

University of Groningen

Modeling production configuration using nested colored object-oriented Petri-nets with changeable structures

Zhang, Lianfeng (Linda); Jiao, Jianxin (Roger)

Published in:
Journal of Intelligent Manufacturing

DOI:
[10.1007/s10845-008-0111-7](https://doi.org/10.1007/s10845-008-0111-7)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2009

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Zhang, L. L., & Jiao, J. R. (2009). Modeling production configuration using nested colored object-oriented Petri-nets with changeable structures. *Journal of Intelligent Manufacturing*, 20(4), 359-378.
<https://doi.org/10.1007/s10845-008-0111-7>

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

Modeling production configuration using nested colored object-oriented Petri-nets with changeable structures

Lianfeng (Linda) Zhang · Jianxin (Roger) Jiao

Received: 9 October 2007 / Accepted: 7 May 2008 / Published online: 27 July 2008
© The Author(s) 2008

Abstract Configuring production processes based on process platforms has been well recognized as an effective means for companies to provide product variety while maintaining mass production efficiency. The production processes of product families involve diverse variations in manufacturing and assembly processes resulted from a large variety of component parts and assemblies. This paper develops a multilevel system of nested colored object-oriented Petri nets with changeable structures to model the configuration of production processes. To capture the semantics associated with production configuration decisions, some unique modeling mechanisms are employed, including colored Petri nets, object-oriented Petri nets, changeable Petri net structures, and net nesting. The modeling formalism comprises resource nets, manufacturing nets, assembly nets and process nets. The paper demonstrates how these net definitions are applied to the specification of production process variants at different levels of abstraction. Also reported is a case study in an electronics company. The system model is further analyzed with focus on conflict prevention and deadlock detection.

Keywords Process platform · Production configuration · Colored Petri net · Object-oriented Petri net · Net nesting

Introduction

Developing product families and platforms has been well recognized as a means for companies to provide sufficient variety with mass production efficiency (Simpson 2004; Jiao et al. 2007). While seeking technical solutions is the major concern in design, it is at the production stage that product costs are actually committed and product quality and lead times are determined *per se*. For a given design, the actual cost depends on how the production is planned and to what extent the economy of scale can be realized within the existing manufacturing capabilities. This implies that the claimed rationale of product family design can only be fulfilled at the production stage (Jiao and Tseng 2004).

Product differentiation inevitably leads to an exponentially increased number of process variations, involving machines, tools, fixtures, setups, cycle times, and labor (Wortmann et al. 1997). Nonetheless, the common components and basic product structures embedded in product variety inherently enable similarity and thus reusability in the corresponding production processes (Martinez et al. 2000). Therefore, companies are interested in configuring existing operations and manufacturing processes to take advantage of repetitions (Schierholt 2001). Besides leveraging the cost of delivering variety, exploiting process families around process platforms can reduce development risks by reusing proven elements and knowledge in a firm's activities (Sawhney 1998).

Production configuration entails a conceptual structure and an overall logical organization of producing a family of customized products. It provides a generic umbrella to capture and utilize commonality, within which each new product fulfillment is instantiated and extended so as to anchor production planning to a common process structure (Martinez et al. 2000). The common process structure contains such process elements as abstract process concepts for

L. Zhang
Department of Operations, University of Groningen,
Groningen, The Netherlands

J. Jiao (✉)
School of Mechanical and Aerospace Engineering, Nanyang
Technological University, Singapore, Singapore
e-mail: jjiao@ieee.org

parts and assemblies in relation to product families, operations, estimated cycle times, as well as manufacturing resources. Within a process platform, proper process elements are selected and arranged to form an optimal production process for each individual product in the family. Decisions regarding production configuration are deemed to be complicated. In particular, a number of fundamental issues need to be addressed, as elaborated below.

Challenges for production configuration

Variety handling

Individual customer expectations lead to a large number of product items, such as diverse parts and assemblies. These product items are specified by various design parameters in conjunction with particular value instances. On the shop floor, producing such a variety of parts, assemblies and final products necessitates distinct processes, operations and manufacturing resources. It is imperative to deal with the large number of product and process elements while leveraging upon existing product families and process platforms.

Process change

In spite of similarity inherent in product and process families, each product is fulfilled by using a unique production process. Production processes of different products may differ from one another in specific operations, manufacturing resources, and process flows. In addition, within a company's manufacturing capabilities, there may exist a number of alternative production processes for producing one product. Such a wide range of possible changes in production processes must be explicitly considered in order to determine an optimal production configuration.

Levels of abstraction

Prasad (1998) studies the product and process complexity associated with providing variety and highlights the importance of determining the right amount of decomposition. This leads to a granularity paradox, indicating tradeoffs associated with process details and product hierarchies. Production configuration must support refinement of decision making through decomposition at different levels of abstraction.

Constraint satisfaction

In general, production configuration involves four types of constraints. Inclusion conditions specify the circumstances under which a process concept and an operation are to be included in a configuration. This type of constraints coincides with the fact that not all of the processes and operations

in a process platform are necessary for producing each individual product. The sequence or precedence relations among operations/processes constitute another type of constraints. Each sequence constraint is specified as a binary relationship between two processes or operations in the form of a predecessor and a successor. The third type of constraints is commonly used to determine execution rules regarding operation details, for example, to specify machines and tools to be used for a particular product. Finally, coordination constraints must be addressed in order to maintain the consistency of product and process variety. More specifically, these constraints control the granularity of variety derivation from certain product hierarchies to proper process elements.

Production configuration modeling support

Stobaugh and Telesio (1983) represent early efforts to develop configuration models for manufacturing. They purport to list manufacturing tasks down to a few “generic tasks” and then address major decisions through “choices within each type”. Hill (1994) describes five generic process types across 25 dimensions with an intention to capture product and market requirements, manufacturing characteristics, investment and cost issues, and infrastructure choices. Hayes and Wheelwright (1984) develop a product–process matrix to address internal fit at the process choice level. Bozarth and McDermott (1998) examine the configuration research in the broad area of manufacturing strategies and conclude that configuration models are well suited to studying complex multivariate organizational phenomena. Siddique (2005) develops a method to explicitly consider existing assembly plant configuration and resources during selection of assembly processes for new product family members. Benjaafar et al. (2002) discuss the flexible, modular and reconfigure layout configurations for meeting the needs of multiproduct enterprises. Jiao et al. (2004) study the modeling of process variety using object-oriented Petri-Nets with changeable structures.

Existing research on production configuration mostly emphasizes on analysis of basic rationale or discussion of isolated empirical studies. There is limited attempt to explore the modeling issues that support production configuration. To solve configuration problems, many approaches such as logic-based approaches, resource-based approaches, constraint-based approaches, and case-based reasoning approaches have been reported (Sabin and Weigel 1998). Most modeling work deals with the synthesis issue, that is, how a valid configuration is assembled from instances of a fixed set of predefined component types (Mittal and Frayman 1989). While most reported work focuses on individual products only, production configuration involves families of both products and processes. In addition to the representation of variety, understanding the underlying logic of product and

process families and modeling the variant derivation process are of primary importance.

Strategy for solution

Owing to their executability, graphical representation and mathematic support, Petri nets (PNs) have been well recognized as a powerful modeling, simulation and evaluation tool for complex flows and processes (Peterson 1981). Among many extensions, the colored PNs (CPNs) and the object-oriented PNs (OPNs) are able to provide a concise, flexible and manageable representation of large manufacturing systems (Jensen 1992), and to enhance the maintainability and reusability of objects in modeling (Wang 1996). CPNs and OPNs are further extended to consider changeable structures (Jiang et al. 1999).

Aiming at modeling support, this paper applies PN techniques to deal with the unique requirements posed by fundamental issues of production configuration. More specifically, a formalism of nested colored object-oriented PNs with changeable structures (NCOPNs-CSs) is developed for production configuration modeling. The CPNs, OPNs and changeable PNs are employed to define a set of nets that form the modeling formalism. Configuration constraints are handled by attaching various data regarding product items, process elements and manufacturing resources to colored tokens. In conjunction with OPNs, colored tokens are also used to deal with product and process variety. A change handling mechanism is incorporated into the modeling formalism to cope with process variations. A multilevel modeling system is further developed based on net nesting to tackle explicitly the granularity issue.

Related work

To capture the hierarchical nature of assembly planning, Thomas et al. (1996) propose a hierarchical PN framework, where each control plan corresponding to an assembly robot operation is viewed as a lower level representation of assembly tasks at the higher level. Adamous et al. (1998) introduce a hierarchical model of flexible assembly system control using OPNs, which is built upon hierarchical decomposition of the system via coordination among lower level decomposed system elements. To accommodate qualitative modeling of multiple mobile robot systems, Lopez-Mellado and Almeyda-Canepa (2003) develop a three level scheme of PNs, in which tokens themselves are defined as PNs as well.

For large and complex systems, Morandin et al. (2000) combine PN techniques with a modular strategy by considering shared resources and alternative process plans. This modular PN strategy is further extended to the concept of virtual PNs (Morandin and Kato 2003). Sets of virtual places

and virtual transitions constitute a basis for connecting separated PN modules.

Li and Lara-Rosano (1999) develop a hybrid object-oriented colored PNs (OOCPNs) by enhancing OOCPNs with time delay and firing speed. The hybrid OOCPNs are used to model batch-based electronic component manufacturing systems. To facilitate the automatic generation of PN models for manufacturing system control and scheduling, He et al. (2000) propose a decomposition methodology according to the nesting mechanism inherent in IDEF3 modeling. An IDEF3 model representing a manufacturing process is first decomposed into sub IDEF3 models based on similarity of resources. Then the decomposed IDEF3 models are transformed into respective PN control models. Finally the sub PN models are aggregated to form a complete system model.

With focus on the multiple alternative sequences of disassembling a product at the end of its useful life, Kumar Singh et al. (2003) present an expert enhanced high-level colored disassembly Petri net (EEHLCDPN) to model optimal operations and paths of disassembly. In the EEHLCDPN, control places together with other relevant net elements are introduced to deal with the selection of disassembly operations. Kumar et al. (2004) develop a formalism of extended neuro fuzzy Petri nets (ENFPNs) to solve the machine-loading problems in flexible manufacturing systems. Such major concerns in machine-loading problems as determination of optimal job sequences, job reallocation and operation-machine allocation are judged based on multiple alternatives, for which fuzzy AND/OR rules are introduced on the basis of neuron networks in ENFPNs. Similarly, Kumar et al. (2005) discuss expert enhanced colored fuzzy Petri nets to model the scheduling problems in reconfigurable manufacturing systems with focus on information delays. The selection of optimal job flows is accomplished by incorporating control places, inhibitor arcs and fuzzy truth values associated with transitions.

Problem description

This study assumes that the production process of an end product consists of a number of sequenced processes, each of which is to produce either a component assembly or a component part in the product hierarchy. Such a process, in turn, is formed by several ordered operations. The relevant definitions are elaborated below.

Definition 1 A process platform, Ω , consisting of a set of production process variants, $\{R_r\}_N$, for producing the set of product variants in a family, is defined as a tuple, $\Omega = \langle \Psi, \rightarrow \rangle$, where $\Psi = \{P_i\}_n$ is a set of process classes, each of which is for producing a family of product items, be it a

part type or an assembly type; and \rightarrow is the sequence relation between two process classes in Ψ .

Definition 2 The sequence relation \rightarrow defines a binary relationship between two process classes, P_i and P_j , $\forall i \neq j \in [1, n]$, such that $P_i \rightarrow P_j$, indicating that P_i should be completed before commencement of P_j . Furthermore, a transitive closure of \rightarrow is reflexive, so that with Ψ the set of \rightarrow forms a tree.

Definition 3 With respect to various product item families, Ψ can be further classified into two sets, i.e., $\Psi = \Psi^M \cup \Psi^S$, where $\Psi^M = \{P_i^M\}_a$ is a set of master process classes that are compulsory to all process variants; $\Psi^S = \{P_i^S\}_b$ is a set of selective process classes that are optional to process variants; and $a + b = n$.

If $P_i \in \Psi^M$, P_i is said to produce a family of assemblies or parts that are common to all variants in the product family. If $P_i \in \Psi^S$, it means that assemblies or parts produced by any instance of P_i are optional to the set of product variants.

Definition 4 Each P_i is defined as a tuple, $P_i \langle \omega_i, \succ \rangle$, where $\omega_i = \{O_{ij}\}_m$ is a set of operations classes; and \succ is the precedence relation between two operations classes in ω_i .

Definition 5 The precedence relation \succ defines a binary relationship between two operations classes, O_{is} and O_{it} , $\forall s \neq t$, such that $O_{is} \succ O_{it}$, meaning that O_{is} should be performed before O_{it} . A transitive closure of \succ is reflexive, so that with ω_i the set of \succ forms a tree representing P_i .

Definition 6 Two sets are distinguished within ω_i of P_i , i.e., $\omega_i = \omega_i^M \cup \omega_i^S$, where $\omega_i^M = \{O_{ij}^M\}_x$ is a set of master operations classes necessary to all $\{R_r\}_N$; $\omega_i^S = \{O_{ij}^S\}_y$ is a set of selective operations classes optional to $\{R_r\}_N$; and $x + y = m$. For each O_{ij} of P_i , if $O_{ij} \in \omega_i^M$, then $P_i \in \Psi^M$. Likewise, if $P_i \in \Psi^S$, then either $O_{ij} \in \omega_i^M$ or $O_{ij} \in \omega_i^S$.

Definition 7 A production process variant, R_r , consists of a series of sequenced processes, i.e., $R_r \langle \Psi_r^*, \rightarrow \rangle$, where $\Psi_r^* = \{P_{ri}^*\}_n$ is a set of specific processes, each of which is to produce a particular product item, be it a part or an assembly; and \rightarrow is the sequence relation between two processes, P_{ra}^* and P_{rb}^* , $\forall a \neq b$, such that $P_{ra}^* \rightarrow P_{rb}^*$, suggesting that process P_{ra}^* for producing item Ia must be completed before producing item Ib through P_{rb}^* .

Definition 8 Each P_{ri}^* of production process variant R_r contains an operations set, i.e., $P_{ri}^* = \langle \omega_{ri}^*, \succ \rangle$, where $\omega_{ri}^* = \{O_{rij}^*\}$ is a set of specific operations; and \succ is the precedence relation between operations in $\{O_{rij}^*\}_{N \times n \times m}$, such that $O_{ris}^* \succ O_{rit}^*$, $\forall s \neq t$, denoting that O_{ris}^* must be performed before O_{rit}^* .

A process class P_i belongs to one of these types: (1) A type of manufacturing processes consisting of a series of machining operations and non-machining operations for manufacturing a part family, e.g., material transfer; (2) A type of assembly processes involving a series of assembly operations and non-assembly operations for producing an assembly family; and (3) A mixed type of processes involving machining operations, assembly operations, and/or non-machining/assembly operations for forming an assembly family. If P_{ri}^* is to produce a part, then its corresponding $\{O_{rij}^*\}_{N \times n \times m}$ comprises all machining operations. If P_{ri}^* is to form an assembly, then O_{rij}^* may be a machining operation or an assembly operation.

Both manufacturing and assembly processes employ a number of material handling devices, a number of buffers, and a set of machine classes. Each machine class, in turn, exhibits a number of similar machines. These machines are either necessary or optional for producing a product family. Material handlers such as AGVs, robots, and human operators may be used to transfer materials, semi-finished items (i.e., WIP), and finished items from one location to another. A number of buffers, including input buffers, WIP buffers and output buffers, are also used to store materials, WIP and finished items or products, respectively. Figure 1 summarizes the process elements and relationships involved in production configuration.

Net definitions in NCOPNs-CSs

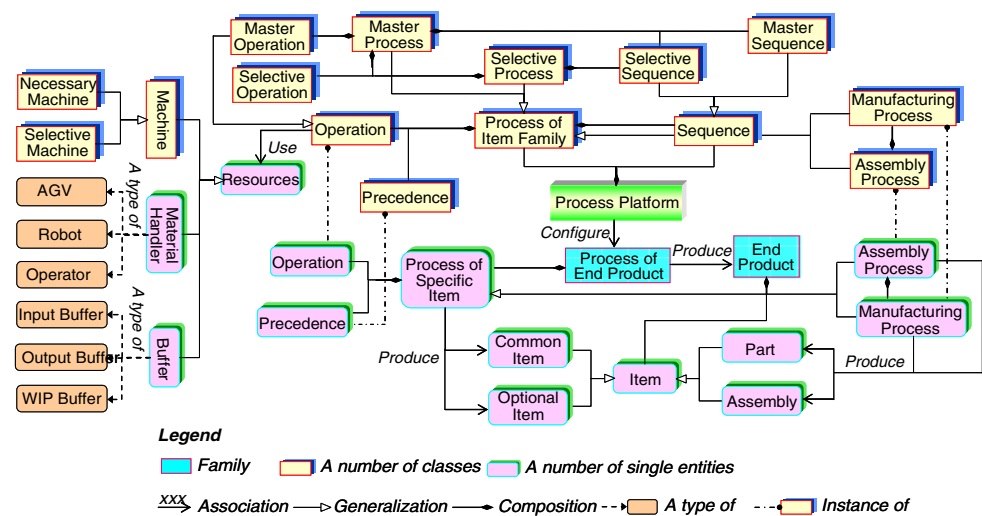
Resource net

A production system comprises a set of input and output buffers, WIP buffers, machining and assembly machines, and material handlers. Each of them performs certain functionality that is achieved through the internal activities, and thus can be modeled as OPNs from an object-oriented perspective.

Definition 9 A resource net is defined as a tuple, $RNet = (G^{OPN}, \omega)$, where $RNet$ is a resource net, be it an input buffer, output buffer, WIP buffer, machining machine, assembly machine, or material handler. G^{OPN} is a structure of OPN, such that,

$$G^{OPN} = (SP, IM, OM, AT, C^{SP}, C^{IM}, C^{OM}, C^{AT}, I^{id}, O^{id}).$$

where $SP = \{sp_i\}_{N^{SP}}$ is a finite set of state places; $IM = \{im_k\}_{N^{IM}}$ is a finite set of input message places, $SP \cap IM = \phi$, $sp \cup IM \neq \phi$; $OM = \{om_l\}_{N^{OM}}$ is a finite set of output message places, $IM \cap OM = \phi$, $SP \cap OM = \phi$, $SP \cap IM \cap OM = \phi$; and $AT = \{at_j\}_{N^{AT}}$ is a finite set of activity transitions.

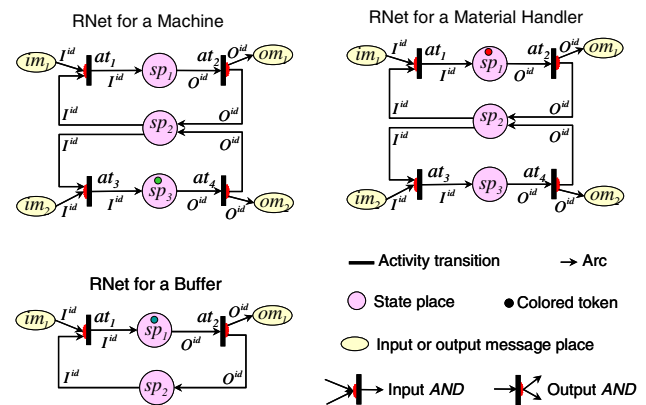
Fig. 1 Process elements and relationships in production configuration

$C^{SP}(i) = \{c_j^{sp_i}\}_{N^{sp_i}}$ is a set of colored tokens associated with a state place, sp_i . $C^{IM}(k) = \{c_j^{im_k}\}_{N^{im_k}}$ is a set of colored tokens associated with an input message place, im_k . $C^{OM}(l) = \{c_j^{om_l}\}_{N^{om_l}}$ is a set of colored tokens associated with an output message place, om_l . $C^{AT}(j) = \{c_i^{at_j}\}$ is a set of colored tokens associated with an activity transition, at_j . Moreover, $C^{SP}(i)$, $C^{IM}(k)$, $C^{OM}(l)$, and $C^{AT}(j)$ are determined by the items that resource objects can process.

$I^{id}(SP, AT/c^{at_j}) : C^{SP} \times C^{AT} \rightarrow C^{SP} \cup \phi$ or $I^{id}(IM, AT/c^{at_j}) : C^{IM} \times C^{AT} \rightarrow C^{IM} \cup \phi$ is an input identity function for arcs that connect state places SP or input message places IM to activity transitions AT with a firing colored token, $c^{at_j} \in C^{AT}(j)$, $\exists j \in [1, N^{AT}]$. $O^{id}(SP, AT/c^{at_j}) : C^{SP} \times C^{AT} \rightarrow C^{SP} \cup \phi$ or $O^{id}(OM, AT/c^{at_j}) : C^{OM} \times C^{AT} \rightarrow C^{OM} \cup \phi$ is an output identity function for arcs that connect activity transitions AT with a firing colored token, c^{at_j} , to state places SP or output message places OM .

$\omega : (SP \rightarrow M^{C^{SP}}, IM \rightarrow M^{C^{IM}}, OM \rightarrow M^{C^{OM}})$ is a marking function, such that $\omega(SP) = M^{C^{SP}}$, $\omega(IM) = M^{C^{IM}}$, $\omega(OM) = M^{C^{OM}}$, where $M^{C^{SP}}$, $M^{C^{IM}}$ and $M^{C^{OM}}$ are the families of all multisets over C^{SP} , C^{IM} and C^{OM} , respectively.

The set of state places, $\{sp_i\}_{N^{sp}}$, specify all possible states that a resource object may possess. As shown in Fig. 2 and Table 1, a machine has 3 states, namely idle, setting up and processing. A material handler possesses 3 states as well, i.e., idle, transferring parts from buffers to machines, and transferring parts from machines to buffers. All types of buffers assume two states: idle vs. capacity available for loading parts. The AND relations among input arcs indicate that tokens in the input places must appear at the same time to

**Fig. 2** States and activities of resource objects

enable activity transitions with certain colored tokens. After transition firing, tokens should be simultaneously added into places with AND relation arcs.

To enable communication among objects, input message places IM and output message places OM are defined as the interface for objects to send and receive messages (i.e., tokens). For example, in Fig. 2, im_i of the $RNet$ for a material handler is used for receiving those messages sent by machines with request for loading parts. om_j holds the messages to be sent to other machines about the completion of part transferring. AT describes all activities that a resource object can perform. Input identity functions $I^{id}(SP, AT/c^{at_j})$ and $I^{id}(IM, AT/c^{at_j})$ specify the set of pre-conditions for enabling AT with a colored token, c^{at_j} , whilst output identity functions $O^{id}(SP, AT/c^{at_j})$ and $O^{id}(OM, AT/c^{at_j})$ regulate the set of post-conditions of the firing of AT . ω represents the marking of the resource object after each transition firing.

Table 1 States and activities of resource objects

Place	Material handler (MH) RNet	Machine RNet	Buffer RNet
im_1	A request (from machine) for transferring part in	A request (from buffer) for setting up	A message (from MH) of parts ready to be moved in
im_2	A request (from machine) for transferring part out	A request (from MH) for loading parts	
om_1	A message of part transferring completion	A message of completing setup	A message of parts ready to be moved out
om_2	A message of part transferring completion	A message of completing part unloading	
sp_1	Transferring parts into machine	Setting up	Part loaded in buffer
sp_2	Idle	Idle	Idle and capacity available
sp_3	Transferring parts out of machine	Processing parts	
at_1	Start transferring parts into machine	Start setting up	Loading part into buffer
at_2	End transferring parts	End setting up	Unloading part from buffer
at_3	Start transferring parts out of machine	Start loading parts	
at_4	End transferring parts out	End part unloading	

The dynamic behavior of a resource object is characterized by the set of state places, $\{Sp_i\}_{N^{SP}}$, and the set of activity transitions, $\{at_j\}_{N^{AT}}$. Only when the connected input state place Sp_i and message place im_k hold the colored tokens as specified by the input identity functions, i.e., $\omega(sp_i) \geq I^{id}(sp_i, at_j/c^{at_j})$ and $\omega(im_k) \geq I^{id}(im_k, at_j/c^{at_j})$, can an activity of at_j be carried out, that is, activity transition at_j is activated. The firing of at_j results in the removal of tokens from the set of input places as specified by the input identity functions and the addition of colored tokens to the set of connected output places as specified in the output identify functions.

Manufacturing net

Definition 10 A manufacturing net is defined as a tuple, $MNet = (G^{OPN_{s-cs}}, \mu)$, where $MNet$ is a manufacturing net representing the processes of manufacturing a part family. $G^{OPN_{s-cs}}$ is a system of OPNs with changeable structures, such that,

$$G^{OPN_{s-cs}} = (P, T, C^P, C^T, I, O),$$

where $P = \{P_i\}_{N^P}$ is a finite set of places, representing the set of resource objects involved in the manufacturing process; $T = \{t_j\}_{N^T}$ is a finite set of transitions, $P \cup T \neq \emptyset$, $P \cap T = \emptyset$, $T = T_1 \cup T_2$, where T_1 is a set of input OR relation transition (i.e., transitions with input arcs bearing OR relations), and T_2 is a set of ordinary transitions; $C^P(i) = \{c_j^{p_i}\}_{N^{P_i}}$ is a finite set of colored tokens associated with a place, P_i ; and $C^T(j) = \{c_j^{t_j}\}_{N^{t_j}}$ is a finite set of colored tokens associated with a transition, t_j .

$I(p_i, t_j/c^{t_j}) : C^P(i) \times C^T(j) \rightarrow C^P(i) \cup \phi$ is an input transform function for an arc, (p_j, t_j) , $\forall p_i \in {}^\circ t_j$, that connects a place to a transition with a firing colored token, c^{t_j} .

Likewise, $O(p_i, t_j/c^{t_j}) : C^P(i) \times C^T(j) \rightarrow C^P(i) \cup \phi$ is an output transform function for an arc, (p_j, t_j) , $\forall p_i \in {}^\circ t_j$, that connects a transition with a firing colored token, c^{t_j} , to a place. Moreover, $\mu : P \rightarrow M^{C^P}$ is a marking function, $\mu(P) = M^{C^P}$, where M^{C^P} is the family of all multisets over C^P .

To accommodate process variations, e.g., adding or removing resources, or changing the execution order of two machining or non-machining operations, the OPNs with changeable structures (OPNs-cs) in Jiang et al. (1999) is adopted to define the net structure $G^{OPN_{s-cs}}$ of an $MNet$. The set of places, $\{p_i\}_{N^P}$, are defined for the set of resource objects. Each place p_i carries out either a machining operation or a non-machining operation. Therefore, the set of colored tokens, $\{c_j^{p_i}\}_{N^{P_i}}$, associated with each place p_i are used to represent the operations, the employed resource objects, as well as the corresponding product items.

Furthermore, the set of transitions $\{t_j\}_{N^T}$ specify the starting or ending of the relevant operations. The colored token of each transition t_j is defined exactly the same as that of its input place. The input and output transform functions specify the type and number of colored tokens to be removed/added from/to the relevant input and output places. In other words, each transition t_j , colored token set $\{c_i^{t_j}\}_{N^{t_j}}$, and the associated input and output transform functions determine the message passing relationship between two places, p_a and p_b , in accordance with the flow of a part's manufacturing process. If a transition, t_j , is to be activated with respect to a firing colored token, c^{t_j} , each of its input place p_i must contain colored tokens, the type and number of which should be greater or equal to the one specified by the input transform function, i.e., $\mu(p_i) \geq I(p_i, t_j/c^{t_j})$, $p_i \in {}^\circ t_j$.

Assembly net

Definition 11 An assembly net is defined as a tuple, $ANet = (G^{OPN_{s-cs}}, \mu)$, where $ANet$ is an assembly net representing the assembly processes of an assembly family. $G^{OPN_{s-cs}}$ is a system of OPNs with changeable structures, $G^{OPN_{s-cs}} = (P, T, C^P, C^T, I, O)$, where T , C^T , I , O , and μ carry the same meaning as that for an $MNet$.

P is a finite set of places, $\{p_i\}_{N^P}$, $P = P^1 \cup P^2$, where $P^1 = \{p_i^1\}_{N^{P^1}}$ is a set of places corresponding to $MNets$ and/or lower level $ANets$ of child assemblies; $P^2 = \{p_i^2\}_{N^{P^2}}$ is a set of places representing the resource objects and the performed subassembly processes; and $P^1 \cap P^2 = \emptyset$, $N^{P^1} + N^{P^2} = N^P$.

C^P is a finite set of colored tokens associated with all places, $C^P = C^{P^1} \cup C^{P^2}$, where each $C^{P^1}(i) = \{c_j^{p_i^1}\}_{N^{IF_{p_i^1}}}$ is a set of colored tokens associated with a place, p_i^1 , indicating the set of part or assembly variants of a family produced by the nested $MNet$ or $ANet$ at the lower level; and $C^{P^2}(i) = \{c_j^{p_i^2}\}_{N^{P_j^2}}$ is a set of colored tokens associated with a resource object p_i^2 .

An $ANet$ is defined to represent the processes of producing a family of assemblies. Changes to component items that form assembly variants cause variations in the corresponding assembly processes. To deal with such process changes, the structural change handling mechanism is introduced to the $ANet$. Unlike those places in an $MNet$, the places in an $ANet$ may not be always related to the resource objects. They are also defined for the child parts and child assemblies produced by the $MNets$ or $ANets$. If a place, p_i , is defined for $MNets$ or $ANet$ at a lower level, the set of colored tokens, $\{c_j^{p_i^l}\}_{N^{IF_{p_i^l}}}$, are specified to indicate the set of similar item variants (either parts or assemblies), which are produced by the processes nested in the places. If it represents a resource object, the set of colored tokens are assigned according to the process flow of the assembly and used to indicate the assembly operations or non-assembly operations along with the associated resource objects.

Process net

Definition 12 A process net is defined as a tuple, $PNet = (G^{OPN_{s-cs}}, \mu)$, where $PNet$ is a process net representing the conceptual production processes of producing a family of end products. $G^{OPN_{s-cs}}$ is a system of OPNs with changeable structures, such that $G^{OPN_{s-cs}} = (P, T, C^P, C^T, I, O)$, where T , C^T , I , O , and μ carry the same meaning as that for $MNet$.

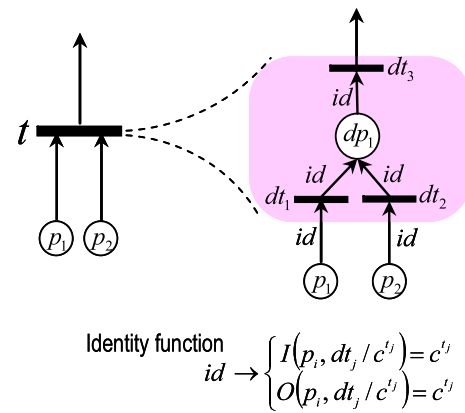


Fig. 3 Transition decomposition

$P = \{p_i\}_{N^P}$ is a finite set of places, $P = P^M \cup P^A \cup P^R$, where $P^M = \{p_i^M\}_{N^{P^M}}$ is a set of places corresponding to $MNets$; $P^A = \{p_i^A\}_{N^{P^A}}$ is a set of places representing $ANets$; $P^R = \{p_i^R\}_{N^{P^R}}$ is a set of places indicating resource objects and operations performed; and $P^M \cap P^A \cap P^R = \emptyset$, $N^{P^M} + N^{P^A} + N^{P^R} = N^P$.

C^P is a finite set of colored tokens associated with all places, $C^P = C^{P^M} \cup C^{P^A} \cup C^{P^R}$, where each $C^{P^M}(i) = \{c_j^{p_i^M}\}_{N^{PF_{p_i^M}}}$ is a set of colored tokens associated with a place, p_i^M , and indicates the set of similar part variants produced by the represented $MNet$; $C^{P^A}(i) = \{c_j^{p_i^A}\}_{N^{AF_{p_i^A}}}$ is a set of colored tokens associated with a place, p_i^A , and indicates the set of assembly variants produced by the represented $ANet$; and $C^{P^R}(i) = \{c_j^{p_i^R}\}_{N^{P_i^R}}$ is the set of colored tokens associated with the set of resource objects along with the performed operations.

A $PNet$ is defined for the abstract processes of producing a family of end products. Similar to an $ANet$, the structural change handling mechanism is employed in $PNet$ to handle process variations. The places in a $PNet$ are specified to represent either resource objects carrying out certain operations, or $MNets$ manufacturing parts, or $ANets$ producing assemblies. If a place, p_i , is defined for an $MNet$ or $ANet$, the set of colored tokens, $\{c_j^{p_i^M}\}_{N^{PF_{p_i^M}}}$ or $\{c_j^{p_i^A}\}_{N^{AF_{p_i^A}}}$, are specified to indicate the set of part or assembly variants in a family. If a place, p_i , is specified to represent a resource object, the set of colored tokens are used to indicate the operations carried out by the objects.

To specify firing conditions for transitions with respect to firing colored tokens, each of these transitions, i.e., T_1 in an $MNet$, $ANet$ or $PNet$, is decomposed into several input transitions, a state place and an output transition. As shown in Fig. 3, the decomposed input and output transform

functions as well as the state place are assigned with the same colored tokens as those associated with the input places of the original transition.

For a single resource object, whose number is one, there may exist more than one input arc. Therefore, conflict may occur when multiple objects or subprocesses require a single object to perform multiple operations at the same time. To maintain a 1-bounded property and the safeness of an object place, inhibitor arcs, $Inh(p_i, t_j), \forall t_j \in {}^\circ p_i$, are introduced to these objects (Wang and Wu 1998). The inhibitor arcs of a resource object are drawn from the set of input transitions to the single object. Different from those general input arcs, inhibitor arcs are indicated as dotted lines with circles at the transition ends. If $Inh(p_i, t_j) = 1$, it implies that no operation request can be passed to the object represented by the place unless the object is not occupied, that is, there is no token in the place.

Nested net system for production configuration

Net nesting

Definition 13 A multilevel nested net system is defined as a tuple, $NNSys = (PNet, R, M, A, PP, SP, \varphi)$, where $NNSys$ is the multilevel nested net system for modeling production configuration; $PNet$ is the process net describing conceptual production processes of a product family; and $\varphi(pp_i) : PP \rightarrow SP$ is a port place assignment function. It is defined from PP to SP , establishing a binary relationship between them.

$R = \{r_i\}_{NR}$ is a finite set of $RNets$, $R = R^1 \cup R^2 \dots \cup R^O$, where each $R^O = \{r_i^O\}_{NR^O}, \forall O \in [1, O]$, is a finite set of $RNets$ representing the behaviors of the objects that are in the same $MNets$ and $ANets$, or the $PNet$. $M = \{m_i\}_{NM}$ is a finite set of $MNets$, $M = M^1 \cup M^2 \dots \cup M^P$, where each $M^P = \{m_i^P\}_{NM^P}, \forall P \in [1, P]$, is a finite set of $MNets$ nested in such places that are in the same nets at the immediate higher levels. Likewise, $A = \{a_i\}_{NA}$ is a finite set of $ANets$, $A = A^1 \cup A^2 \dots \cup A^Q$, where each $A^q = \{a_i^q\}_{NA^q}, \forall q \in [1, Q]$ represents the set of $ANets$ that are nested in the same nets at the immediate higher levels.

$PP = \{pp_i\}_{NPP}$ is a finite set of port places representing resource objects in A and M . The messages in the output message places of each pp_i will be sent to the input message places of such resource objects that are connected with those places representing $MNets$ or $ANets$, of which these port places belong to. In other words, the places representing message receiving objects and the places that nest $MNets$ or $ANets$ of port places belong to the same nets.

$SP = \{sp_i\}_{N^SP}$ is a finite set of socket places indicating resource objects in A and the $PNet$. The input message places

of each sp_j will receive messages sent from resource objects of the associated pp_i that belong to the lower level $MNets$ or $ANets$, which are represented by the places connected to sp_i in the same $ANets$ or the $PNet$.

Performing as an abstraction mechanism, $NNSys$ facilitates the selection of conceptual processes with right amount of details. Within $NNSys$, the highest level is the $PNet$, while a number of $RNets$, $MNets$ and $ANets$ are located at the second level. Each of these nets provides more details for the respective places in the $PNet$. The nets at any lower level provide detailed descriptions of the assembly and manufacturing processes nested in the places of the nets at its immediate higher level. At the lowest level of each path, all nets become $RNets$, whilst a mixture of $RNets$, $MNets$ and $ANets$ can be found at any arbitrary level. Figure 4 demonstrates an $N + 2$ level net system with nested $MNets$ and $ANets$, as well as encapsulated $RNets$. Table 2 summarizes descriptions of places and transitions of different nets at each level.

Message exchange among objects between two nets at adjacent levels is implemented through the port places at the lower level nested nets and the socket places at the higher-level host nets. Both port and socket places are only defined for resource objects. Figure 5 illustrates such a mechanism. When a token representing a part is produced in the $MNet$ at level $i + 1$, which is nested in place p_2 in the $ANet$ at level i , it is loaded into the output buffer represented by place p_6 . Then a token with the same color appears in place p_2 in the $ANet$ at level i . Meanwhile, a message requesting machine setup from the output buffer p_6 in the nested $MNet$ is sent to the place p_3 , representing an assembly machine, in the $ANet$ at level i .

Support to production configuration

Consistent with the hierarchical structure of a product, production configuration can be regarded as a recursive process of configuring process elements through various levels of abstraction. For a product variant with an N level hierarchy, production configuration is carried out for various product items at each level. Figure 6 draws an analogy of decision making among the product hierarchy, production configuration and the nested net system model. For a product with an N level hierarchy, the iterative process refinement will form a process hierarchy with $N + 1$ levels, wherein the processes at the $(N + 1)$ -th level are the detailed manufacturing processes for parts at the N -th level of the product hierarchy. Since parts can be at each level of the product hierarchy, detailed manufacturing processes can thus be found at each level of the $N + 1$ process hierarchy except the first level. In relation to the detailed manufacturing processes at the $(N + 1)$ -th level of the process hierarchy, $MNets$ are constructed at level $N + 1$ of the nested net system model. $RNets$ representing internal

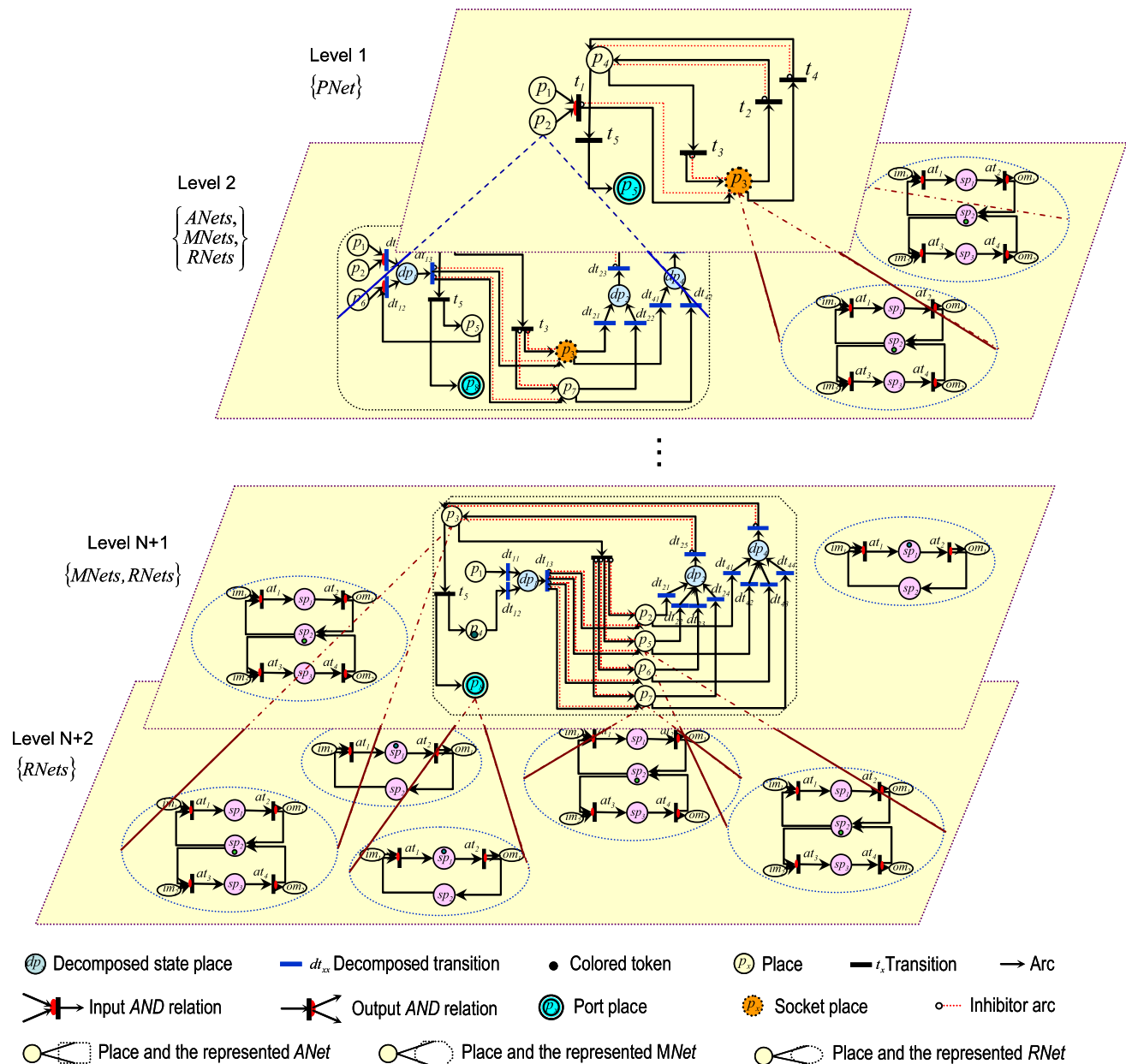


Fig. 4 An example of nested net system

behaviors of resource objects in the *MNets* at level $N + 1$ are established at level $N + 2$ of the production configuration model. Therefore, the production configuration model for a product with an N level product hierarchy will assume $N + 2$ levels.

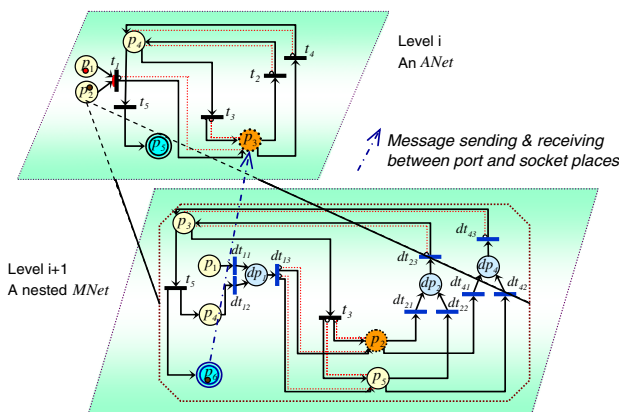
Case study

The NCOPNs-CSs formalism is tested using an industrial case of vibration motor production. Figure 7(a) shows the

major product items of motors and Fig. 7(b) shows the hierarchies of two motor variants, VM_1 and VM_2 . Descriptions of the product items are given in Table 3. The motor process platform has been constructed a priori, including sets of master and selective processes and various process elements regarding operations, precedence, manufacturing resources, and product items. In addition, execution rules for operation details, inclusion conditions for selective process elements and other constraints regulating execution orders of processes and operations have been identified by domain experts.

Table 2 An example of nested net system

Place	Level 1	Level 2	Level N+1
p_1	Part a_1 & its manufacturing process	Part $a_1 p_1$ of A_1 & its manufacturing process	Input buffer in a part's $MNet$
p_2	Assembly A_1 & its assembly process	Part $a_1 p_2$ of A_1 & its manufacturing process	A machining machine & its machining operation
p_3	An assembly machine & its operation	An assembly machine & its operation in A_1 's $ANet$	A material handler in a part's $MNet$
p_4	A material handler & its operation	A material handler & its operation in A_1 's $ANet$	WIP buffer in a part's $MNet$
p_5	Output buffer in the final product's $PNet$	WIP buffer in A_1 's $ANet$	A machining machine & its machining operation
p_6		Part $a_1 p_3$ of A_1 & its manufacturing process	A machining machine & its machining operation
p_7		An assembly machine & its operation in A_1 's $ANet$	A machining machine & its machining operation
p_8		Output buffer in A_1 's $ANet$	Output buffer of a part's $MNet$
dp_1		Decomposed state place of transition t_1 in A_1 's $ANet$	Decomposed state place of transition t_1 in the $MNet$
dp_2		Decomposed state place of transition t_2 in A_1 's $ANet$	Decomposed state place of transition t_2 in the $MNet$
dp_4		Decomposed state place of transition t_4 in A_1 's $ANet$	Decomposed state place of transition t_4 in the $MNet$
$dt_{11/12/13}$		Decomposed transitions of t_1 in A_1 's $ANet$	Decomposed transitions of t_1 in the $MNet$
$dt_{21/22/23/24/25}$		Decomposed transitions of t_2 in A_1 's $ANet$	Decomposed transitions of t_2 in the $MNet$
t_3	A transition in the $PNet$	A transition in A_1 's $ANet$	A transition in a part's $MNet$
$dt_{41/42/43/44/45}$		Decomposed transitions of t_4 in A_1 's $ANet$	Decomposed transitions of t_4 in the $MNet$
t_5	A transition in the $PNet$	A transition in A_1 's $ANet$	A transition in a part's $MNet$

**Fig. 5** Communication between nets at two adjacent levels through port and socket places

Specification of $PNet$

In accordance with the specifications of three child items at the first level of the product hierarchies, two conceptual production processes are determined for VM_1 and VM_2 , respectively. A conceptual production process reflects the assembly

flow of a number of items (parts and/or assemblies) rather than the detailed processes of manufacturing a part. Both the two processes require machine ($W_c M_1$), whereas the production of VM_1 needs another assembly machine $Rh I_1$. The process flows for VM_1 and VM_2 are detailed in Table 4, together with the required manufacturing resources.

The $PNet$ is then constructed according to the process flows of the configured conceptual processes, as shown in Fig. 8. The places representing operations and resource objects and their relevant colored tokens are defined, as shown in Table 4. In Fig. 8, dt_{xx} and dp_{xx} denote the decomposed transitions and state places of three transitions, T_1 , T_2 and T_3 , whose input arcs bear *OR* relations. In addition, the inhibitor arcs have been applied to three single resources, including a material handler (p_4) and two assembly machines, $W_c M_1(p_3)$ and $Rh I_1(p_7)$. The colored token in place p_8 (i.e., the output buffer) indicates that one VM_1 or VM_2 has been produced. To deal with process changes, e.g., removal $Rh I_1$ of VM_1 from production process, the structural change handling mechanism is enacted to modify the input/output transform functions through the message sending and receiving relations in the $PNet$.

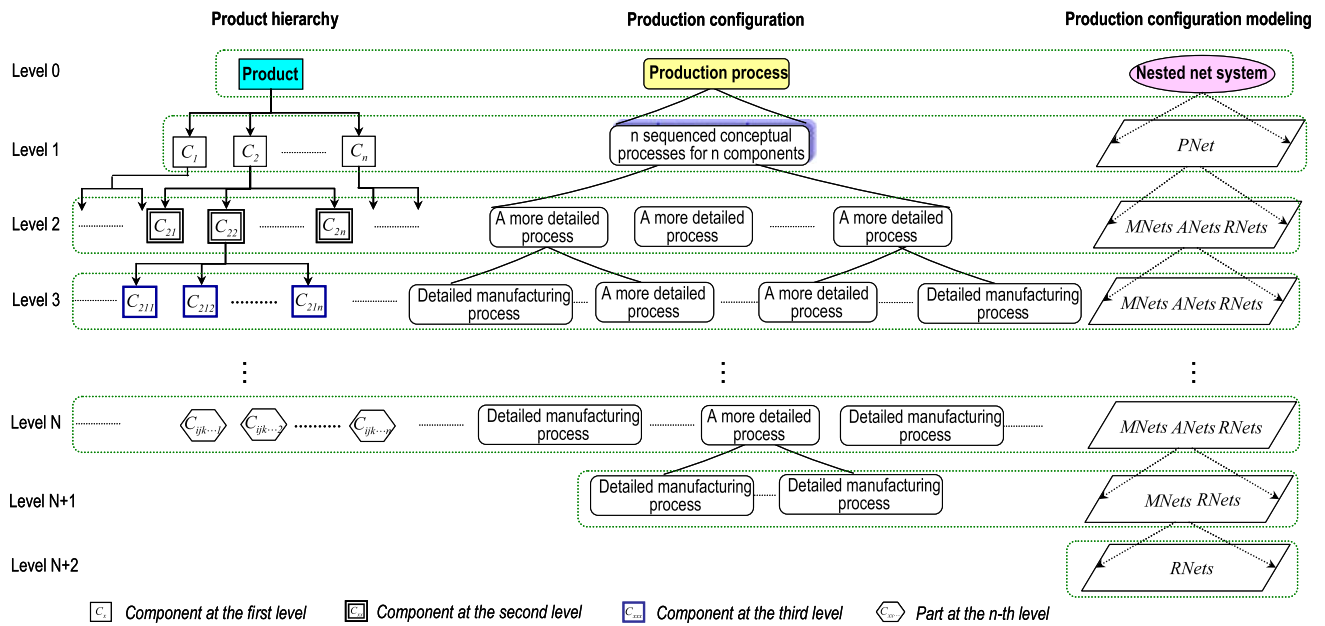


Fig. 6 Analogy of decision making for granularity

Fig. 7 Hierarchies of motor variants

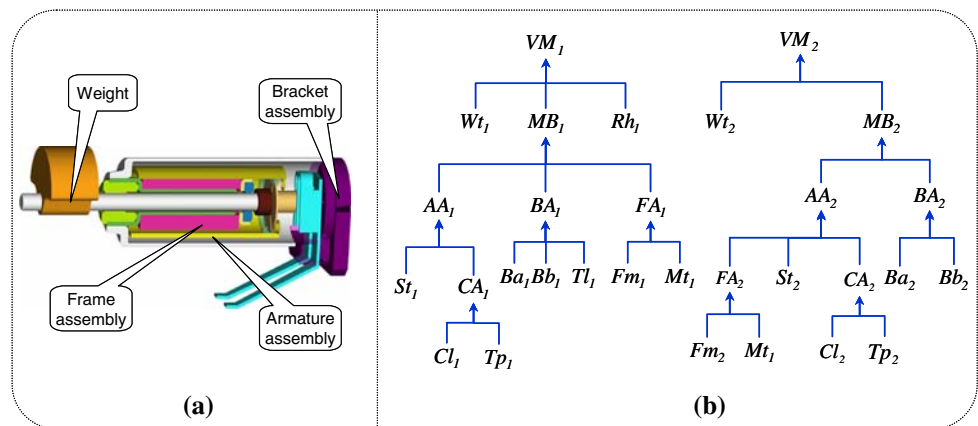


Table 3 Specific product items

Abbreviations	Items
VM1/VM2	Vibration motor variants 1 and 2
Wt1/Wt2	Weight variants 1 and 2
Rh1/Rh2	Rubber holder variants 1 and 2
MB1/MB2	Mainbody variants 1 and 2
FA1/FA2	Frame assembly variants 1 and 2
AA1/AA2	Armature assembly variants 1 and 2
BA1/BA2	Bracket assembly variants 1 and 2
CA1/CA2	Coil assembly variants 1 and 2
Fm1/Fm2	Frame variants 1 and 2
Mt1	Magnet variant 1
St1/St2	Shaft variants 1 and 2
Ba1/Ba2	Bracket a variants 1 and 2
Bb1/Bb2	Bracket b variants 1 and 2
Tl1/Tl2	Terminal variants 1 and 2
Cl1/Cl2	Coil variants 1 and 2
Tp1/Tp2	Tape variants 1 and 2

Table 4 Process flows of VM_1 and VM_2

Process flow of VM_1			Process flow of VM_2		
Operations	Resource (or process)/place	Colored token	Operations	Resource (or process)/place	Colored token
Part Wt_1 is produced	Manufacturing process/ p_1	Wt_1	Part Wt_2 is produced	Manufacturing process/ p_1	Wt_2
Assembly MB_1 is produced	Assembly process/ p_2	MB_1	Assembly MB_2 is produced	Assembly process/ p_2	MB_2
WcM_1 setting up	WcM_1/p_3	$VM_1 \cdot 0$	WcM_1 setting up	WcM_1/p_3	$VM_2 \cdot 0$
Being transferred	Material handler/ p_4	$VM_1 \cdot 1$	Being transferred	Material handler/ p_4	$VM_2 \cdot 1$
Being assembled	WcM_1/p_3	$VM_1 \cdot 2$	Being assembled	WcM_1/p_3	$VM_2 \cdot 2$
Being transferred	Material handler/ p_4	$VM_1 \cdot 3$	Being transferred	Material handler/ p_4	$VM_2 \cdot 3$
Staying in buffer	WIP buffer/ p_5	$VM_1 \cdot 4$	Staying in buffer	Output buffer/ p_8	VM_2
Part Rh_1 is produced	Manufacturing process/ p_6	Rh_1			
Rh_1 setting up	Rh_1/p_7	$VM_1 \cdot 5$			
Being transferred	Material handler/ p_4	$VM_1 \cdot 6$			
Being assembled	Rh_1/p_7	$VM_1 \cdot 7$			
Being transferred	Material handler/ p_4	$VM_1 \cdot 8$			
Staying in buffer	Output buffer/ p_8	VM_1			

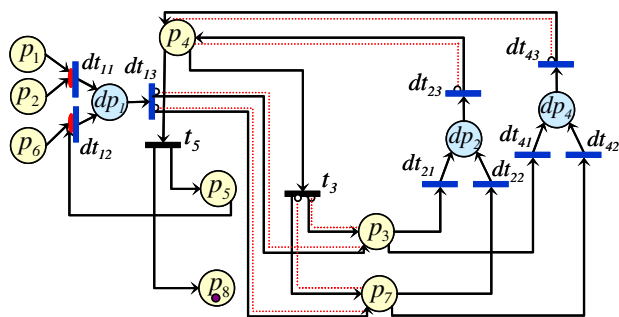
**Fig. 8** The $PNet$ for VM_1 and VM_2

Table 5 shows the detailed list of colored tokens assigned to the places and transitions (including those decomposed transitions and state places), as well as the input and output transform functions. Generalizing the two variant case to the whole motor family, the colored tokens assigned to places p_1 , p_2 and p_6 indicate all the family members that are produced by the nested processes. The tokens assigned to place p_8 (the output buffer) thus represent the entire motor family.

$MNet$ for part family

Figure 9 illustrates the $MNet$ for a part family, consisting of two variants, Wt_1 and Wt_2 . Their respective manufacturing processes are configured as shown in Table 6, where the colored tokens are specified to represent both the operations and the employed manufacturing resources. While both of them employ the same machine (WMM_1 and WMM_2), the manufacturing processes differ from one another in their flows. The places associated with the net are defined, as shown in

Table 7. Table 8 summarizes specifications of all the places, transitions, colored tokens, input and output arcs and transform functions defined for the $MNet$. The colored tokens are assigned to different places, thus differentiating specific operations related to individual part variants of the family,

$ANet$ for assembly family

As shown in Fig. 10, the $ANet$ for assembly variants MB_1 and MB_2 is constructed according to the configured conceptual process flows in Table 9. The colored tokens in relation to the respective manufacturing resources and the performed operations are shown in Table 9. The mechanism for handling process changes is adopted in the $ANet$ to describe that fact that assembly machine FCM_1 is required for producing MB_1 , but not for MB_2 . Table 10 summarizes specifications of all the places, transitions, colored tokens, input and output arcs and transform functions defined for the $ANet$. Places p_1 , p_2 and p_6 represent the respective assembly processes of three families: AAs, BAs and FAs. The tokens are specified for each of the places, denoting specific assembly variants of a family.

Specification of $RNet$

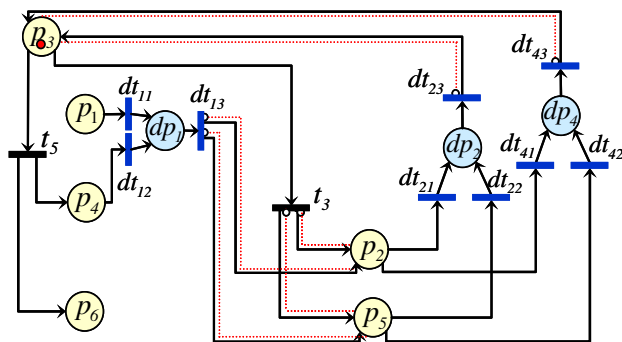
The $MNet$, $ANet$ and $PNet$ are constructed to model the manufacturing and assembly processes of parts, assemblies and final products, respectively. The internal behavior of each resource object is modeled by constructing an $RNet$. Figure 11 shows an example of the $RNet$ for the final product assembly machine WcM_1 in the $PNet$ (Fig. 8).

Table 5 Descriptions of the *PNet* in Fig. 8

Place	Colored token	Transition	Colored token	Input arc	Input arc transform function	Output arc	Output arc transform function
p_1	$\{Wt_1, Wt_2\}$	dt_{11}	$\{Wt_1MB_1, Wt_2MB_2\}$	(p_1, dt_{11})	$\{Wt_1MB_1, Wt_2MB_2\}$	(dp_1, dt_{11})	$\{Wt_1MB_1, Wt_2MB_2\}$
p_2	$\{MB_1, MB_2\}$			(p_2, dt_{11})	$\{Wt_1MB_1, Wt_2MB_2\}$		
p_3	$\left\{ \begin{array}{l} VM_1 \cdot 0, VM_1 \cdot 2 \\ VM_2 \cdot 0, VM_2 \cdot 2 \end{array} \right\}$	dt_{12}	$\{VM_1 \cdot 4Rh_1\}$	(p_5, dt_{12})	$\{VM_1 \cdot 4Rh_1\}$	(dp_1, dt_{12})	$\{VM_1 \cdot 4Rh_1\}$
p_4	$\left\{ \begin{array}{l} VM_1 \cdot 1, VM_1 \cdot 3, \\ VM_2 \cdot 1, VM_2 \cdot 3, \\ VM_1 \cdot 6, VM_1 \cdot 8 \end{array} \right\}$			(p_6, dt_{12})	$\{VM_1 \cdot 4Rh_1\}$		
		dt_{13}	$\left\{ \begin{array}{l} VM_1 \cdot 4Rh_1, \\ Wt_1MB_1, Wt_2MB_2 \end{array} \right\}$	(dp_1, dt_{13})	$\left\{ \begin{array}{l} VM_1 \cdot 4Rh_1, \\ Wt_1MB_1, \\ Wt_2MB_2 \end{array} \right\}$	(p_3, dt_{13})	$\{VM_1 \cdot 0, VM_2 \cdot 0\}$
p_5	$\{VM_1 \cdot 4\}$	dt_{21}	$\{VM_1 \cdot 0, VM_2 \cdot 0\}$	(p_3, dt_{21})	$\{VM_1 \cdot 0, VM_2 \cdot 0\}$	(dp_2, dt_{21})	$\{VM_1 \cdot 0, VM_2 \cdot 0\}$
p_6	$\{Rh_1\}$	dt_{22}	$\{VM_1 \cdot 5\}$	(p_7, dt_{22})	$\{VM_1 \cdot 5\}$	(dp_2, dt_{22})	$\{VM_1 \cdot 5\}$
p_7	$\{VM_1 \cdot 5, VM_1 \cdot 7\}$	dt_{23}	$\{VM_2 \cdot 0, VM_1 \cdot 0, VM_1 \cdot 5\}$	(dp_2, dt_{23})	$\{VM_2 \cdot 0, VM_1 \cdot 0, VM_1 \cdot 5\}$	(p_4, dt_{23})	$\{VM_2 \cdot 1, VM_1 \cdot 1, VM_1 \cdot 6\}$
p_8	$\{VM_1, VM_2\}$	t_3	$\{VM_2 \cdot 1, VM_1 \cdot 1, VM_1 \cdot 6\}$	(p_4, t_3)	$\{VM_2 \cdot 1, VM_1 \cdot 1, VM_1 \cdot 6\}$	(p_3, t_3)	$\{VM_1 \cdot 2, VM_2 \cdot 2\}$
dp_1	$\left\{ \begin{array}{l} VM_1 \cdot 4Rh_1, \\ Wt_1MB_1, Wt_2MB_2 \end{array} \right\}$	dt_{41}	$\{VM_1 \cdot 2, VM_2 \cdot 2\}$	(p_3, dt_{41})	$\{VM_1 \cdot 2, VM_2 \cdot 2\}$	(dp_4, dt_{41})	$\{VM_1 \cdot 2, VM_2 \cdot 2\}$
dp_2	$\left\{ \begin{array}{l} VM_2 \cdot 0, \\ VM_1 \cdot 0, VM_1 \cdot 5 \end{array} \right\}$	dt_{42}	$\{VM_1 \cdot 7\}$	(p_7, dt_{42})	$\{VM_1 \cdot 7\}$	(dp_4, dt_{42})	$\{VM_1 \cdot 7\}$
		dt_{43}	$\{VM_2 \cdot 2, VM_1 \cdot 2, VM_1 \cdot 7\}$	(dp_4, dt_{43})	$\{VM_2 \cdot 2, VM_1 \cdot 2, VM_1 \cdot 7\}$	(p_4, dt_{43})	$\{VM_2 \cdot 3, VM_1 \cdot 3, VM_1 \cdot 8\}$
dp_4	$\left\{ \begin{array}{l} VM_2 \cdot 2, \\ VM_1 \cdot 2, VM_1 \cdot 7 \end{array} \right\}$	t_5	$\{VM_2 \cdot 3, VM_1 \cdot 3, VM_1 \cdot 8\}$	(p_4, t_5)	$\{VM_2 \cdot 3, VM_1 \cdot 3, VM_1 \cdot 8\}$	(p_5, t_5)	$\{VM_1 \cdot 4\}$
						(p_8, t_5)	$\{VM_1 \cdot 2, VM_2 \cdot 2\}$

Nested net system model

In accordance with the product hierarchies in Fig. 7, a number of *RNets*, *MNets* and *ANets* are constructed for all the parts, assemblies and manufacturing resources contained in

**Fig. 9** The *MNet* for part variants Wt_1 and Wt_2

the process platform. Then the system model for production configuration is composed based on net nesting, as shown in Fig. 12. For illustrative simplicity, not all the *RNets*, *MNets* and *ANets* are shown in the figure. Also not included is level 6 for the *RNets*.

On completion of producing a particular item, the objects represented by port places start to send messages that request for certain operations to the objects represented by socket places. For example, when one MB_2 is produced in the *ANet* at level 2 and loaded into output buffer p_8 (also a port place), a message requesting machine setup from p_8 is sent to socket place (representing machine WcM_1) within the *PNet*. Meanwhile, place p_2 (nesting the *ANet*) holds the same colored token as that of MB_2 . The inhibitor arcs are then applied to those single resource objects, for example, machine WcM_1 (p_3) and the material handler (p_4) within the *PNet*. As a consequence, each time objects are guaranteed to perform one task only.

Table 6 Process flows and colored tokens for part variants Wt_1 and Wt_2

Process flow of Wt_1			Process flow of Wt_2		
Operations	Resource/place	Colored token	Operations	Resource/place	Colored token
Staying in buffer	Input buffer/ p_1	$Wt_1 \cdot 0$	Staying in buffer	Input buffer/ p_1	$Wt_2 \cdot 0$
WMM_1 setting up	WMM_1/p_2	$Wt_1 \cdot 1$	setting up	WMM_2/p_5	$Wt_2 \cdot 1$
Being transferred	Material handler/ p_3	$Wt_1 \cdot 2$	Being transferred	Material handler/ p_3	$Wt_2 \cdot 2$
Being machined	WMM_1/p_2	$Wt_1 \cdot 3$	Being machined	WMM_2/p_5	$Wt_2 \cdot 3$
Being transferred	Material handler/ p_3	$Wt_1 \cdot 4$	Being transferred	Material handler/ p_3	$Wt_2 \cdot 4$
Staying in buffer	WIP buffer/ p_4	$Wt_1 \cdot 5$	Staying in buffer	WIP buffer/ p_4	$Wt_2 \cdot 5$
WMM_2 setting up	WMM_2/p_5	$Wt_1 \cdot 6$	WMM_1 setting up	WMM_1/p_2	$Wt_2 \cdot 6$
Being transferred	Material handler/ p_3	$Wt_1 \cdot 7$	Being transferred	Material handler/ p_3	$Wt_2 \cdot 7$
Being machined	WMM_2/p_5	$Wt_1 \cdot 8$	Being machined	WMM_1/p_2	$Wt_2 \cdot 8$
Being transferred	Material handler/ p_3	$Wt_1 \cdot 9$	Being transferred	Material handler/ p_3	$Wt_2 \cdot 9$
Staying in buffer	Output buffer/ p_6	Wt_1	Staying in buffer	Output buffer/ p_6	Wt_2

Table 7 Resources and operations associated with the *MNet* and the *ANet*

Place	<i>MNet</i> for parts Wt_1 & Wt_2	<i>ANet</i> for assemblies MB_1 & MB_2
p_1	Input buffer	Assembly processes of assemblies AA_1 & AA_2
p_2	Machining machine WMM_1	Assembly processes of assemblies BA_1 & BA_2
p_3	Material handler	Assembly machine AIM_1
p_4	WIP buffer	Material handler
p_5	Machining machine WMM_2	WIP buffer
p_6	Output buffer	Assembly process of assembly FA_1
p_7		Assembly machine FCM_1
p_8		Output buffer

Table 8 Colored tokens, input and output transform functions for the *MNet*

Place	Colored token	Transition	Colored token	Input arc	Input arc transform function	Output arc	Output arc transform function
p_1	$\{Wt_1 \cdot 0, Wt_2 \cdot 0\}$	dt_{11}	$\{Wt_1 \cdot 0, Wt_2 \cdot 0\}$	(p_1, dt_{11})	$\{Wt_1 \cdot 0, Wt_2 \cdot 0\}$	(dp_1, dt_{11})	$\{Wt_1 \cdot 0, Wt_2 \cdot 0\}$
p_2	$\left\{ \begin{array}{l} Wt_1 \cdot 1, Wt_2 \cdot 3, \\ Wt_2 \cdot 6, Wt_2 \cdot 8 \end{array} \right\}$	dt_{12}	$\{Wt_1 \cdot 5, Wt_2 \cdot 5\}$	(p_4, dt_{12})	$\{Wt_1 \cdot 5, Wt_2 \cdot 5\}$	(dp_1, dt_{12})	$\{Wt_1 \cdot 5, Wt_2 \cdot 5\}$
		dt_{13}	$\left\{ \begin{array}{l} Wt_1 \cdot 0, Wt_1 \cdot 5, \\ Wt_2 \cdot 0, Wt_2 \cdot 5 \end{array} \right\}$	(dp_1, dt_{13})	$\left\{ \begin{array}{l} Wt_1 \cdot 0, Wt_1 \cdot 5 \\ Wt_2 \cdot 0, Wt_2 \cdot 5 \end{array} \right\}$	(p_2, dt_{13}) (p_5, dt_{13})	$\{Wt_1 \cdot 1, Wt_2 \cdot 6\}$ $\{Wt_1 \cdot 6, Wt_2 \cdot 1\}$
p_3	$\left\{ \begin{array}{l} Wt_1 \cdot 2, Wt_1 \cdot 4, Wt_1 \cdot 7, \\ Wt_1 \cdot 9, Wt_2 \cdot 2, Wt_2 \cdot 4, \\ Wt_2 \cdot 7, Wt_2 \cdot 9 \end{array} \right\}$	dt_{21}	$\{Wt_1 \cdot 1, Wt_2 \cdot 6\}$	(p_2, dt_{21})	$\{Wt_1 \cdot 1, Wt_2 \cdot 6\}$	(dp_2, dt_{21})	$\{Wt_1 \cdot 1, Wt_2 \cdot 6\}$
		dt_{22}	$\{Wt_1 \cdot 6, Wt_2 \cdot 1\}$	(p_5, dt_{22})	$\{Wt_1 \cdot 6, Wt_2 \cdot 1\}$	(dp_2, dt_{22})	$\{Wt_1 \cdot 6, Wt_2 \cdot 1\}$
p_4	$\{Wt_1 \cdot 5, Wt_2 \cdot 5\}$	dt_{23}	$\left\{ \begin{array}{l} Wt_1 \cdot 1, Wt_1 \cdot 6, \\ Wt_2 \cdot 1, Wt_2 \cdot 6 \end{array} \right\}$	(dp_2, dt_{23})	$\left\{ \begin{array}{l} Wt_1 \cdot 1, Wt_1 \cdot 6, \\ Wt_2 \cdot 1, Wt_2 \cdot 6 \end{array} \right\}$	(p_3, dt_{23})	$\left\{ \begin{array}{l} Wt_1 \cdot 2, Wt_1 \cdot 7, \\ Wt_2 \cdot 2, Wt_2 \cdot 7 \end{array} \right\}$
p_5	$\left\{ \begin{array}{l} Wt_1 \cdot 6, Wt_1 \cdot 8, \\ Wt_2 \cdot 1, Wt_2 \cdot 3 \end{array} \right\}$	t_3	$\left\{ \begin{array}{l} Wt_1 \cdot 2, Wt_1 \cdot 7, \\ Wt_2 \cdot 2, Wt_2 \cdot 7 \end{array} \right\}$	(p_3, t_3)	$\left\{ \begin{array}{l} Wt_1 \cdot 2, Wt_1 \cdot 7, \\ Wt_2 \cdot 2, Wt_2 \cdot 7 \end{array} \right\}$	(p_2, t_3) (p_5, t_3)	$\{Wt_1 \cdot 3, Wt_2 \cdot 8\}$ $\{Wt_1 \cdot 8, Wt_2 \cdot 3\}$
p_6	$\{Wt_1, Wt_2\}$	dt_{41}	$\{Wt_1 \cdot 3, Wt_2 \cdot 8\}$	(p_2, dt_{41})	$\{Wt_1 \cdot 3, Wt_2 \cdot 8\}$	(dp_4, dt_{41})	$\{Wt_1 \cdot 3, Wt_2 \cdot 8\}$
dp_1	$\left\{ \begin{array}{l} Wt_1 \cdot 0, Wt_1 \cdot 5, \\ Wt_2 \cdot 0, Wt_2 \cdot 5 \end{array} \right\}$	dt_{42}	$\{Wt_1 \cdot 8, Wt_2 \cdot 3\}$	(p_5, dt_{42})	$\{Wt_1 \cdot 8, Wt_2 \cdot 3\}$	(dp_4, dt_{42})	$\{Wt_1 \cdot 8, Wt_2 \cdot 3\}$
dp_2	$\left\{ \begin{array}{l} Wt_1 \cdot 1, Wt_1 \cdot 6, \\ Wt_2 \cdot 1, Wt_2 \cdot 6 \end{array} \right\}$	dt_{43}	$\left\{ \begin{array}{l} Wt_1 \cdot 3, Wt_1 \cdot 8, \\ Wt_2 \cdot 3, Wt_2 \cdot 8 \end{array} \right\}$	(dp_4, dt_{43})	$\left\{ \begin{array}{l} Wt_1 \cdot 3, Wt_1 \cdot 8, \\ Wt_2 \cdot 3, Wt_2 \cdot 8 \end{array} \right\}$	(p_3, dt_{43})	$\left\{ \begin{array}{l} Wt_1 \cdot 4, Wt_1 \cdot 9, \\ Wt_2 \cdot 4, Wt_2 \cdot 9 \end{array} \right\}$
dp_4	$\left\{ \begin{array}{l} Wt_1 \cdot 3, Wt_1 \cdot 8, \\ Wt_2 \cdot 3, Wt_2 \cdot 8 \end{array} \right\}$	t_5	$\left\{ \begin{array}{l} Wt_1 \cdot 4, Wt_1 \cdot 9, \\ Wt_2 \cdot 4, Wt_2 \cdot 9 \end{array} \right\}$	(p_3, t_5)	$\left\{ \begin{array}{l} Wt_1 \cdot 4, Wt_1 \cdot 9, \\ Wt_2 \cdot 4, Wt_2 \cdot 9 \end{array} \right\}$	(p_4, t_5) (p_6, t_5)	$\{Wt_1 \cdot 5, Wt_2 \cdot 5\}$ $\{Wt_1, Wt_2\}$

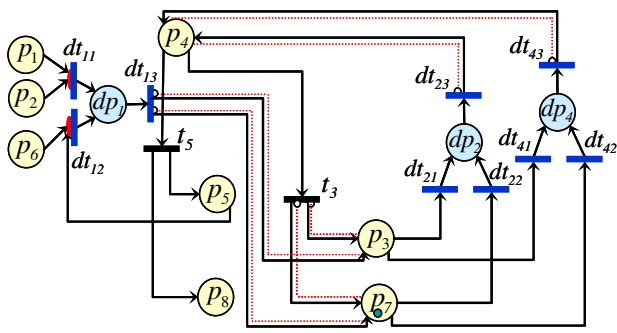


Fig. 10 The ANet for assembly variants MB_1 and MB_2

System analysis and evaluation

Deadlock detection and conflict prevention are widely adopted as performance indicators for testing a built system model (Wang 1996; Wang and Wu 1998). This section adopts these criteria for performance evaluation of the nested net system model. The focus is on the *MNet* for W_{t1} and W_{t2} .

Generally there are two types of input conflicts and one type of output conflicts (Wang 1996). Type I input conflicts involve more than one output transition to be connected to one output place. Type II conflicts involve two or more than two input places to be connected to one transition via an *OR* relation. Whenever a transition possesses more than one output place that is connected through an *OR* relation, an output conflict may occur.

Figure 13 shows the *MNet* for part family W_t . It contains no type I input conflict, as all of them are removed by transition decomposition. A type II input conflict may occur at places p_2 and p_5 , since both of them are connected to two output transitions. An output conflict may occur at transitions

dt_{13} , t_3 and t_5 , since all of them exhibit two output places with *OR* relations. Through assignment of colored tokens to places and transitions and in conjunction with input and output transform functions, the decisions regarding which transition to fire and which place to add tokens to can be determined according to the colors. Therefore, output conflicts are resolved by assigning different colors.

The deadlock detection algorithm in Wang and Wu (1998) is applied to the *MNet* in Fig. 13, as elaborated below.

(1) The initial state is set as that raw materials of two parts are in the input buffer. And the goal state is set as that both two parts are produced and loaded into the output buffer. The initial marking M_0 and the goal state marking M_g become the following,

$$\begin{aligned} M_0 &= (\mu(p_1), \mu(p_2), \mu(p_3), \mu(p_4), \mu(p_5), \mu(p_6), \\ &\quad \mu(dp_1), \mu(dp_2), \mu(dp_4),) \\ &= (W_{t1} \cdot 0 + W_{t2} \cdot 0, 0, 0, 0, 0, 0, 0, 0, 0) \\ M_g &= (0, 0, 0, 0, 0, 0, W_{t1} + W_{t2}, 0, 0, 0). \end{aligned}$$

(2) Construct an incidence matrix, W , such that,

$$\begin{aligned} W^+ &= [w_{ij}^+]_{m \times n}, \quad \text{where } w_{ij}^+ = O(p_i, t_j / C^{t_j}), \\ &\quad \forall i \in [1, m], j \in [1, n]; \text{ and} \\ W^- &= [w_{ij}^-]_{m \times n}, \quad \text{where } w_{ij}^- = I(p_i, t_j / C^{t_j}), \\ &\quad \forall i \in [1, m], j \in [1, n]. \end{aligned}$$

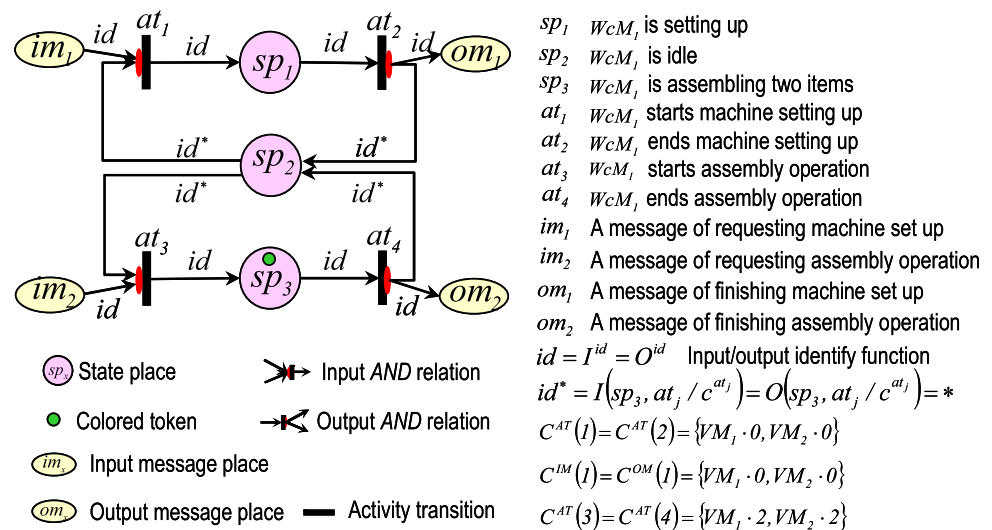
Set $m = 9$ (the total number of places in the *MNet*) and $n = 11$ (the total number of transitions in the *MNet*). Then,

Table 9 Process flows for assembly variants MB_1 and MB_2

Process flow of MB_1			Process flow of MB_2		
Operation	Resource (or process)/ place	Colored token	Operation	Resource (or process)/ place	Colored token
Assembly AA_1 is produced	Assembly process/ p_1	AA_1	Assembly AA_2 is produced	Assembly process/ p_1	AA_2
Assembly BA_1 is produced	Assembly process/ p_2	BA_1	Assembly BA_2 is produced	Assembly process/ p_2	BA_2
AIM_1 setting up	AIM_1/p_3	$MB_1 \cdot 0$	AIM_1 setting up	AIM_1/p_3	$MB_2 \cdot 0$
Being transferred	Material handler/ p_4	$MB_1 \cdot 1$	Being transferred	Material handler/ p_4	$MB_2 \cdot 1$
Being assembled	AIM_1/p_3	$MB_1 \cdot 2$	Being assembled	AIM_1/p_3	$MB_2 \cdot 2$
Being transferred	Material handler/ p_4	$MB_1 \cdot 3$	Being transferred	Material handler/ p_4	$MB_2 \cdot 3$
Staying in buffer	WIP buffer/ p_5	$MB_1 \cdot 4$	Staying in buffer	Output buffer/ p_8	MB_2
Assembly FA_1 is produced	Assembly process/ p_6	FA_1			
FCM_1 setting up	FCM_1/p_7	$MB_1 \cdot 5$			
Being transferred	Material handler/ p_4	$MB_1 \cdot 6$			
Being assembled	FCM_1/p_7	$MB_1 \cdot 7$			
Being transferred	Material handler/ p_4	$MB_1 \cdot 8$			
Staying in buffer	Output buffer/ p_8	MB_1			

Table 10 Colored tokens, input and output transform functions for the *ANet*

Place	Colored token	Transition	Colored token	Input arc	Input arc transform function	Output arc	Output arc transform function
p_1	$\{AA_1, AA_2\}$	dt_{11}	$\{AA_1BA_1, AA_2BA_2\}$	(p_1, dt_{11})	$\{AA_1BA_1, AA_2BA_2\}$	(dp_1, dt_{11})	$\{AA_1BA_1, AA_2BA_2\}$
p_2	$\{BA_1, BA_2\}$			(p_2, dt_{11})	$\{AA_1BA_1, AA_2BA_2\}$		
p_3	$\{MB_1 \cdot 0, MB_1 \cdot 2, MB_2 \cdot 0, MB_2 \cdot 2\}$	dt_{12}	$\{MB_1 \cdot 4FA_1\}$	(p_5, dt_{12})	$\{MB_1 \cdot 4FA_1\}$	(dp_1, dt_{12})	$\{MB_1 \cdot 4FA_1\}$
				(p_6, dt_{12})	$\{MB_1 \cdot 4FA_1\}$		
p_4	$\{MB_1 \cdot 1, MB_1 \cdot 3, MB_2 \cdot 1, MB_2 \cdot 3, MB_1 \cdot 6, MB_1 \cdot 8\}$	dt_{13}	$\{MB_1 \cdot 4FA_1, AA_1BA_1, AA_2BA_2\}$	(dp_1, dt_{13})	$\{MB_1 \cdot 4FA_1, AA_1BA_1, AA_2BA_2\}$	(p_3, dt_{13})	$\{MB_1 \cdot 0, MB_2 \cdot 0\}$
p_5	$\{MB_1 \cdot 4\}$	dt_{21}	$\{MB_1 \cdot 0, MB_2 \cdot 0\}$	(p_3, dt_{21})	$\{MB_1 \cdot 0, MB_2 \cdot 0\}$	(dp_2, dt_{21})	$\{MB_1 \cdot 5\}$
p_6	$\{FA_1\}$	dt_{22}	$\{MB_1 \cdot 5\}$	(p_7, dt_{22})	$\{MB_1 \cdot 5\}$	(dp_2, dt_{22})	$\{MB_1 \cdot 5\}$
p_7	$\{MB_1 \cdot 5, MB_2 \cdot 7\}$	dt_{23}	$\{MB_1 \cdot 0, MB_2 \cdot 0, MB_1 \cdot 5\}$	(dp_2, dt_{23})	$\{MB_1 \cdot 0, MB_2 \cdot 0, MB_1 \cdot 5\}$	(p_4, dt_{23})	$\{MB_1 \cdot 0, MB_2 \cdot 0, MB_1 \cdot 5\}$
p_8	$\{MB_1, MB_2\}$	t_3	$\{MB_1 \cdot 1, MB_2 \cdot 1, MB_1 \cdot 6\}$	(p_4, t_3)	$\{MB_1 \cdot 1, MB_2 \cdot 1, MB_1 \cdot 6\}$	(p_3, t_3)	$\{MB_1 \cdot 2, MB_2 \cdot 2\}$
						(p_7, t_3)	$\{MB_1 \cdot 7\}$
dp_1	$\{MB_1 \cdot 4FA_1, AA_1BA_1, AA_2BA_2\}$	dt_{41}	$\{MB_1 \cdot 2, MB_2 \cdot 2\}$	(p_3, dt_{41})	$\{MB_1 \cdot 2, MB_2 \cdot 2\}$	(dp_4, dt_{41})	$\{MB_1 \cdot 2, MB_2 \cdot 2\}$
dp_2	$\{MB_2 \cdot 0, MB_1 \cdot 0, MB_1 \cdot 5\}$	dt_{42}	$\{MB_1 \cdot 7\}$	(p_7, dt_{42})	$\{MB_1 \cdot 7\}$	(dp_4, dt_{42})	$\{MB_1 \cdot 7\}$
		dt_{43}	$\{MB_1 \cdot 2, MB_2 \cdot 2, MB_1 \cdot 7\}$	(dp_4, dt_{43})	$\{MB_1 \cdot 2, MB_2 \cdot 2, MB_1 \cdot 7\}$	(p_4, dt_{43})	$\{MB_1 \cdot 2, MB_2 \cdot 2, MB_1 \cdot 7\}$
dp_4	$\{MB_2 \cdot 2, MB_1 \cdot 2, MB_1 \cdot 7\}$	t_5	$\{MB_1 \cdot 3, MB_2 \cdot 3, MB_1 \cdot 8\}$	(p_4, t_5)	$\{MB_1 \cdot 3, MB_2 \cdot 3, MB_1 \cdot 8\}$	(p_5, t_5)	$\{MB_1 \cdot 4\}$
						(p_8, t_5)	$\{MB_1, MB_2\}$

Fig. 11 Specification of internal behavior of assembly machine WcM_1 

$$W^+ = \begin{bmatrix} dt_{11} & dt_{12} & dt_{13} & dt_{21} & dt_{22} & dt_{23} & t_3 & dt_{41} & dt_{42} & dt_{43} & t_5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ dp_1 \\ dp_2 \\ dp_3 \end{matrix}$$

$$W^- = \begin{bmatrix} id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & id & 0 & 0 & 0 & id & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & id & 0 & 0 & 0 \\ 0 & id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & id & 0 & 0 & 0 & 0 & id & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & id & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & id & 0 \end{bmatrix}$$

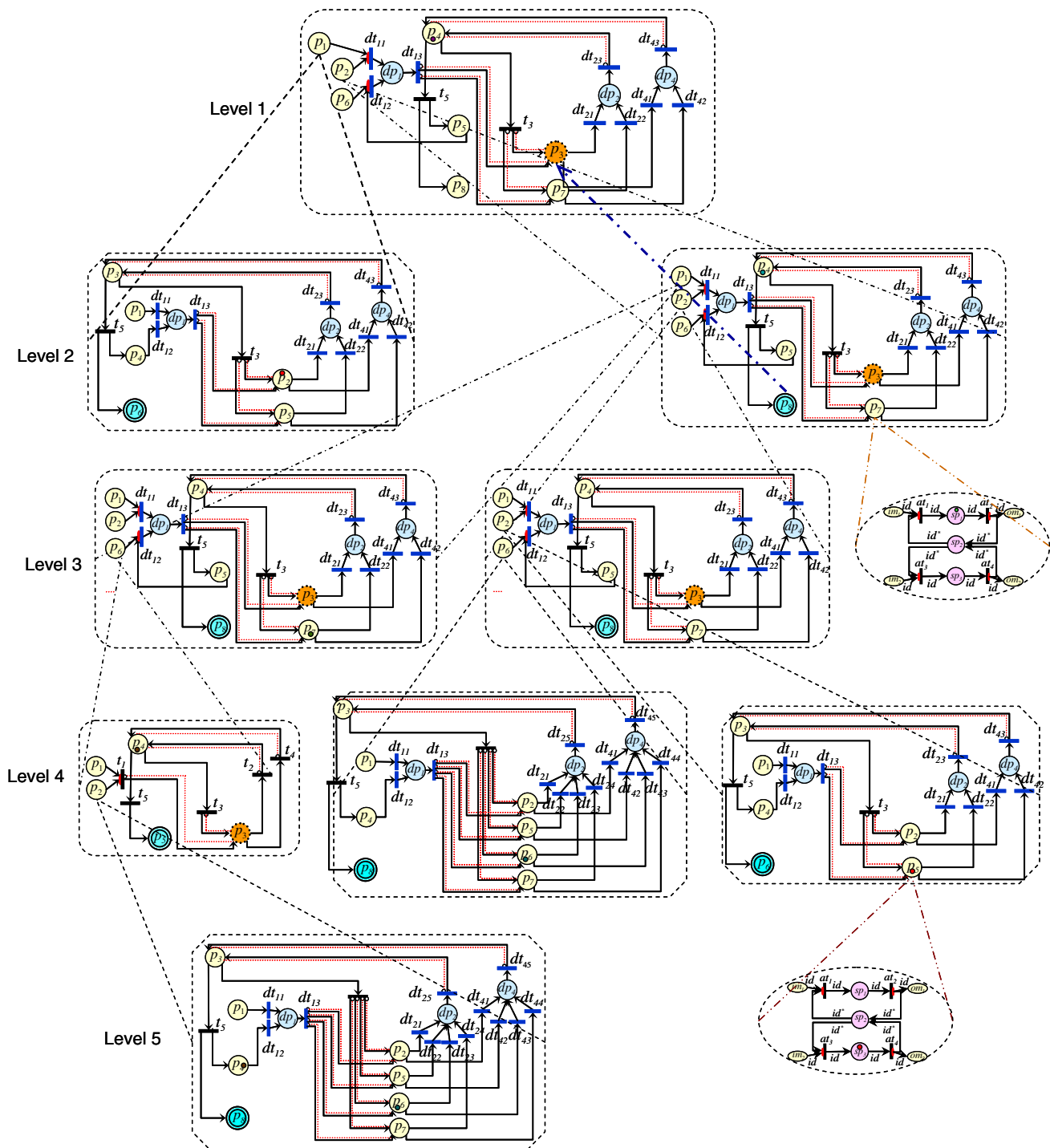


Fig. 12 Nested net system model for VM_1 and VM_2

$$W = W^+ - W^-$$

$$= \begin{bmatrix} -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & O_1 & -id & 0 & 0 & O_2 & -id & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & O_3 & -id & 0 & 0 & O_4 & -id \\ 0 & -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & O_5 \\ 0 & 0 & O_6 & 0 & -id & 0 & O_7 & 0 & -id & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & O_8 \\ id & id & -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & id & id & -id & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & id & id & -id & 0 \end{bmatrix}.$$

(3) At the initial marking M_0 (i.e., $k=0$), the set of enabled transitions, $T_{enable-0}$, are related to their firing colored tokens c^{t_j} , that is, $T_{enable-0} = \{dt_{11}/W_{t1} \cdot 0, dt_{11}/W_{t2} \cdot 0\}$.

(4) Transition $dt_{11}/W_{t1} \cdot 0$ is selected to fire.

(5) The characteristic vector \bar{S}_k of a firing sequence S is set to have entry c^{t_j} for the transition selected to fire and 0 for all others. Hence,

$$\bar{S}_0 = (dt_{11}, dt_{12}, dt_{13}, dt_{21}, dt_{22}, dt_{23}, t_3, dt_{41}, dt_{42}, dt_{43}, t_5)^{-1}$$

$$= (W_{t1} \cdot 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^{-1}$$

(6) The following state marking, $M_{k+1} = M_{0+1} = M_1$, after firing the transition $dt_{11}/W_{t1} \cdot 0$ is computed according to $M_{k+1}^{-1} = M_k^{-1} + W\bar{S}_k$. Thereby, $M_1 = (W_{t2} \cdot 0, 0, 0, 0, 0, 0, 0, W_{t1} \cdot 0, 0, 0, 0)$. As $M_1 \neq M_g$, k is set to be $k+1$ (i.e., $k=0+1=1$). Then go to step (3).

As for marking M_1 , it is true that $T_{enable-1} = \{dt_{13}/W_{t1} \cdot 0, dt_{11}/W_{t2} \cdot 0\}$. Thus transition $dt_{11}/W_{t2} \cdot 0$ is selected to fire. Then $\bar{S}_1 = (W_{t2} \cdot 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^{-1}$. Therefore $M_{k+1} = M_2$ is computed as $M_2 = (0, 0, 0, 0, 0, 0, 0, W_{t1} \cdot 0 + W_{t2} \cdot 0, 0, 0, 0)$. Since $M_2 \neq M_g$, k is set to 2. Go to step (3).

Since $T_{enable-2} = \{dt_{13}/W_{t1} \cdot 0, dt_{13}/W_{t2} \cdot 0\}$, $dt_{13}/W_{t1} \cdot 0$ is selected to fire. As a result, $\bar{S}_2 = (0, 0, W_{t1} \cdot 0, 0, 0, 0, 0, 0, 0, 0, 0)^{-1}$, and thus M_3 is computed as $M_3^{-1} = M_2^{-1} + W\bar{S}_2$. According to the output functions of the $MNet$ (Fig. 12), it can infer that $O_6(W_{t1} \cdot 0) = 0$ and $O_6(W_{t1} \cdot 0) = W_{t1} \cdot 1$. Considering the input identity function defined for the $MNet$, we have $-id(W_{t1} \cdot 0) = -W_{t1} \cdot 0$. Furthermore,

$$M_3^{-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ W_{t1} \cdot 0 + W_{t2} \cdot 0 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \begin{bmatrix} -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & O_1 & -id & 0 & 0 & O_2 & -id & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & O_3 & -id & 0 & 0 & O_4 \\ 0 & -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & O_5 \\ 0 & 0 & O_6 & 0 & -id & 0 & O_7 & 0 & -id & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & O_8 \\ id & id & -id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & id & id & -id & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & id & id & -id & 0 \end{bmatrix}$$

$$\times \begin{bmatrix} 0 \\ 0 \\ W_{t1} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

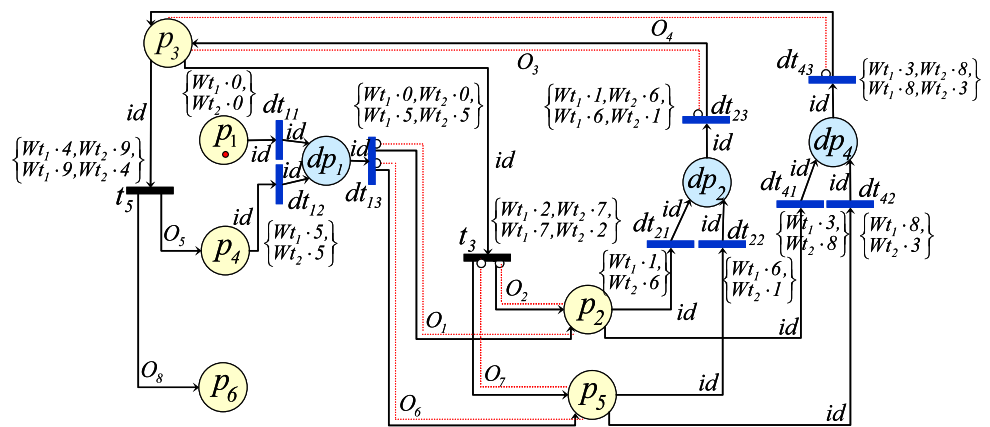
$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ W_{t1} \cdot 0 + W_{t2} \cdot 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ O_1(W_{t1} \cdot 0) \\ 0 \\ 0 \\ O_6(W_{t1} \cdot 0) \\ 0 \\ -id(W_{t1} \cdot 0) \\ 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ W_{t1} \cdot 0 + W_{t2} \cdot 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ W_{t1} \cdot 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -W_{t1} \cdot 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ W_{t1} \cdot 1 \\ 0 \\ 0 \\ 0 \\ W_{t2} \cdot 0 \\ 0 \\ 0 \end{bmatrix}$$

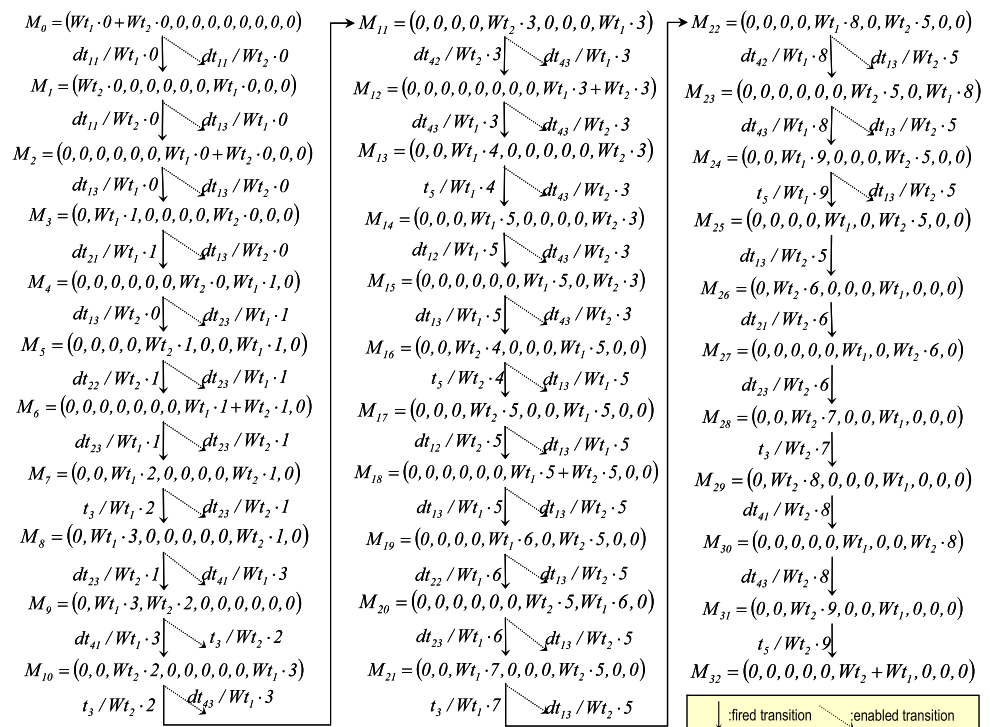
Since $M_3 \neq M_g$, the deadlock detection process is continued. On completion, the deadlock analysis reaches the final goal marking, $M_g = (0, 0, 0, 0, 0, 0, W_{t1} + W_{t2}, 0, 0, 0)$. Figure 14 shows an example of feasible sequence that leads to the final goal state. Accordingly, it concludes that the $MNet$ for W_{t1} and W_{t2} is live and deadlock free.

Conclusions

The multilevel system of nested colored object-oriented PNs with changeable structures excels in modeling details of production configuration at different levels of abstraction. The differentiation mechanism inherent in CPNs and OPNs facilitates the representation of various process elements in relation to diverse product items. Attaching data to tokens to form colored tokens helps track the manufacturing resources, performed operations and product items throughout configuration, and thus leads to efficiency in handling multiple configuration constraints. Net nesting alleviates the complexity of production configuration through decomposition. Through

Fig. 13 The *MNet* of part variants Wt_1 and Wt_2 

$$\begin{aligned}
 O_1 &= \begin{cases} O(p_2, dt_{13} / Wt_1 \cdot 0) = Wt_1 \cdot 1 \\ O(p_2, dt_{13} / Wt_2 \cdot 5) = Wt_2 \cdot 6 \end{cases} & O_2 &= \begin{cases} O(p_2, t_3 / Wt_1 \cdot 2) = Wt_1 \cdot 3 \\ O(p_3, t_3 / Wt_2 \cdot 7) = Wt_2 \cdot 8 \end{cases} & O_3 &= \begin{cases} O(p_3, dt_{23} / Wt_1 \cdot 1) = Wt_1 \cdot 2 \\ O(p_3, dt_{23} / Wt_1 \cdot 6) = Wt_1 \cdot 7 \\ O(p_3, dt_{23} / Wt_2 \cdot 1) = Wt_2 \cdot 2 \\ O(p_3, dt_{23} / Wt_2 \cdot 6) = Wt_2 \cdot 7 \end{cases} \\
 O_5 &= \begin{cases} O(p_4, t_5 / Wt_1 \cdot 4) = Wt_1 \cdot 5 \\ O(p_4, t_5 / Wt_2 \cdot 4) = Wt_2 \cdot 5 \end{cases} & O_6 &= \begin{cases} O(p_5, dt_{13} / Wt_2 \cdot 0) = Wt_2 \cdot 1 \\ O(p_5, dt_{13} / Wt_1 \cdot 5) = Wt_1 \cdot 6 \end{cases} & O_4 &= \begin{cases} O(p_3, dt_{43} / Wt_1 \cdot 3) = Wt_1 \cdot 4 \\ O(p_3, dt_{43} / Wt_1 \cdot 8) = Wt_1 \cdot 9 \\ O(p_3, dt_{43} / Wt_2 \cdot 3) = Wt_2 \cdot 4 \\ O(p_3, dt_{43} / Wt_2 \cdot 8) = Wt_2 \cdot 9 \end{cases} \\
 O_7 &= \begin{cases} O(p_5, t_3 / Wt_1 \cdot 7) = Wt_1 \cdot 8 \\ O(p_5, t_3 / Wt_2 \cdot 2) = Wt_2 \cdot 3 \end{cases} & O_8 &= \begin{cases} O(p_6, t_5 / Wt_1 \cdot 9) = Wt_1 \\ O(p_6, t_5 / Wt_2 \cdot 9) = Wt_2 \end{cases}
 \end{aligned}$$

Fig. 14 A feasible firing sequence leading to the goal marking

specifying port places, socket places and message exchange, the continuity of model execution can be maintained from the lowest level to the highest level.

Comparing with existing PN tools for either manufacturing of parts or assembly processes, the proposed nested PNs demonstrate an advantage in elucidating the entire production

process from part manufacturing to final product assembly. It contributes to the consensus on a holistic view of product differentiation and process variation. The executability of PNs allows the possibility of developing simulation, mathematical analysis and real-time control models for managing product and process variety while leveraging upon existing

product families and process platforms. Decision support to production configuration also paves an avenue for future research.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

- Adamou, M., Zerhouni, S. N., & Bourjault, A. (1998). Hierarchical modeling and control of flexible assembly systems using object-oriented Petri nets. *International Journal of Computer Integrated Manufacturing*, 11(1), 18–33. doi:10.1080/095119298130949.
- Benjaafar, S., Heragu, S. S., & Irani, S. A. (2002). Next generation factory layouts: Research challenges and recent progress. *Interfaces*, 32(6), 58–76. doi:10.1287/inte.32.6.58.6473.
- Bozarth, C., & McDermott, C. (1998). Configurations in manufacturing strategy: A review and directions for future research. *Journal of Operations Management*, 16(3), 427–439. doi:10.1016/S0272-6963(98)00022-9.
- Hayes, R., & Wheelwright, S. (1984). *Restoring our Competitive Edge*. New York, NY: Wiley.
- He, D. W., Strege, B., Tolle, H., & Kusiak, A. (2000). Decomposition in automatic generation of Petri nets for manufacturing system control and scheduling. *International Journal of Production Research*, 38(6), 1437–1457. doi:10.1080/002075400188942.
- Hill, T. (1994). *Manufacturing strategy: Text and cases*. Homewood, IL: Irwin.
- Jensen, K. (1992). *Colored Petri Nets: Basic concepts, analysis methods and practical use*. Berlin: Springer-Verlag.
- Jiang, Z., Zuo, M. J., Tu, P. Y., & 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(6), 445–458. doi:10.1007/s001700050089.
- Jiao, J., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: A start-of-the-art review. *Journal of Intelligent Manufacturing*, 18(1), 5–29. doi:10.1007/s10845-007-0003-2.
- Jiao, J., Tseng, M. M. (2004). Customizability analysis in design for mass customization. *Computer-Aided Design*, 36(8), 745–757. doi:10.1016/j.cad.2003.09.012.
- Jiao, J., Zhang, L., & Prasanna, K. (2004). Process variety modeling for process configuration in mass customization: An approach based on object-oriented Petri-Nets with changeable structures. *International Journal of Flexible Manufacturing Systems*, 16(4), 335–361. doi:10.1007/s10696-005-5171-9.
- Kumar, R., Kumar, S., & Tiwari, M. K. (2005). An expert enhanced coloured fuzzy Petri net approach to reconfigurable manufacturing systems involving information delays. *International Journal of Advanced Manufacturing Technology*, 26, 922–933. doi:10.1007/s00170-003-1890-9.
- Kumar, R. R., Kumar Singh, A., & Tiwari, M. K. (2004). A fuzzy based algorithm to solve the machine-loading problems of a FMS and its neuro fuzzy Petri net model. *International Journal of Advanced Manufacturing Technology*, 23, 318–341. doi:10.1007/s00170-002-1499-4.
- Kumar Singh, A., Tiwari, M. K., & Mukhopadhyay, S. K. (2003). Modelling and planning of the disassembly processes using an enhanced expert Petri net. *International Journal of Production Research*, 41(16), 3761–3792. doi:10.1080/0020754031000109125.
- Li, X. O., & Lara-Rosano. (1999). Modeling an electronic component manufacturing system using object oriented colored Petri nets. In *Proceedings of the 6th IEEE International Conference on Electronics, Circuits, and Systems*, Pafos, Cyprus.
- Lopez-Mellado, E., & 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), 122–136.
- Martinez, M. T., Favrel, J., & Ghodous, P. (2000). Product family manufacturing plan generation and classification. *Concurrent Engineering: Research and Applications*, 8(1), 12–22.
- Mittal, S., & Frayman, F. (1989). Towards a generic model of configuration tasks. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence* (pp. 1395–1401). San Francisco: Morgan Kaufmann.
- Morandin Junior, O., & Kato, E. R. R. (2003). Virtual Petri nets as a modular modeling method for planning and control tasks of FMS. *International Journal of Computer Integrated Manufacturing*, 18(2–3), 100–106.
- Morandin Junior, O., Kato, E. R. R., Politano, P. R., Camargo, H. A., Porto, A. J. V., & Inamasu, R. Y. (2000). A modular modeling approach for automated manufacturing systems based on shared resources and process planning using Petri nets. In *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN.
- Peterson, J. L. (1981). *Petri Net theory and the modeling of systems*. Englewood Cliffs, NJ: Prentice-Hall.
- Prasad, B. (1998) Designing products for variety and how to manage complexity. *Journal of Product & Brand Management*, 7(3), 208–222. doi:10.1108/10610429810222840.
- Sabin, D., & Weigel, R. (1998) Product configuration frameworks — a survey. *IEEE Intelligent Systems & Their Applications*, 13(4), 42–49.
- Sawhney, M. S. (1998). Leveraged high-variety strategies: From portfolio thinking to platform thinking. *Journal of the Academy of Marketing Science*, 26(1), 54–61. doi:10.1177/0092070398261006.
- Schierholt, K. (2001). Process configuration: Combining the principles of product configuration and process planning. *AIEDAM*, 15(5), 411–424.
- Siddique, Z. (2005). Assembly process selection to minimize existing assembly system modification cost during new product family member design *Proceedings of ASME Design Engineering Technical Conferences DETC2005-85016*, Long Beach, CA.
- Simpson, T. W. (2004). Product platform design and customization: Status and promise. *AIEDAM*, 18(1), 3–20.
- Stobaugh, R., & Telesio, P. (1983). Match manufacturing policies and product strategy. *Harvard Business Review*, 61(2), 113–120.
- Thomas, J. P., Nissanke, N., Baker, K. D. (1996). A hierarchical Petri net framework for the representation and analysis of assembly. *IEEE Transactions of Robotics and Automation*, 12(2), 268–279. doi:10.1109/70.488946.
- Wang, L. C. (1996). Object-oriented Petri nets for modeling and analysis of automated manufacturing systems. *Computer Integrated Manufacturing Systems*, 26(2), 111–125. doi:10.1016/0951-5240(95)00032-1.
- Wang, L. C., & Wu, S. Y. (1998). Modeling with colored timed object-oriented Petri nets for automated manufacturing systems. *Computers and Industrial Engineering*, 34(2), 463–480. doi:10.1016/S0360-8352(97)00145-9.
- Wortmann, J. C., Muntslag, D. R., & Timmermans, P. J. M. (1997). *Customer Driven Manufacturing*. London: Chapman & Hall.