	An idle machine	$\alpha(p_{27}) = \{m_6\}$
$p_{28}$	An end-product buffer	$\alpha(p_{28}) = \{FP_1, FP_2, FP_3\}$

Table 4: The places, represented system elements and colored tokens/nets

Places	System Elements	Nets/Colored Tokens
$P_{1/4/11}$	Input assembly buffers for AAs/BAs/FAs	$ANets$ for producing assembly families AA/BA/FA
$P_2$	AAs are ready to be processed	$\alpha(p_2) = \{AA_1, AA_2, AA_3\}$
$P_3$	BAs are ready to be processed	$\alpha(p_2) = \{BA_1, BA_2, BA_3\}$
$P_{5/13}$	Conceptual machines for assembling ABs/MBs	$\alpha(p_5) = \{m_1, m_2, m_3\} / \alpha(p_{13}) = \{m_4, m_5\}$
$P_{6/7/8}$	Alternative idle machines for forming ABs	$\alpha(p_6) = \{m_1\} / \alpha(p_7) = \{m_2\} / \alpha(p_8) = \{m_3\}$
$P_9$	Alternative machines processing AAs & BAs	$\alpha(p_9) = \{AB_1, AB_2, AB_3\}$
$P_{10}$	ABs are ready to be processed	$\alpha(p_{10}) = \{AB_1, AB_2, AB_3\}$
$P_{12}$	FAs are ready to be processed	$\alpha(p_{12}) = \{FA_1, FA_2\}$
$P_{14/15}$	Alternative idle machines for assembling MBs	$\alpha(p_{14}) = \{m_4\} / \alpha(p_{15}) = \{m_5\}$
$P_{16}$	Alternative machines processing ABs & FAs	$\alpha(p_{16}) = \{MB_1, MB_2, MB_3\}$
$P_{17}$	MBs are ready to be processed	$\alpha(p_{17}) = \{MB_1, MB_2, MB_3\}$
$P_{18}$	A buffer for Wts and RhS	$\alpha(p_{18}) = \{Wt_1, Wt_2, Rh_1, Rh_2\}$
$P_{19}$	A workstation for assembling final motors	$\alpha(p_9) = \{m_6\}$
$P_{20}$	Final motors are being assembled at the workstation	$\alpha(p_{20}) = \{VM_1, VM_2, VM_3\}$
$P_{21}$	A buffer for final motors	$\alpha(p_{21}) = \{VM_1, VM_2, VM_3\}$

Table 5: The places, represented elements and colored tokens/nets

Places	System Elements	Nets/Colored Tokens
$P_{22/25/31}$	Input part buffers for Bas/Bbs/Tls	$MNets$ for producing part families Ba/Bb/Tl
$P_{23}$	Bas are ready to be processed	$\alpha(p_{23}) = \{Ba_1, Ba_2, Ba_3\}$
$P_{24}$	Bbs are ready to be processed	$\alpha(p_{24}) = \{Bb_1, Bb_2, Bb_3\}$
$P_{26/33}$	Conceptual machines for assembling Babs/BAs	$\alpha(p_{26}) = \{m_7, m_8\} / \alpha(p_{33}) = \{m_9, m_{10}, m_{11}\}$
$P_{27/28}$	Alternative idle machines for forming ABs	$\alpha(p_6) = \{m_1\} / \alpha(p_7) = \{m_2\} / \alpha(p_8) = \{m_3\}$
$P_{29}$	Alternative machines processing Bas & Bbs	$\alpha(p_{29}) = \{Bab_1, Bab_2, Bab_3\}$
$P_{30}$	Babs are ready to be processed	$\alpha(p_{30}) = \{Bab_1, Bab_2, Bab_3\}$
$P_{32}$	Tls are ready to be processed	$\alpha(p_{32}) = \{Tl_1, Tl_2\}$
$P_{34/35/36}$	Alternative idle machines for producing BAS	$\alpha(p_{34}) = \{m_9\} / \alpha(p_{35}) = \{m_{10}\} / \alpha(p_{36}) = \{m_{11}\}$
$P_{37}$	Alternative machines processing Babs & Tls	$\alpha(p_{37}) = \{BA_1, BA_2, BA_3\}$
$P_{38}$	A buffer for BAS	$\alpha(p_{38}) = \{BA_1, BA_2, BA_3\}$

Table 6: The places, represented elements and colored tokens/nets

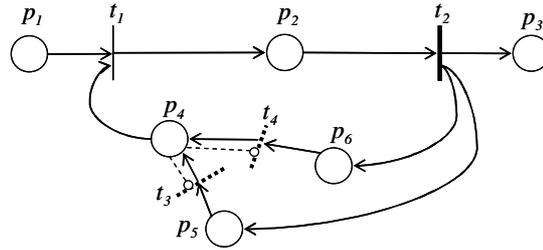
Places	System Elements	Colored Tokens
$P_{39}$	A raw material buffer for Bas	$\alpha(p_{39}) = \{Ba_1,1, Ba_2,1, Ba_3,1\}$
$P_{40}$	A conceptual machines for manufacturing Bas	$\alpha(p_{40}) = \{m_{12}, m_{13}\}$
$P_{41/42}$	Alternative idle machines for manufacturing Bas	$\alpha(p_{41}) = \{m_{12}\} / \alpha(p_{42}) = \{m_{13}\}$
$P_{43}$	Alternative machines processing Ba raw materials	$\alpha(p_{43}) = \{Ba_1,2, Ba_2,2, Ba_3,2\}$
$P_{44}$	Ba WIPs are ready to be processed	$\alpha(p_{44}) = \{Ba_1,2, Ba_2,2, Ba_3,2\}$
$P_{45}$	The second machine for manufacturing Bas	$\alpha(p_{45}) = \{m_{14}\}$
$P_{46}$	The second machine processing Ba WIPs	$\alpha(p_{46}) = \{Ba_1, Ba_2, Ba_3\}$
$P_{47}$	A buffer for Bas	$\alpha(p_{47}) = \{Ba_1, Ba_2, Ba_3\}$

Table 7: The configured production processes for the 3 motor variants

Machine(Operation; Output item)		
$VM_1$	$VM_2$	$VM_3$
$m_{13}$ (Manufacturing operation; $Ba_1$ )	$m_{13}$ (Manufacturing operation; $Ba_2$ )	$m_{12}$ (Manufacturing operation; $Ba_3$ )
$m_{14}$ (Manufacturing operation; $Ba_1$ )	$m_{14}$ (Manufacturing operation; $Ba_2$ )	$m_{14}$ (Manufacturing operation; $Ba_2$ )
$m_{23}$ (Manufacturing operation; $Bb_1$ )	$m_{22}$ (Manufacturing operation; $Bb_2$ )	$m_{23}$ (Manufacturing operation; $Bb_3$ )
$m_{24}$ (Manufacturing operation; $Bb_1$ )	$m_{24}$ (Manufacturing operation; $Bb_2$ )	$m_{24}$ (Manufacturing operation; $Bb_3$ )
$m_{26}$ (Manufacturing operation; $Tl_1$ )	$m_{25}$ (Manufacturing operation; $Tl_2$ )	$m_{25}$ (Manufacturing operation; $Tl_2$ )
$m_7$ (Assembly operation; $BA_1$ )	$m_8$ (Assembly operation; $Bab_2$ )	$m_7$ (Assembly operation; $Bab_3$ )
Nonexist	$m_{10}$ (Assembly operation; $BA_2$ )	$m_{10}$ (Assembly operation; $BA_3$ )
$m_{27}$ (Manufacturing operation; $Tp_1$ )	$m_{28}$ (Manufacturing operation; $Tp_2$ )	$m_{27}$ (Manufacturing operation; $Tp_1$ )
$m_{29}$ (Manufacturing operation; $Cl_1$ )	$m_{29}$ (Manufacturing operation; $Cl_1$ )	$m_{29}$ (Manufacturing operation; $Cl_2$ )
$m_{30}$ (Manufacturing operation; $Cl_1$ )	$m_{30}$ (Manufacturing operation; $Cl_1$ )	$m_{30}$ (Manufacturing operation; $Cl_2$ )
$m_{21}$ (Assembly operation; $CA_1$ )	$m_{20}$ (Assembly operation; $CA_2$ )	$m_{20}$ (Assembly operation; $CA_3$ )
$m_{16}$ (Assembly operation; $AA_1$ )	$m_{15}$ (Assembly operation; $AA_2$ )	$m_{16}$ (Assembly operation; $AA_1$ )
$m_{18}$ (Assembly operation; $FA_1$ )	$m_{17}$ (Assembly operation; $FA_2$ )	$m_{19}$ (Assembly operation; $FA_3$ )
$m_1$ (Assembly operation; $AB_1$ )	$m_3$ (Assembly operation; $AB_2$ )	$m_2$ (Assembly operation; $AB_3$ )
$m_5$ (Assembly operation; $MB_1$ )	$m_5$ (Assembly operation; $MB_2$ )	$m_4$ (Assembly operation; $MB_3$ )
$m_6$ (Assembly operation; $VM_1$ )	$m_6$ (Assembly operation; $VM_2$ )	$m_6$ (Assembly operation; $VM_3$ )

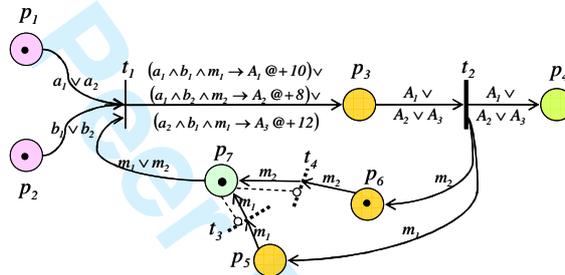


A LIST OF FIGURES



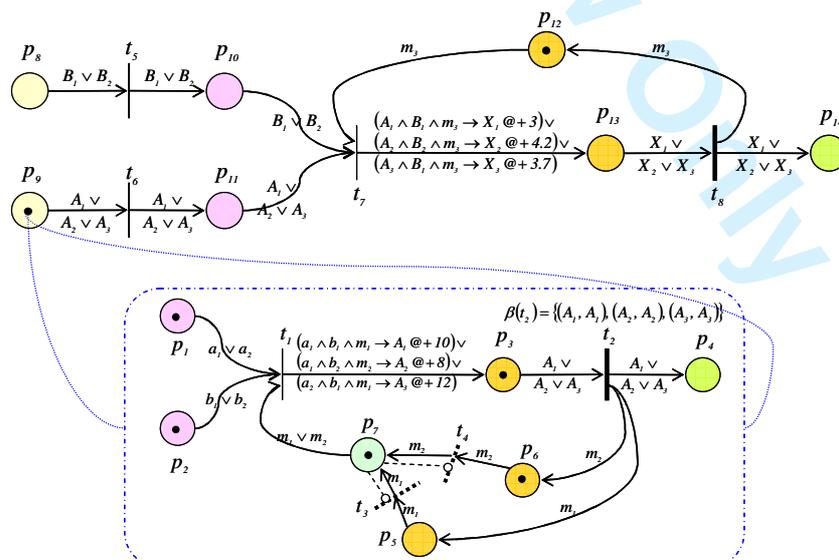
○ : Place; -----○ : Inhibitor arc; — : Logical transition  
 → : Arc; — : Timed transition; ..... : Reconfigurable transition

Figure 1: A basic PN structure and the graphical formalism



$\beta(t_1) = \{(A_1, \varepsilon), (A_2, \varepsilon), (A_3, \varepsilon)\}; \beta(t_2) = \{(A_1, A_1), (A_2, A_2), (A_3, A_3)\}; \beta(t_3) = \{(m_1, \varepsilon)\}; \beta(t_4) = \{(m_2, \varepsilon)\}$   
 $E((p_1, t_1), (A_1, \varepsilon)) = a_1; E((p_1, t_1), (A_2, \varepsilon)) = a_2; E((p_2, t_1), (A_1, \varepsilon)) = b_1; E((p_2, t_1), (A_2, \varepsilon)) = b_2; E((p_2, t_1), (A_3, \varepsilon)) = b_3; E((t_3, p_7), (m_1, \varepsilon)) = m_1; E((t_4, p_7), (m_2, \varepsilon)) = m_2$   
 $E((t_1, p_3), (A_1, \varepsilon)) = a_1 \wedge b_1 \wedge m_1 \rightarrow A_1 @ +10; E((t_1, p_3), (A_2, \varepsilon)) = a_1 \wedge b_2 \wedge m_2 \rightarrow A_2 @ +8$   
 $E((t_1, p_3), (A_3, \varepsilon)) = a_2 \wedge b_1 \wedge m_1 \rightarrow A_3 @ +12; E((p_2, t_2), (A_1, A_1)) = A_1; E((p_3, t_2), (A_2, A_2)) = A_2$   
 $E((p_3, t_2), (A_3, A_3)) = A_3; E((t_3, p_4), (A_1, A_1)) = A_1; E((t_3, p_4), (A_2, A_2)) = A_2; E((t_3, p_4), (A_3, A_3)) = A_3$   
 $E((t_2, p_5), (A_1, \varepsilon)) = m_1; E((t_2, p_5), (A_2, \varepsilon)) = m_2; E((t_2, p_5), (A_3, \varepsilon)) = m_3; E((t_2, p_6), (A_2, A_2)) = m_2; E((p_5, t_3), (m_1, \varepsilon)) = m_1$   
 $E((p_6, t_4), (m_2, \varepsilon)) = m_2; E((p_7, t_1), (A_1, \varepsilon)) = m_1; E((p_7, t_1), (A_2, \varepsilon)) = m_2; E((p_7, t_1), (A_3, \varepsilon)) = m_3$

Figure 2: An MNet



$\beta(t_5) = \{(B_1, B_1, \varepsilon), (B_2, B_2, \varepsilon)\}; \beta(t_6) = \{(A_1, A_1, \varepsilon), (A_2, A_2, \varepsilon), (A_3, A_3, \varepsilon)\}; \beta(t_7) = \{(\varepsilon, X_1, \varepsilon), (\varepsilon, X_2, \varepsilon), (\varepsilon, X_3, \varepsilon)\}; \beta(t_8) = \{(\varepsilon, X_1, X_1), (\varepsilon, X_2, X_2), (\varepsilon, X_3, X_3)\}$

Figure 3: An ANet including the nested MNet

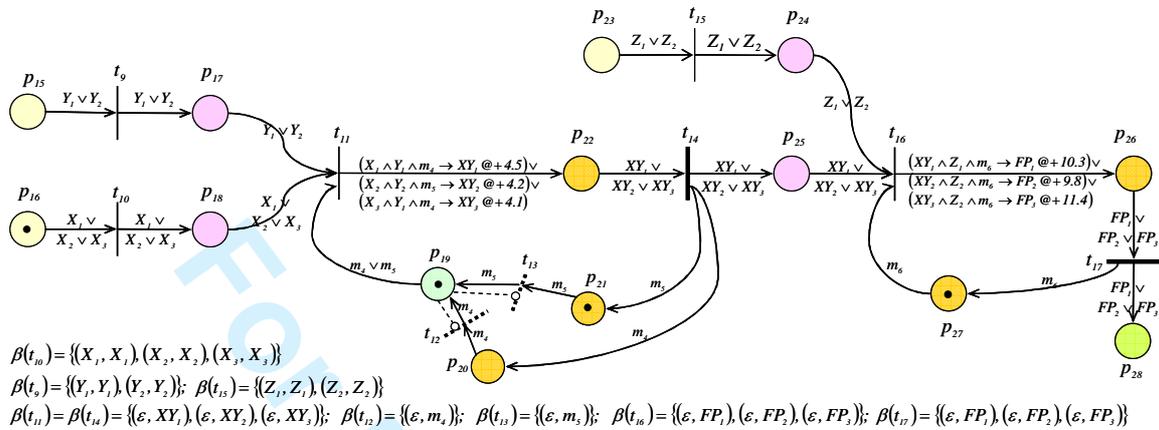


Figure 4: A PNet

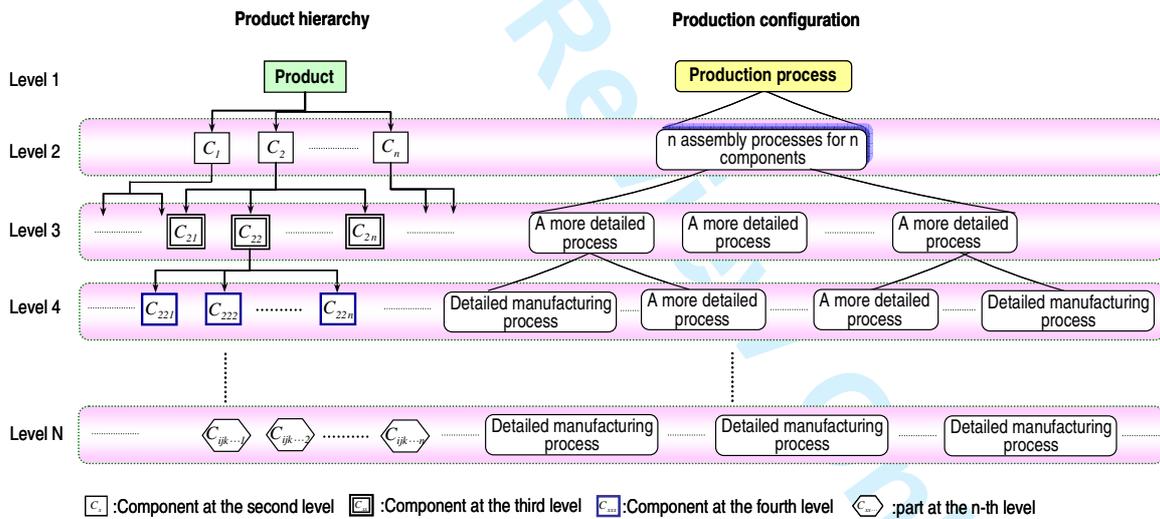


Figure 5: Parallelism between product hierarchy and production configuration

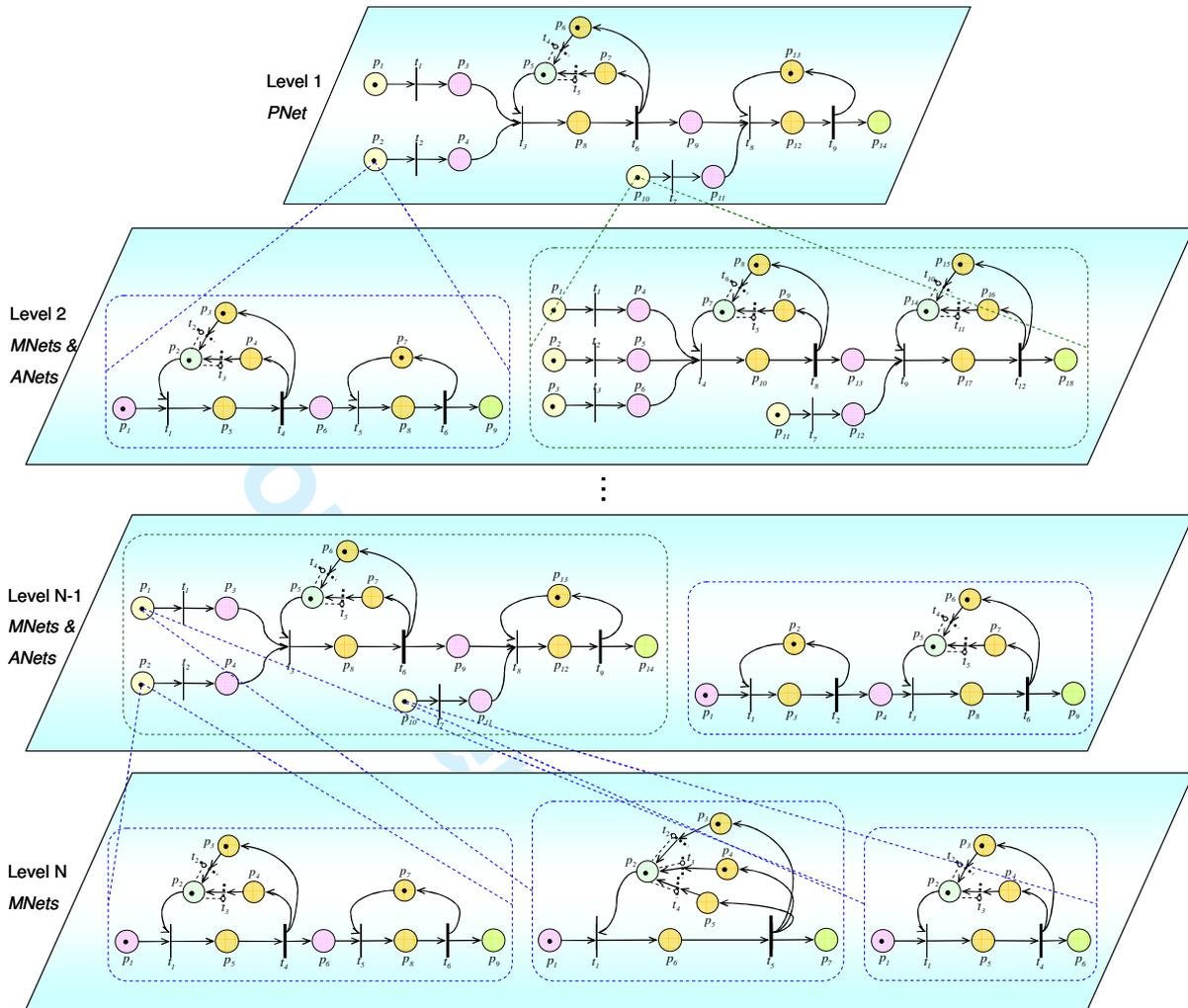


Figure 6: An example of multilevel nested net system

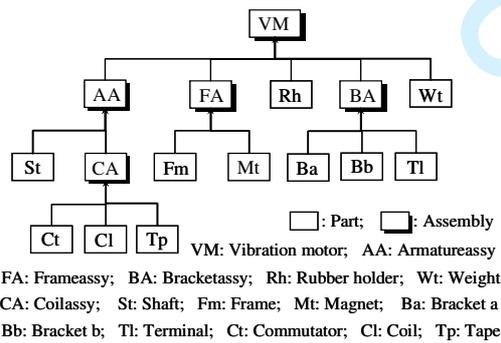


Figure 7: The common product structure of the motor family

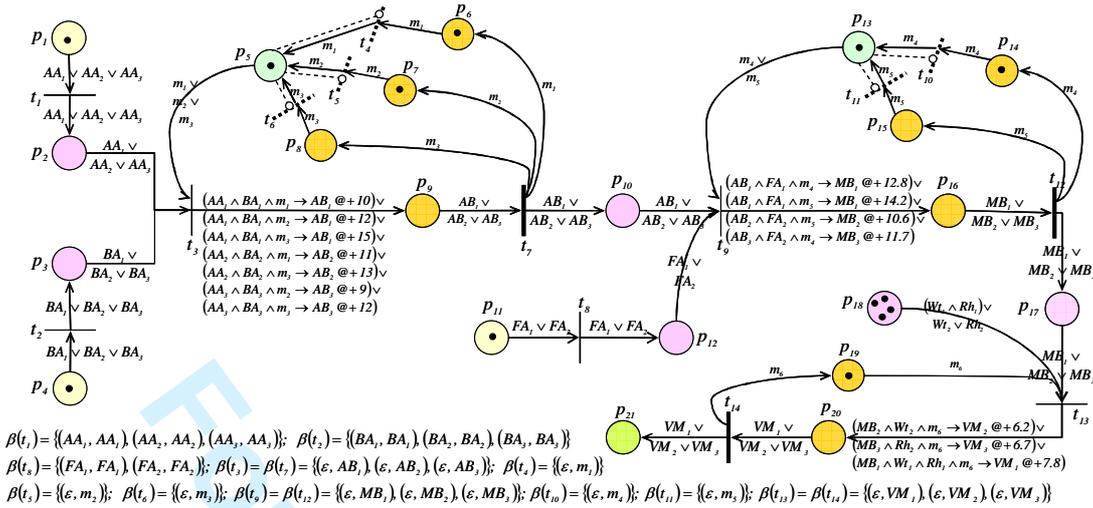


Figure 8: The PNet of the motor family

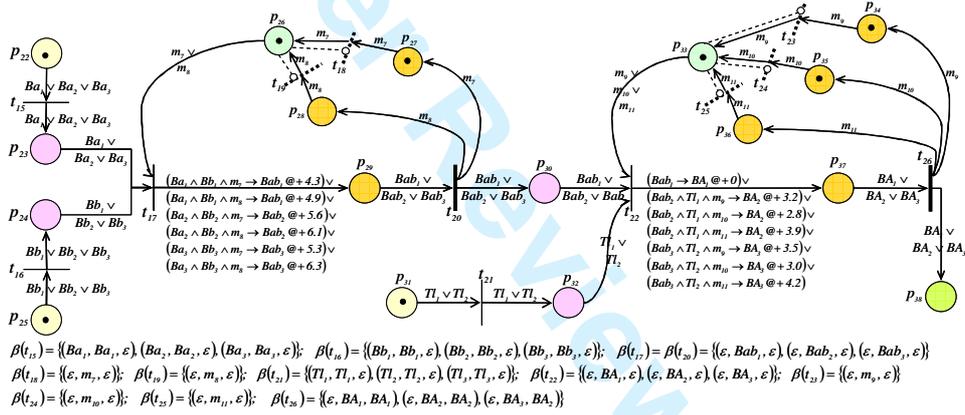


Figure 9: The ANet of the bracketassy family

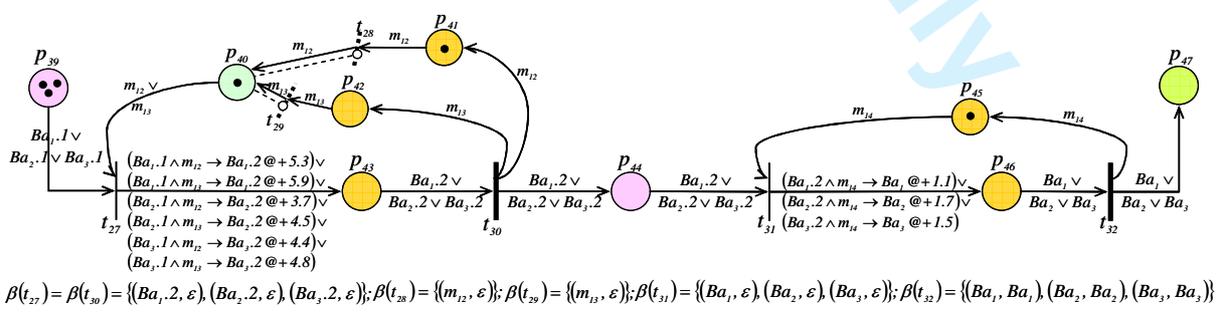


Figure 10: The MNet of the bracket a family

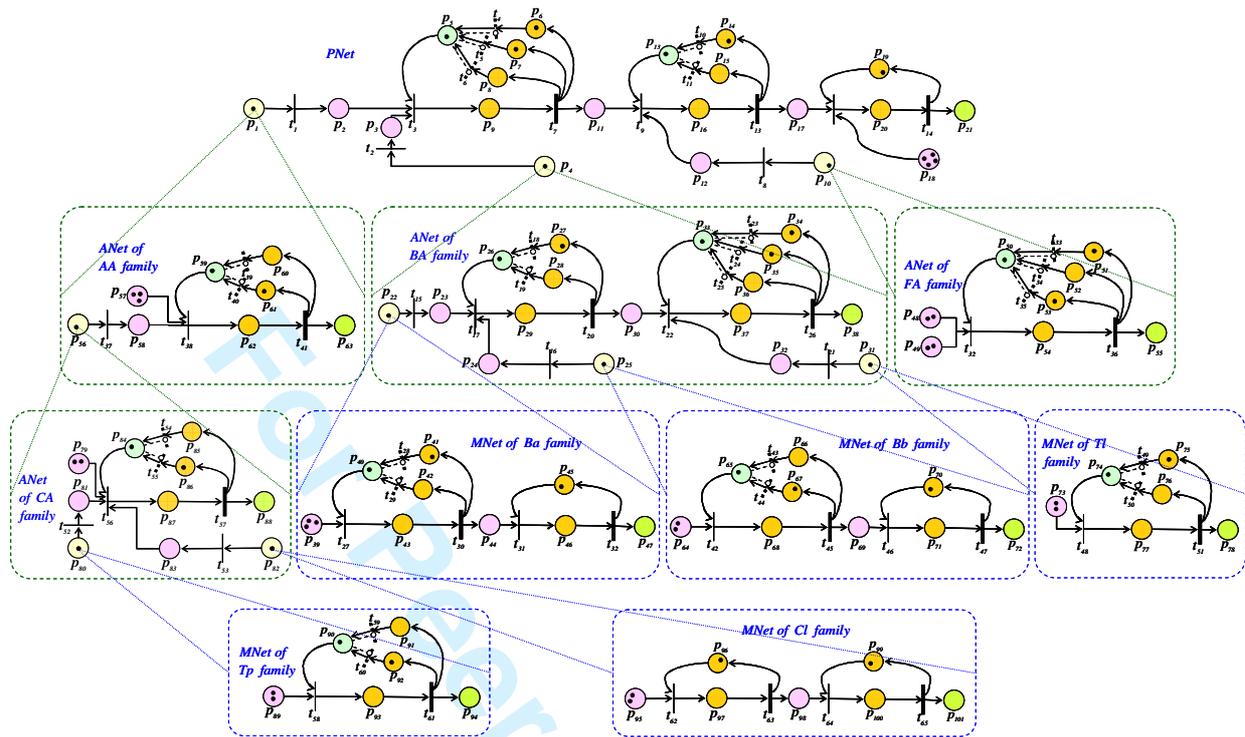


Figure 11: The multilevel nested net system of production configuration

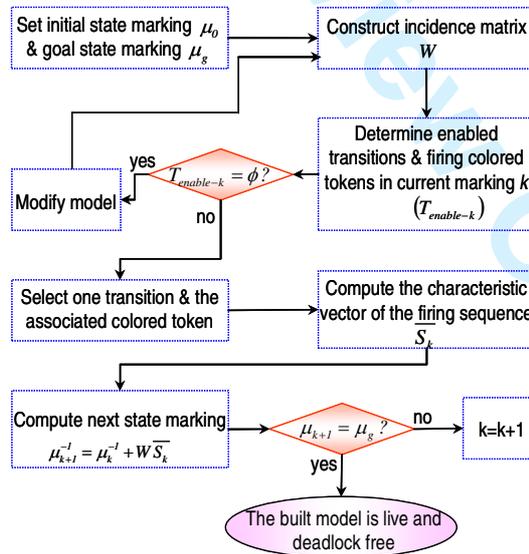


Figure 12: Procedure of deadlock detection algorithm (adapted from Wang and Wu, 1998)

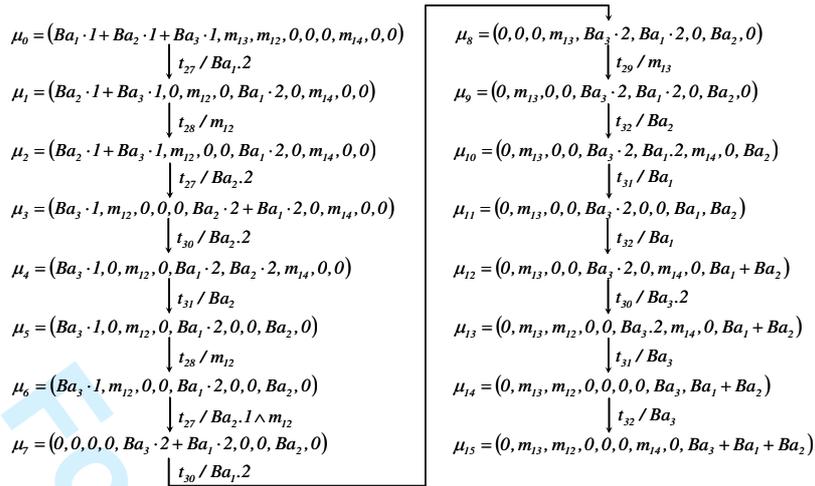


Figure 13: A feasible firing sequence leading to the goal state