



**HAL**  
open science

# A Methodology Integrating Petri Nets and Knowledge-based Systems to Support Process Family Planning

Linda Zhang, Qianli Xu, Petri Helo

► **To cite this version:**

Linda Zhang, Qianli Xu, Petri Helo. A Methodology Integrating Petri Nets and Knowledge-based Systems to Support Process Family Planning. International Journal of Production Research, 2011, pp.1. 10.1080/00207543.2011.555787 . hal-00732518

**HAL Id: hal-00732518**

**<https://hal.science/hal-00732518>**

Submitted on 15 Sep 2012

**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.



### **A Methodology Integrating Petri Nets and Knowledge-based Systems to Support Process Family Planning**

Journal:	<i>International Journal of Production Research</i>
Manuscript ID:	TPRS-2010-IJPR-1157
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	21-Nov-2010
Complete List of Authors:	Zhang, Linda; University of Groningen, Operations Xu, Qianli; Nanyang Technological University Helo, Petri; University of Vaasa
Keywords:	PETRI NETS, INTEGRATION, PROCESS PLANNING
Keywords (user):	Product family, Process family

SCHOLARONE™  
Manuscripts

# A Methodology Integrating Petri Nets and Knowledge-based Systems to Support Process Family Planning

Linda L. Zhang<sup>\*1</sup>, Qianli Xu<sup>2</sup>, Petri Helo<sup>3</sup>

<sup>1</sup>Department of Operations, University of Groningen, Groningen, The Netherlands

<sup>2</sup>Department of Computer Graphics & Interfaces,

Institute for Infocomm Research, A\*STAR, Singapore

<sup>3</sup>Logistics Systems Research Group, University of Vaasa, Vaasa, Finland

## Abstract

Planning production processes for product families has been well recognized as an effective means of achieving successful product family development. However, most existing approaches do not lend themselves to planning production processes with focus on the optimality of the cohort of a product family. This paper addresses process family planning for product families. In view of the advantages of Petri nets (PNs) for modeling large systems, the potential of knowledge-based systems (KBSs) for solving complex problems and the analogy in between, we develop a methodology by integrating PNs and KBSs to support process family planning. An integrated product-process family structure, called IP<sup>2</sup>S, is proposed to organize all data pertaining to a product family and the corresponding process family, thereby anchoring planning to one platform. With the IP<sup>2</sup>S, a formal PN model of process family planning is further developed by integrating the principles of several well-defined PN extensions. Thus, this paper also contributes to visualizing the dynamic behaviors and reasoning behind process family planning. The methodology is applied to process family planning for a truck family. The preliminary results demonstrate the feasibility and potential of using the methodology to support process family planning.

**Keywords:** Product family, process family, production processes, Petri net, knowledge-based system

---

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

## 1. INTRODUCTION

A production process describes routings, operations and manufacturing resources (e.g., machines, tools, fixtures, jigs) that are adopted to materialize a design. In accordance with the items located at different levels of a product's BOM (bill of materials) hierarchy, it includes a number of sequenced subprocesses, each of which consists of multiple ordered manufacturing and/or assembly operations. Most existing approaches to planning processes, be they manufacturing processes for parts or assembly processes for assemblies, are of trial and error based on individual planner's experience and intuition due to the lack of well-structured mechanisms (Huang et al., 2004; Tong et al., 2004; Zha, 2004). As a result, it is not uncommon that given the same product, planners develop different production processes. This inevitably causes unnecessary production changeovers, such as variations in routings, operations and manufacturing resources, on shop floors. Production performance in terms of cost, lead-time and quality deteriorates with the inclusion of avoidable production changeovers (Zhang, 2007; Zhang and Rodrigues, 2009). Furthermore, these approaches address the optimality of individual products without considering the impact from/on the production of other products, which are to be produced using the same manufacturing resources existing on shop floors (Gupta and Krishnan, 1998).

To survive, manufacturing companies nowadays strive to develop product families, each of which consists of a number of customized products. Successful product family development relies on the efficiency in both designing and producing product families (Kuttner and Karjust, 2006). On one hand, most of the solution methods reported in the literature can assist companies to quickly design these customized products (or product family members) at low costs (Forza and Salvador, 2002; Simpson, 2004). On the other hand, the key to achieving efficiency in producing these products is to plan production processes that can maintain production stability by eliminating the unnecessary production changeovers and by reducing the complexities in material handling on shop floors (Kuipers et al., 2004; Zhang, 2007). Given the existing manufacturing capabilities, such production processes can only be planned by considering the optimality of the cohort of a product family rather than individual products (Kuttner and Karjust, 2006; Zhang and Rodrigues, 2009). In this regard, the

1  
2  
3 traditional approaches to planning processes (discussed above) do not lend themselves to realizing  
4 product family production efficiency. Consequently, it raises the importance in developing solution  
5 methods to plan production processes for a product family by considering the production optimality of  
6 the cohort (i.e., process family planning). In this study, we, thus, tackle process family planning by  
7 developing a methodology.  
8  
9

10  
11  
12  
13  
14 Process family planning is inherently complex due to the large number of individual products  
15 together with the resulting diverse component items (Hunynen, 1992; Martinez et al., 2000). The  
16 complexity is often exacerbated by 1) the multiple feasible operations and the corresponding  
17 machines, tools and fixtures; 2) many routing alternatives; and 3) various precedence constraints  
18 among operations (Zha, 2004; Zhang and Jiao, 2008). Moreover, the finite manufacturing resources  
19 and the often conflicting production performance metrics make the problem more difficult. Other  
20 researchers have also reported the difficulties and complexities in planning production processes in  
21 the context of product family development (Hayes et al., 2005; Pisano, 1997).  
22  
23  
24  
25  
26  
27  
28  
29  
30

31  
32 Petri nets (PNs) are able to shed light on the reasoning behind complex processes by modeling  
33 and visualizing system's dynamic behaviors (Peterson, 1981). Compared with the complicated  
34 symbols, variables, equations, etc. in mathematical models, the graphical representation of PN models  
35 offers an easier intuitive understanding of systems in consideration (Zha and Du, 2001). In addition,  
36 the underlying mathematical equations enable formal analysis of PN models (Zha et al., 2002).  
37 Another advantage of PNs is that extensions and formalisms can be developed based on conventional  
38 PNs to meet the modeling requirements of different problem domains. Some well-recognized  
39 extensions include colored PNs (CPNs; Jensen, 1997), timed PNs (TPNs; Ramachandani, 1974) and  
40 nested PNs (NPNs; Lomazova, 2000), etc.  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50

51  
52 As a branch of artificial intelligence (AI), knowledge-based systems (KBSs) refer to computer  
53 systems that rely on knowledge and reasoning to perform difficult tasks, which are usually undertaken  
54 by domain experts (Mockler and Dologite, 1992; Parsaye and Chignell, 1988). They are able to  
55 handle the complexities involved in decision making, thereby supporting companies to quickly make  
56 better decisions (Haddock and Hartshorn, 1989; Harhalakis et al., 1994; Hynynen, 1992; Liang, 1988;  
57 Rao et al., 2005; Singh and Sekhon, 2002). When the processes in consideration are large and involve  
58  
59  
60

1  
2  
3 complex decision making, companies often resort to KBSs. A KBS not only helps retain the  
4 knowledge scattered in the memories of individual domain experts but assists to find solutions using a  
5 mechanism by mimicking the reasoning process of domain experts (Mockler and Dologite, 1992).  
6  
7 Recognizing the correspondence between PNs and KBSs, researchers have reported integrated  
8 solution methodologies based on PNs and KBSs to support modeling and decision making in design,  
9  
10 planning and control in manufacturing (e.g., Zha and Du, 2001; Zha et al., 2002; Zha et al., 1998a;  
11  
12 1998b).

13  
14  
15  
16  
17  
18 In view of the above advantage of PNs for modeling large systems, the potential of KBSs for  
19 solving complex problems and the analogy in between, we develop the methodology to support  
20 process family planning by integrating PNs and KBSs. In the methodology, an integrated product-  
21 process family structure, termed as IP<sup>2</sup>S, is proposed to organize all data pertaining to the product and  
22 process families. The rationale of such an IP<sup>2</sup>S is consistent with that of the generic product-process  
23 structure in our previous work (Jiao et al., 2007; Jiao and Zhang, 2008). Since we focus this study on  
24 these industries, where design and manufacturing technologies are relatively stable (e.g., automobile  
25 industries, PC industries, bicycle industries), the IP<sup>2</sup>S is assumed fixed for a product family during its  
26 lifecycle. Built upon the IP<sup>2</sup>S, a formal PN model of process family planning is subsequently  
27 developed. The PN model not only captures the elements (e.g., product and process family data) and  
28 their relationships inherent in process family planning but visualizes the dynamic behavior of process  
29 family planning, i.e., how the production processes for given product family members are planned. To  
30 implement the proposed methodology, we further develop a prototype for a truck family. In light of  
31 the fact that elaboration on prototype development does not provide additional contribution, we focus  
32 on the definitions and concepts involved in the IP<sup>2</sup>S and the formal PN model. By doing so, we also  
33 avoid the risk of developing a lengthy and loosely focused paper.

34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53 In the rest of this paper, Section 2 presents the literature relevant to planning processes and the  
54 applications of PNs and KBSs in manufacturing industries. Section 3 discusses an overview of the  
55 proposed methodology. Sections 4 and 5 present the details of the construction of the IP<sup>2</sup>S and the  
56 specification of the PN model, respectively. Section 6 introduces the application case and the  
57 preliminary results. Section 7 concludes this paper by outlining avenues for future research.

## 2. RELATED WORK

### 2.1 Planning processes for parts and assemblies

As an important link between design and production, planning production processes has received much attention. In the process planning community, researchers present various methods, methodologies and system prototypes to address process planning for parts. Many studies have been reported with focus on specific industries. For example, [Choi et al. \(1999\)](#), [Lin and Chang \(1997\)](#), [Lin and Deng \(2001\)](#) and [Tor et al. \(2005\)](#) present their methods for planning stamping processes in progressive die design. [Tong and Kwong \(2000\)](#), [Tong et al. \(2004\)](#), [Yeung and Yuen \(1996\)](#) and [Yeung et al. \(1995\)](#) apply AI techniques to facilitate process design for transfer molding of electronic packages in the semiconductor industry. In addition, process planning is approached from a variety of aspects, such as plan evaluation and optimization ([Huang et al., 2004](#); [Li et al., 2004](#)), integration of planning and scheduling ([Aldakhilallah and Ramesh, 1998](#)), planning in one-of-a-kind production ([Tu et al., 2000](#)), generic process planning support independent of specific problem domains ([Yuen et al., 2003](#)).

Recognizing the importance of assembly operations in production, researchers in the assembly planning community have made endeavors to assembly planning, aiming to generate better assembly plans. Most published articles focus on assembly modeling, sequencing and evaluation ([Yuan and Li, 2000](#); [Zha, 2004](#)). In assembly modeling, PN techniques have been intensively investigated to develop models ([Ben-Arieh et al., 2004](#); [Zha, 2004](#); [Zha and Du, 2001](#); [Zhang, 1989](#)). In assembly sequencing and evaluation, the advanced problem-solving methodologies in AI have been adopted to obtain assembly plans ([Thomas, 1992](#); [Zha et al., 1998a](#); [Zha et al., 1998b](#); [Zha and Du, 2001](#); [Zha, 2004](#); [Zhang, 1989](#)).

In summary, most existing studies address either assembly planning or process planning for assemblies or parts at a lower level with focus on operations details, as shown in [Figure 1](#). Essentially, they attempt to determine specific process parameters (e.g., cutting speed, collision-free path, feed rate). In addition, they focus on the process optimality of individuals, be it a part or an assembly. To ensure the effectiveness of these assembly/process planning activities, the production processes of



1  
2  
3 process planning, manufacturing resources planning and shop floor control. In view of the advantages  
4 of high level PNs, Muro-Medrano et al. (1998) introduce knowledge representation-oriented PNs  
5 (KRONs) for modeling discrete event systems. Based on KRONs, they also report the design of a  
6 prototype for scheduling and planning. Similarly, this study attempts to facilitate process family  
7 planning by capitalizing on the advantages of integrating PNs and KBSs.  
8  
9  
10  
11  
12

### 13 14 15 **3. METHODOLOGY OVERVIEW**

16  
17 The complexities in process family planning have raised several requirements towards the design  
18 of appropriate methodologies. Such methodologies should be able to visualize and automate the  
19 generation of a set of production processes in accordance with the set of customized products in a  
20 family. In line with this general requirement, more specific requirements are identified as follows:  
21  
22  
23  
24

- 25  
26 • A mechanism should be designed to capture existing product and process data and their  
27 relationships, thereby anchoring process family planning to a common platform.
- 28  
29 • A formal representation should be constructed to model the dynamic behavior of process  
30 family planning by handling the involved complexities, thus shedding light on the underlying logic for  
31 planning processes families.
- 32  
33 • Computer support incorporating product and process family knowledge should be designed to  
34 facilitate process family planning. In other words, such support should be able to automatically  
35 generate process families for given product families.

36  
37 To satisfy such requirements, this study proposes a methodology by integrating PNs and KBSs, as  
38 shown in Figure 2. In the first step, the IP<sup>2</sup>S is constructed to organize data pertaining to a given  
39 product family and data relevant to the corresponding process family as a single structure. In this way,  
40 product and process family data and knowledge that are stored previously in different media, such as  
41 designer/engineer's memories, design documents, process cards, are captured and maintained in one  
42 unified unit, facilitating downstream product development activities as well. Moreover, since process  
43 data is mapped to product data, the IP<sup>2</sup>S entails a theoretically sound mechanism. Based on such  
44 mechanism, companies are able to plan production processes by considering the existing  
45 manufacturing resources and the common and/or similar routings to be adopted on shop floors.  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60



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

timed PNs model can not only capture the dynamics and reasoning of process family planning but provides readers with an easier intuitive understanding.

The last step attempts to support decision making by automatically generating process families for given product families. (Note, in line with the focus of this study: formulation of the IP<sup>2</sup>S and the corresponding PN model, we generally discuss this last step, which itself is an interesting topic in the field of decision support systems.) In view of the significance of KBSs in decision making support and the analogy between PNs and KBSs, this step proposes KBS development based on the net system model and the IP<sup>2</sup>S. Similar with most reported KBSs, the major functional modules in the KBS should include a user interface, a knowledge base, a database, an inference engine and a working memory. The user interface is to not only obtain input data from users but output suggested production processes. While the knowledge base is to organize all the knowledge involved in planning production processes for a product family, the database is to keep all data pertaining to the product family and the corresponding process family by following the relationships embedded in the IP<sup>2</sup>S. The inference engine controls the entire executive process of the KBS by applying rules in the knowledge base to input data to derive recommended production processes. Containing facts, such as the initial input data, the working memory changes in the course of planning production processes for given product family members. At last, the outputs of the KBS are a set of production processes corresponding to the set of product family members.

#### 4. INTEGRATED PRODUCT-PROCESS FAMILY STRUCTURE

Regardless of the many definitions in the literature, a consensus is that a product family is a set of related products assuming a common product structure and performing a basic function (Fan and Liu, 1999; De Lit, 2000; Simpson, 2004). By following this consensus, this study focuses on such product families that have common product structures. In addition to the common product structure, individual products in a family often possess similar items of same types (i.e., item families). Thus, in line with the common product structure and item families, all data pertaining to a product family can be organized as a generic product structure, as shown in Figure 3(a). Each node in the generic product structure is a generic item in the sense that it represents a class (or family) of similar item variants of



resources to be adopted on shop floors. In turn, the similar production processes can eliminate the unnecessary changeovers in production, thus contributing to production stability and efficiency.

## 5. NESTED COLORED TIMED PN MODEL OF PROCESS FAMILY PLANNING

To cope with the major issues in process family planning, a formalism of CTPNs (colored timed PNs) is developed based on the principles of CPNs, NPNs and TPNs. In accordance with the generic items in the IP<sup>2</sup>S, a number of CTPNs is specified to model the process families. A multilevel system of nested CTPNs, *SysN*, is further developed to model process family planning.

The elements essential to a PN are places, tokens, transitions, arcs and arc expressions describing pre/post conditions of transition firing. By attaching data (so called colors in CPNs) defining real world objects to tokens, the resulting colored tokens can capture the large variety of product items and process elements in a compact manner. In conjunction with the relationships among product items and process elements, these colored tokens can also model how constraints are satisfied when determining machines, operations and other process elements. Defining nets, which represent processes producing child items, as tokens residing in places of nets, which model processes producing product items, can address the granularity concern in process family planning. A unique reconfiguration mechanism, including inhibitor arcs, reconfigurable transitions and generic machine places, is defined to accommodate process variations while building one single and concise model.

### 5.1 Net definitions

*Definition 1:* A CTPN is defined as a 11-tuple:

$$CTPN = (P, T, A, h, \Sigma, \alpha, V, E^T, E, \tau, \mu), \text{ where}$$

$P = P^B \cup P^I \cup P^M \cup P^{GM}$  is a finite nonempty set of places with 4 disjoint subsets:  $P^B$  (buffer places),  $P^I$  (item places),  $P^M$  (machine places) and  $P^{GM}$  (generic machine places);

$T = T^L \cup T^R \cup T^T$ ,  $P \cap T = \emptyset$  is a finite nonempty set of transitions with 3 disjoint subsets:  $T^L$  (logical transitions),  $T^R$  (reconfigurable transitions) and  $T^T$  (timed transitions);

$A \subseteq P \times T \cup T \times P$  is a finite nonempty set of normal arcs;

$h \subseteq P^{GM} \times T^R$ ,  $h \cap A = \emptyset$  is a finite nonempty set of inhibitor arcs;

$\Sigma$  is a set of types or color sets;

$V = \{v_i\}_n \subseteq \Sigma$  is a set of variables;

$\alpha : P \mapsto \Sigma$  is a color assignment function that maps a place,  $p$ , to a color set,  $\alpha(p)$ ;

$\tau \in \mathfrak{R}^+ \cup 0$  is a set of non-negative real numbers representing time delays;

$E^T$  is a timed arc expression function that maps arcs to timed expressions such that

$$\text{Type}(E^T(t, p)) = \alpha(p) \wedge \text{Type}(\text{Var}(E^T(t, p))) \subseteq \Sigma \wedge \tau, \forall (t, p) \in T^L \times P^I;$$

$E$  is an untimed arc expression function that maps arcs other than  $T^L \times P^I$  to untimed expressions consisting of  $v_i \in V$  such that  $\text{Type}(v_i) = \alpha(p)$ ,  $\forall p \notin t \mid t \in T^T$ ; and

$\mu : P \mapsto \Sigma_{MS}$  is a marking function specifying the distribution of colored tokens in all places ( $\Sigma_{MS}$  is the family of all multisets over  $\Sigma$ ).

A CTPN is defined to model processes producing an item family and involves the families of immediate child items only. In other words, a CTPN deals with parent and child items at two adjacent levels of the generic product structure. Since such an item family can be the final product family, an assembly family or a part family, CTPNs can be assembly processes for final product and assembly families or manufacturing processes for part families.

Tokens are to model various product and process elements based on the host places; and tokens residing in same places belong to same types. More specifically, tokens residing in buffer places are defined to represent items ready to be processed or items that have been produced; tokens in item places model items that are being processed by machines (including tools, fixtures, jigs, etc.); tokens in machines places reflect idle machines; and tokens in generic machine places indicate machines that are selected from multiple alternatives. Since the CTPNs are specified taking into account the IP<sup>2</sup>S of the product family in consideration, the data attached to tokens is consistent with the data organized in the IP<sup>2</sup>S of the product family.

Logical transitions are defined to model the preconditions of operations (e.g., the availability of machines). Timed transitions are used to represent operations which take certain time durations. Reconfigurable transitions are introduced to model the selection of machines. The firing of

reconfigurable transitions and logical transitions is atomic and takes no time delay, whilst the firing of timed transitions incurs time delays which are equal to the cycle times of the operations represented. Inhibitor arcs are introduced to connect generic machine places to reconfigurable transitions. They prevent the firing of an enabled reconfigurable transition when there is a token in the associated generic machine place. In this way, only one machine is selected each time for given material items. Essentially, inhibitor arcs, reconfigurable transitions and generic machine places form the reconfiguration mechanism that accommodates machine selection and process variations.

While inhibitor arcs control machine selection, normal arcs describe process flows involving both items and machines. Variables in arc expressions are used to specify process flows. Each variable is defined to belong to a color type, thus having the corresponding set of color instances as values. Expressions attached to normal arcs are defined to specify pre/post conditions of transition firing. Both timed arc expressions and untimed arc expressions consist of variables, which belong to the color types of associated places. Evaluating timed arc expressions results in colored tokens representing input items, output items to be produced, machines and the corresponding operations to be adopted, cycle times to be incurred as well; evaluating untimed arc expressions specifies the preconditions of firing logical transitions (e.g., the presence of material items) and the post conditions of firing timed transitions (e.g., the machines that are released after completing operations). The marking function describes system states by specifying the distribution of colored tokens in all places; and the initial marking  $\mu_0$  shows the initial state in line with the given colored tokens.

*Definition 2:* A multilevel system of nested CTPNs ( $SysN$ ) is defined as a 5-tuple:

$$SysN = (SN, SP^B, \beta, A^R, \gamma), \text{ where}$$

$SN = \{CTPN_i\}_I$  is a set of colored timed PNs;

$SP^B = \{p_a\}_A, \forall p_a \in P^B$  is a set of buffer places;

$\beta : SN \mapsto SP^B$  is a place assignment function that maps  $SN$  to  $SP^B$  such that

$$\forall CTPN_i \neq CTPN_j \in SN, p \in SP^B \cap CTPN_j : \beta(CTPN_i) = p ;$$











1  
2  
3 While this study concerns the problem of planning process families for given product families and  
4 contributes to obtaining production stability, it does not address the integrated design of product and  
5 process families. As pointed out in (Jiao et al., 2007), most of the reported integrated product and  
6 process models focus on individual products rather than product families. Thus, the integrated design  
7 of product and process families may pave a potential avenue for future research. As with product  
8 platforms for product family design, a unified product-process platform may be developed to support  
9 product family design, production and the integration of design and production. Furthermore, this  
10 study is centered around the first step involved in process family planning: the generation of  
11 production processes. It does not address the evaluation of alternative production processes and the  
12 associated issues, such as computation complexity. In this regard, future research efforts might be put  
13 in production process evaluation together with the relevant issues. At last, in view of the fact that this  
14 study does not discuss in detail the issues relevant to the design of a KBS, future efforts might be put  
15 in designing such a system so that the efficiency of process family planning can be improved.  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31

## 32 REFERENCES

- 33  
34 Aldakhilallah, K.A. and Ramesh, R., 1998, An integrated framework for automated process  
35 planning: Design and analysis, *International Journal of Production Research*, 36(4), pp. 939-956.  
36  
37 Ben-Arieh, D., Kumar, R.R. and Tiwari, M.K., 2004, Analysis of assembly  
38 operations' difficulty using enhanced expert high-level colored fuzzy Petri net model, *Robotics and*  
39 *Computer-Integrated Manufacturing*, 20, pp. 385-403.  
40  
41 Choi, J.C., Kim, B.M. and Kim, C., 1999, An automated progressive process planning and die  
42 design and working system for blanking or piercing and bending of a sheet metal product,  
43 *International Journal of Advanced Manufacturing Technology*, 15, pp. 485-497.  
44  
45 De Lit, P., 2000, A comprehensive and integrated approach for the design of a product family  
46 and its assembly system, PhD Dissertation, CAD/CAM Department, University of Libre de Bruxelles,  
47 Brussels, Belgium.  
48  
49 Fan, I.S. and Liu, C.K., 1999, Product family and variants: Definition and models, in *Flexible*  
50 *Automation and Intelligent Manufacturing 1999* (Ashayeri, J., Sullivan, W.G. and Ahmad, M.M.,  
51 Eds). New York: Begell House, pp. 213-224.  
52  
53 Forza, C. and Salvador, F., 2002, Product configuration and inter-firm co-ordination: An  
54 innovative solution from a small manufacturing enterprise, *Computers in Industry*, 49(1), pp. 37-46.  
55  
56 Gupta, S. and Krishnan, V., 1998, Product family-based assembly sequence design  
57 methodology, *IIE Transactions*, 30, pp. 933-945.  
58  
59  
60

1  
2  
3 Haddock, J. and Hartshorn, T.A., 1989, A decision support system for specific machine  
4 selection, *Computers and Industrial Engineering*, 16(2), pp. 277-286.

5  
6 Harhalakis, G., Lin, C.P., Mark, L. and Muro-Medrano, P., 1994, Implementation of rule-based  
7 information systems for integrated manufacturing, *IEEE Transactions on Knowledge and Data*  
8 *Engineering*, 6(6), pp. 892-908.

9  
10 Harhalakis, G., Lin, C.P., Mark, L. and Muro-Medrano, P., 1995, Structured representation of  
11 rule-based specifications in CIM using updated Petri nets, *IEEE Transactions on Systems, Man, and*  
12 *Cybernetics*, 25(1), pp. 130-144.

13  
14 Hayes, R., Pisano, G., Upton, D. and Wheelwright, S., 2005, *Operations, Strategy, and*  
15 *Technology: Pursuing the Competitive Edge*. John Wiley & Sons, Inc.

16  
17 Huang, S.H., Liu, Q. and Musa, R., 2004, Tolerance-based process plan evaluation using  
18 Monte Carlo simulation, *International Journal of Production Research*, 42(23), pp. 4871-4891.

19  
20 Hunynen, J., 1992, Using artificial intelligence technologies in production management,  
21 *Computers in Industry*, 19, pp. 21-35.

22  
23 Jensen, K., 1997, *Colored Petri Nets: Basic Concepts, Analysis Methods and Practical Use*.  
24 Vol. 2, New York: Springer.

25  
26 Jiao, J., Zhang, L. and Pokharel, S., 2007, Process platform planning for variety coordination  
27 from design to production in mass customization manufacturing, *IEEE Transactions on Engineering*  
28 *Management*, 54(1), pp. 112-129.

29  
30 Jiao J., Zhang L., Zhang Y. and Pokharel S., 2008, Association rule mining for product and  
31 process variety mapping, *International Journal of Computer Integrated Manufacturing*, 21(2), pp.  
32 111-124.

33  
34 Kuipers, B.S., De Witte, M.C. and van der Zwaan, A.H., 2004, Design or development?  
35 Beyond the LP-STC debate; inputs from a Volvo truck case, *International Journal of Operations &*  
36 *Production Management*, 24(8), pp. 840-854.

37  
38 Kuttner, R. and Karjust, K., 2006, Coordination of complex tasks of engineering product and  
39 manufacturing process optimization, *Proceedings of the Estonian Academy of Sciences Engineering*,  
40 12(3-1), pp. 163-175.

41  
42 Li, W.D., Ong, S.K. and Nee, A.Y.C., 2004, Optimization of process plans using a constraint-  
43 based tabu search approach, *International Journal of Production Research*, 42(10), pp. 1955-1985.

44  
45 Liang, T., 1988, Expert systems as decision aids: Issues and strategies, *Journal of Information*  
46 *Systems*, 2(4), pp. 41-50.

47  
48 Lin, Z.C. and Chang, Y.C., 1997, Sash sheet metal development and the application of a back-  
49 propagation neural network in the calculation of the developed length, *Journal of Materials*  
50 *Processing Technology*, 69(1), pp. 95-105.

1  
2  
3 Lin, Z.C. and Deng, C.H., 2001, Application of Petri net in the planning of a shearing cut and  
4 bending progressive die workstation, *International Journal of Materials and Product Technology*,  
5 16(6-7), pp. 579-591.

6  
7  
8 Lomazova, I.A., 2000, Nested Petri nets: A formalism for specification and verification of  
9 multi-agent distributed systems, *Fundamenta Informaticae*, 43, pp.195-214.

10  
11 Martinez, M.T., Favrel, J. and Ghodous, P., 2000, Product family manufacturing plan  
12 generation and classification, *Concurrent Engineering: Research and Applications*, 8(1), pp.12–22.

13  
14 Mockler, R.J. and Dologite, D.G., 1992, *Knowledge-based System: An Introduction to Expert*  
15 *Systems*, New York: Macmillan Publishing.

16  
17  
18 Moore, K.E. and Gupta, S.M., 1996, Petri net models of flexible and automated manufacturing  
19 systems: A survey, *International Journal of Production Research*, 34, pp. 161-170.

20  
21 Muro-Medrano, P.R., Banares, J.A. and Villarroel, J.L., 1998, Knowledge representation-  
22 oriented nets for discrete event system applications, *IEEE Transactions on Systems, Man, and*  
23 *Cybernetics-Part A: Systems and Humans*, 28(2), pp. 183-198.

24  
25 Parsaye, K. and Chignell, M., 1988, *Expert Systems for Experts*. John Wiley & Sons, Inc.

26  
27 Peterson, J.L., 1981, *Petri Net Theory and the Modeling of Systems*. Prentice-Hall, Englewood  
28 Cliffs, N.J.

29  
30  
31 Pisano, G.P., 1997, *The Development Factory: Unlocking the Potential of Process Innovation*.  
32 Harvard Business School Press: Boston, MA.

33  
34 Ramachandani, C., 1974, Analysis of asynchronous concurrent systems by timed Petri nets.  
35 Technical Report MAC TR 120, MIT, Cambridge.

36  
37 Rao, M.P., Miller, D.M. and Lin, B., 2005, PET: An expert system for productivity analysis,  
38 *Expert Systems with Applications*, 29, pp. 300-309.

39  
40 Righini, G., 1990, FIRST: A Petri net-based system for simulation of complex distributed  
41 manufacturing systems, *Computer Integrated Manufacturing Systems*, 3(4), pp. 252-263.

42  
43 Righini, G., 1993, Modular Petri nets for simulation of flexible production systems,  
44 *International Journal of Production Research*, 31(10), pp. 2463-2477.

45  
46 Simpson, T.W., 2004, Product platform design and customization: Status and promise, *Artificial*  
47 *Intelligence for Engineering Design, Analysis and Manufacturing*, 18(1), pp. 3–20.

48  
49 Singh, R. and Sekhon, G.S., 2002, An intelligent system for optimal selection of dies and tools  
50 for sheet metal operations, *Proceedings of Mechanical Engineers*, Vol. 216, Part B: *Journal of*  
51 *Engineering Manufacture*, pp. 821-828.

52  
53 Thomas, J.P., 1992, A Petri net framework for representing mechanical assembly sequences,  
54 *Proceedings of 1992 IEEE/RJS International Conference on Intelligent Robots and Systems*, Raleigh,  
55 NC, pp. 2116-2121.

56  
57  
58 Tong, K.W. and Kwong, C.K., 2000, Process design of microchip encapsulation: A case based  
59 reasoning approach, *Journal of Electronics Manufacturing*, 10, pp. 127-138.

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

Tong, K.W., Kwong, C.K. and Yu, K.M., 2004, Intelligent process design system for the transfer molding of electronic packages, *International Journal of Production Research*, 42(10), pp. 1911-1931.

Tonshoff, H.K., Beckendorff, U. and Schaele, M., 1988, Some approaches to represent the interdependence of process planning and process control, *Proceedings of the 19th CIRP International Seminar on Manufacturing Systems*, State College, PA, pp. 165-170.

Tor, S.B., Britton, G.A. and Zhang, W.Y., 2005, A knowledge-based blackboard framework for stamping process planning in progressive die design, *International Journal of Advanced Manufacturing Technology*, 26, pp. 774-783.

Tu, Y.L., Chu, X.L. and Yang, W.Y., 2000, Computer-aided process planning in virtual one-of-a-kind production, *Computers in Industry*, 41, pp. 99-110.

Yan, H.S., Wang, N.S., Cui, X.Y. and Zhang, J.G., 1997, Modeling, scheduling and control of flexible manufacturing systems by extended high-level evaluation Petri nets, *IIE Transactions*, 29, pp. 147-158.

Yeung, T.S. and Yuen, M.M.F., 1996, Implementation study of intelligent system for IC transfer molding process, *Proceedings of SPIE – International Society for Optical Engineering*, 2644, pp. 471-476.

Yeung, T.S., Yuen, M.M.F. and Kwok, W.M., 1995, Development of initial machine setting program for IC transfer molding process, *Proceedings of the 3rd International Conference CIM*, 2, pp. 1445-1452.

Yuan, J.T. and Li, L.C., 2000, Integrating assembly and machining planning using graph-based representation models, *International Journal of Production Research*, 38(12), pp. 2619-2641.

Yuen, C.F., Wong, S.Y. and Venuvinod, P.K., 2003, Development of a generic computer-aided process planning support system, *Journal of Materials Processing Technology*, 139, pp. 394-401.

Zha, X.F., 2004, Planning for STEP-based electro-mechanical assemblies: An integrated approach, *International Journal of Computer Integrated Manufacturing*, 17(4), pp. 305-326.

Zha, X.F. and Du, H.J., 2001, Mechanical systems and assemblies modeling using knowledge-intensive Petri nets formalisms, *Artificial Intelligence for Engineering Design, analysis and Manufacturing*, 15, pp. 145-171.

Zha, X.F., Li, L.L. and Zhang, W.J., 2002, A knowledge Petri net model for flexible manufacturing systems and its application for design and verification of FMS controllers, *International Journal of Computer Integrated Manufacturing*, 15(3), pp. 242-264.

Zha, X.F., Lim, S.Y.E. and Fok, S.C., 1998a, Integrated knowledge-based assembly sequence planning, *International Journal of Advanced Manufacturing Technology*, 14, pp. 50-64.

Zha, X.F., Lim, S.Y.E. and Fok, S.C., 1998b, Integrated knowledge-based Petri net intelligent flexible assembly planning, *Journal of Intelligent Manufacturing*, 9, pp. 235-250.

1  
2  
3 Zha, X.F., Lim, S.Y.E. and Fok, S.C., 2003, A knowledge intensive multi-agent system for  
4 cooperative/collaborative assembly modeling and process planning, *Transactions of the SDPS*, 7(1),  
5 pp. 99-122.  
6  
7

8 Zhang, L., 2007, Process platform-based production configuration for mass customization,  
9 Ph.D. Dissertation, Nanyang Technological University, Singapore.  
10

11 Zhang, L. and Jiao, J., 2008, Modeling production configuration using nested colored object-  
12 oriented Petri nets with changeable structures, *Journal of Intelligent Manufacturing*, In press, 2008.  
13

14 Zhang, L. and Rodrigues, B., 2009, A tree unification approach to constructing generic  
15 processes, *IIE Transactions*, 41(10), pp. 916-929.  
16  
17

18 Zhang, W.X., 1989, Representation of assembly and automatic robot planning by Petri net,  
19 *IEEE Transactions on System, Man, Cybernetics*, 29(2), pp. 418-422.  
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

## A LIST OF FIGURES

Figure 1. Process family planning positioning

Figure 2. A systematic methodology for process family planning

Figure 3. Integrated product-process family structure

Figure 4. Generic structures of product family A

Figure 5. CTPN models of processes of three item families

Figure 6. The net system of process family planning for product family A

Figure 7. The generic product hierarchy of the truck family

Figure 8. The IP<sup>2</sup>S of the truck family

Figure 9. The CTPN model representing process family planning of the truck family

Figure 10. Examples of production rules for executing CTPN

Figure 11. The generated production process based on CTPN simulation

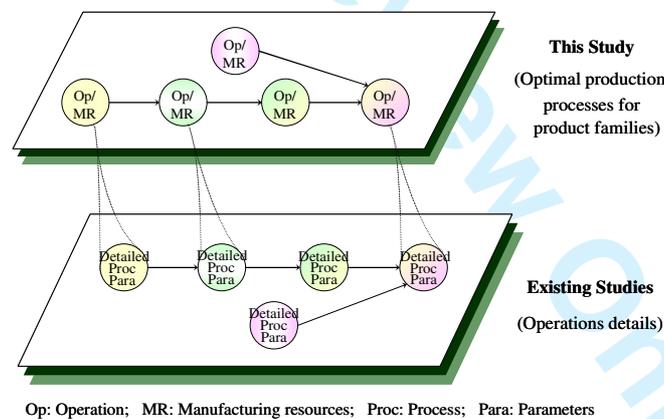


Figure 1. Process family planning positioning

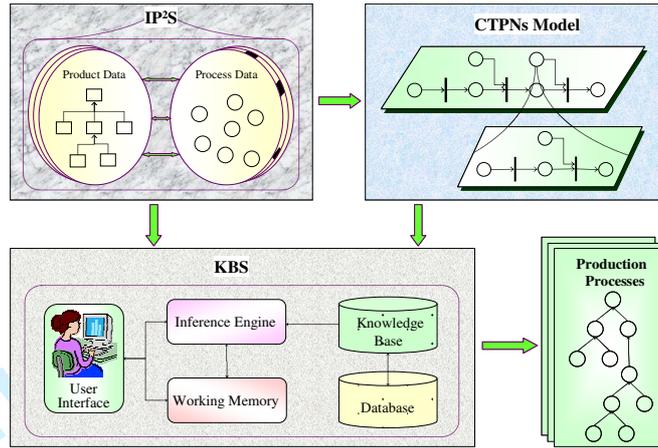


Figure 2. A systematic methodology for process family planning

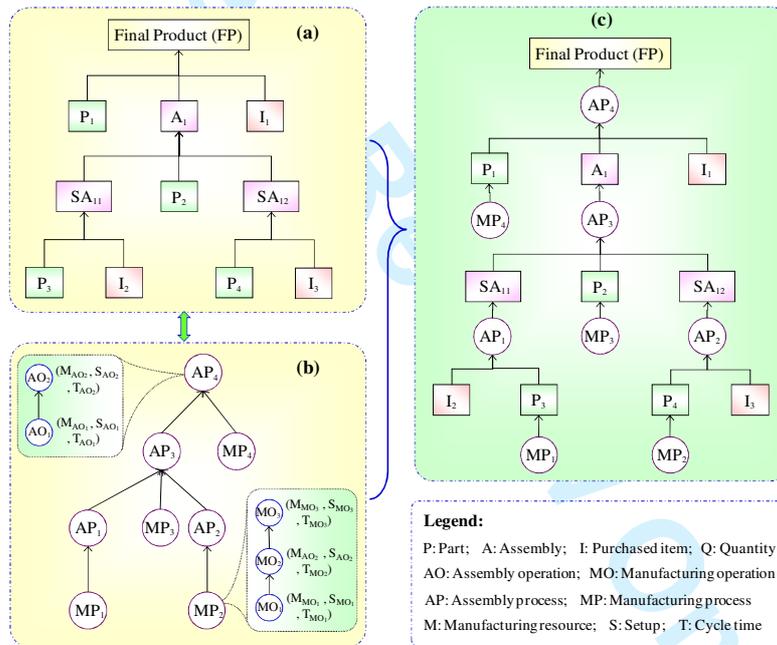


Figure 3. Integrated product-process family structure

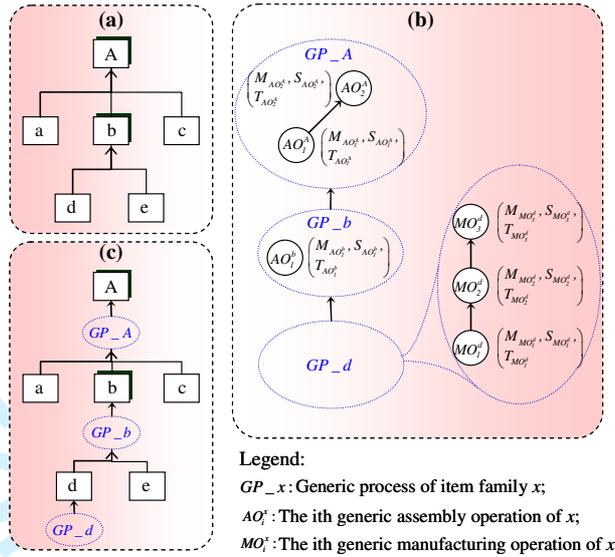


Figure 4. Generic structures of product family A

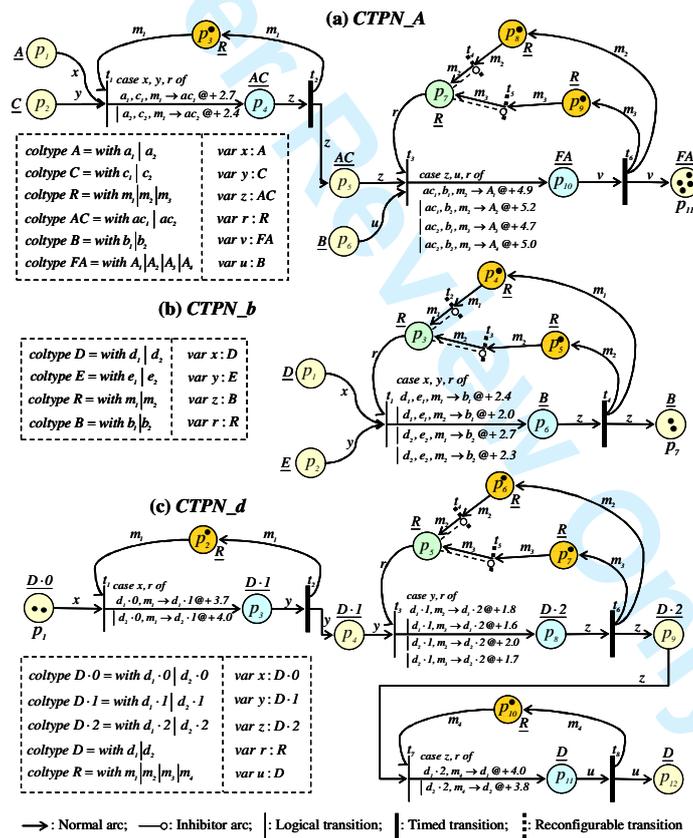


Figure 5. CTPN models of processes of three item families

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

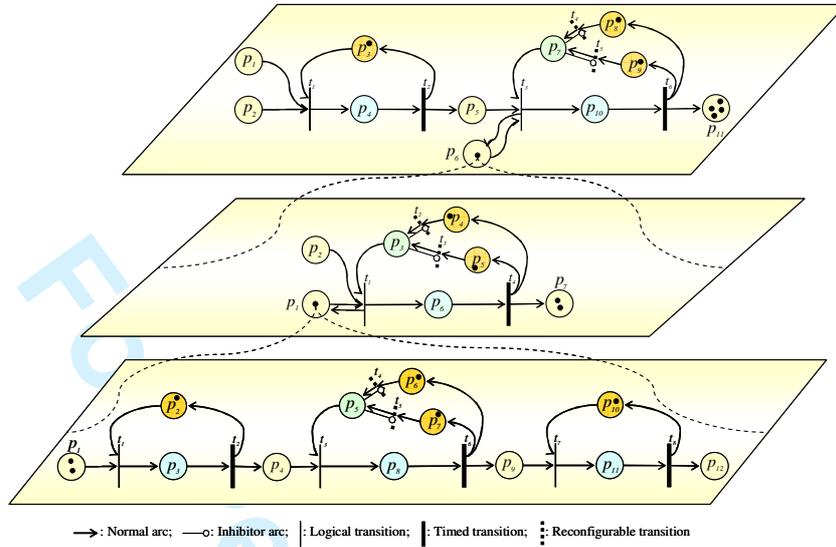


Figure 6. The net system of process family planning for product family A

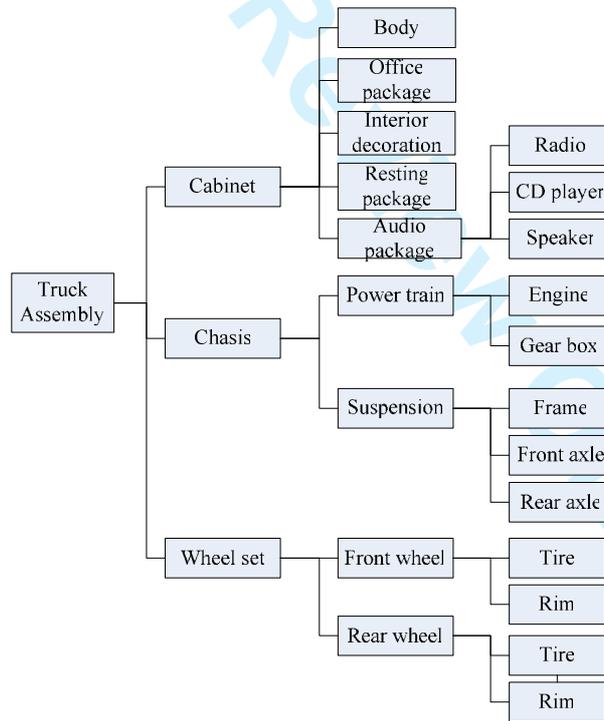


Figure 7. The generic product hierarchy of the truck family

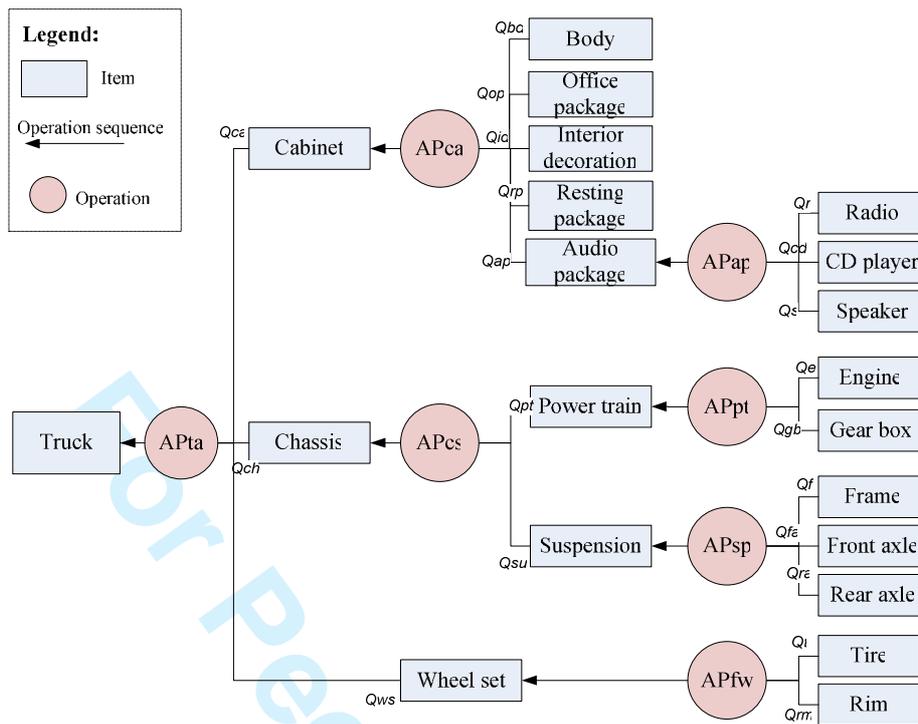


Figure 8. The IP<sup>2</sup>S of the truck family

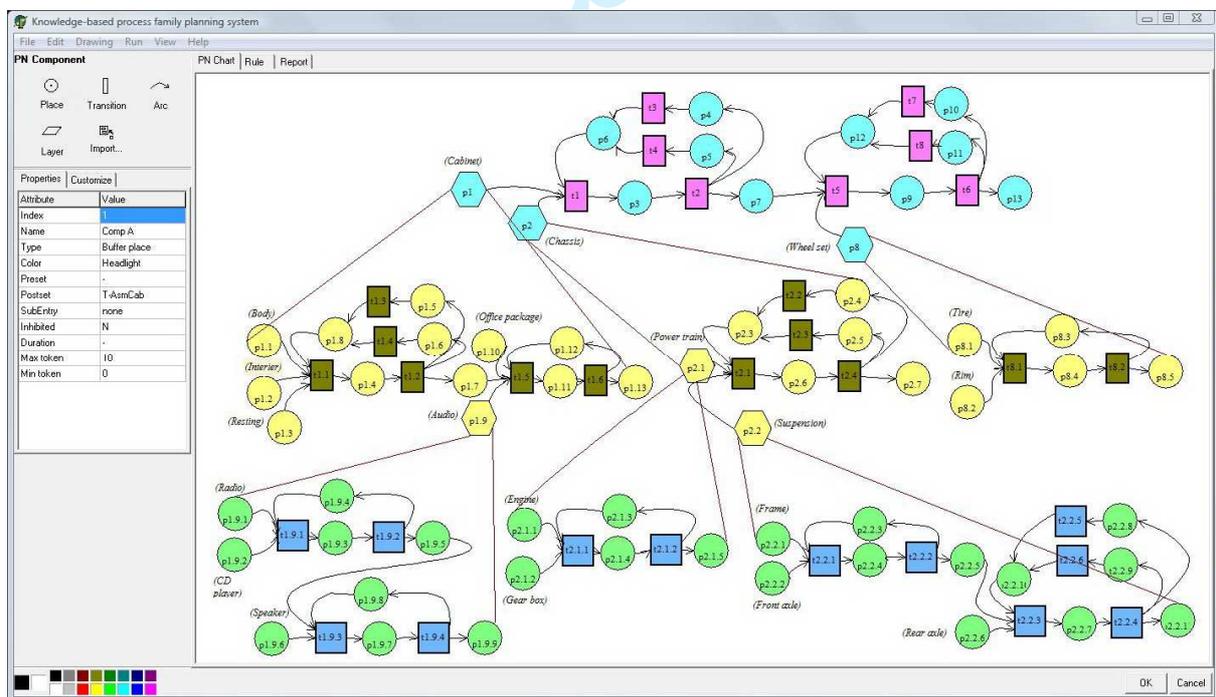


Figure 9. The CTPN model representing process family planning of the truck family

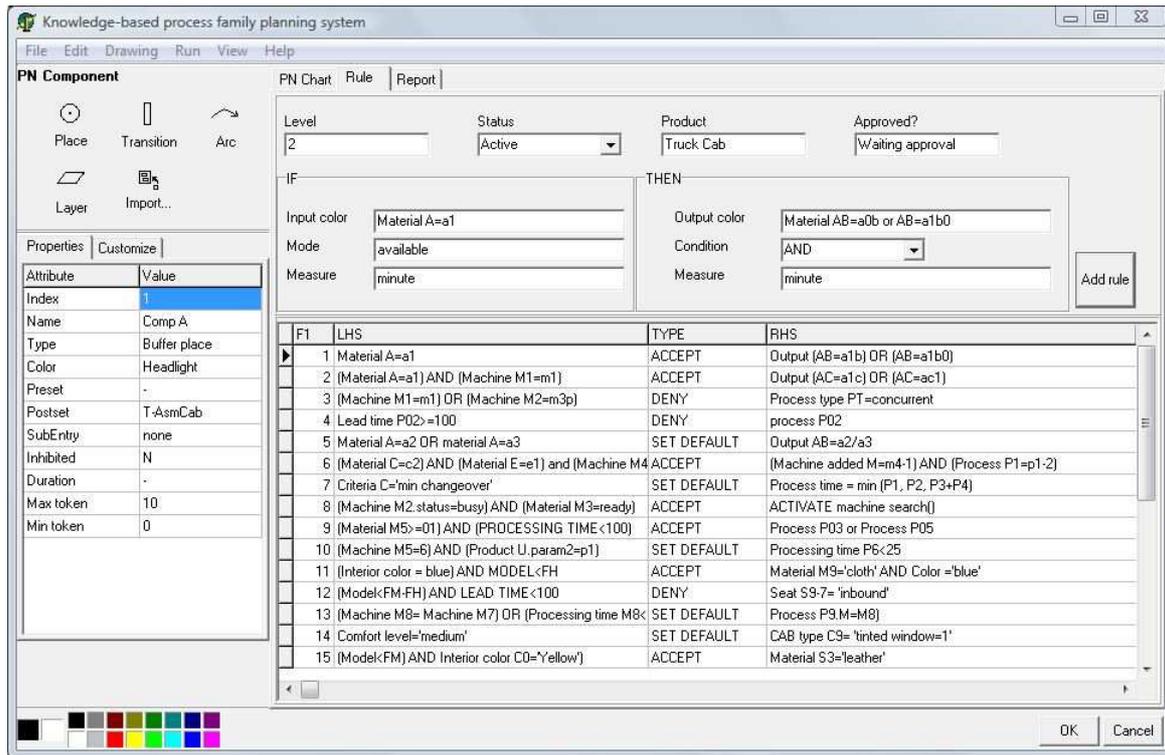


Figure 10. Examples of production rules for executing CTPN

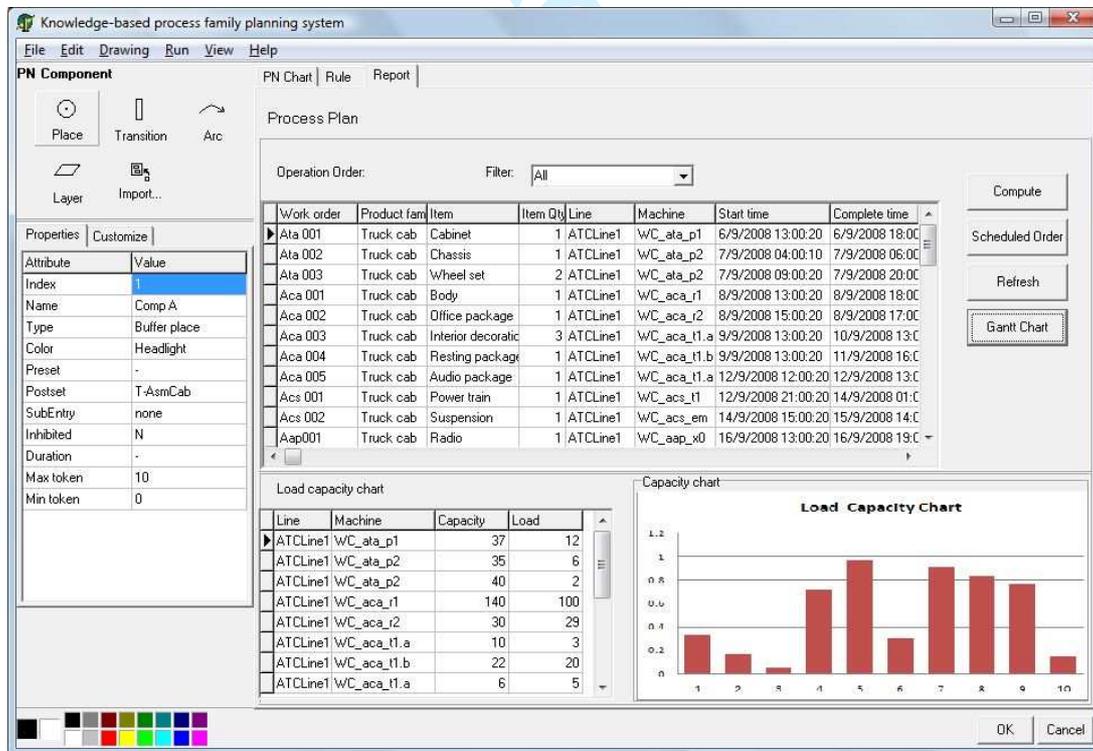


Figure 11. The generated production process based on CTPN simulation

## APPENDIX A

The 3 CTPNs in Figure 5 can be formally described as follows:

(1) *CTPN\_A*

$$P = P^B \cup P^I \cup P^M \cup P^{GM}$$

$$= \left\{ \{p_1, p_2, p_5, p_6, p_{11}\}, \{p_4, p_{10}\}, \{p_3, p_8, p_9\}, \{p_7\} \right\};$$

$$T = T^L \cup T^T \cup T^R$$

$$= \left\{ \{t_1, t_3\}, \{t_2, t_6\}, \{t_4, t_5\} \right\};$$

$$A = \left\{ \begin{array}{l} (p_1, t_1), (p_2, t_1), (p_3, t_1), (t_1, p_4), (p_4, t_2), (t_2, p_3), (t_2, p_5), (p_5, t_3), (p_6, t_3), \\ (p_7, t_3), (t_3, p_{10}), (p_{10}, t_6), (t_6, p_9), (t_6, p_8), (p_8, t_4), (p_9, t_5), (t_4, p_7), (t_5, p_7), \\ (t_6, p_{11}) \end{array} \right\};$$

$$h = \left\{ (p_7, t_4), (p_7, t_5) \right\};$$

$$\Sigma = \{A, C, AC, R, B, FA\};$$

$$V = \{x, y, z, u, v, r\};$$

$$E = \begin{cases} x & \text{if arc} = (p_1, t_1) \\ y & \text{if arc} = (p_2, t_1) \\ m_1 & \text{if arc} = (p_3, t_1); \\ \vdots & \\ v & \text{if arc} = \begin{cases} (p_{10}, t_6) \\ (t_6, p_{11}) \end{cases} \end{cases};$$

$$E^T = \begin{cases} \text{case } x, y, r \text{ of } \begin{cases} a_1, c_1, m_1 \rightarrow ac_1 @ + 2.7 \\ a_2, c_2, m_1 \rightarrow ac_2 @ + 2.4 \end{cases} & \text{if arc} = (p_3, t_1) \\ \text{case } z, u, r \text{ of } \begin{cases} ac_1, b_1, m_2 \rightarrow A_1 @ + 4.9 \\ ac_1, b_2, m_2 \rightarrow A_2 @ + 5.2 \\ ac_2, b_1, m_3 \rightarrow A_3 @ + 4.7 \\ ac_2, b_2, m_3 \rightarrow A_4 @ + 5.0 \end{cases} & \text{if arc} = (t_3, p_{10}) \end{cases};$$

$$\alpha(p) = \begin{cases} \{a_1, a_2\} & \text{if } p = p_1 \\ \{c_1, c_2\} & \text{if } p = p_2 \\ \vdots & \\ \{A_1, A_2, A_3, A_4\} & \text{if } p = p_{10} \\ \{A_1, A_2, A_3, A_4\} & \text{if } p = p_{11} \end{cases};$$

$$\tau = \{2.7, 2.4, 4.9, 5.2, 4.7, 5.0\}; \text{ and}$$

$$\mu_0 = \{0, 0, m_1, 0, 0, 0, 0, m_2, m_3, 0, (A_1, A_2, A_3, A_4)\}.$$

(2) *CTPN\_b*

$$P = P^B \cup P^I \cup P^M \cup P^{GM}$$

$$= \left\{ \{p_1, p_2, p_7\}, \{p_6\}, \{p_4, p_5\}, \{p_3\} \right\};$$

$$\begin{aligned}
T &= T^L \cup T^T \cup T^R \\
&= \{\{t_1\}, \{t_4\}, \{t_2, t_3\}\}; \\
A &= \left\{ (p_1, t_1), (p_2, t_1), (p_3, t_1), (t_1, p_6), (t_2, p_3), (t_3, p_3), (p_4, t_2), (p_5, t_3), (p_6, t_4), \right. \\
&\quad \left. (t_4, p_4), (t_4, p_5), (t_4, p_7) \right\}; \\
h &= \{(p_3, t_2), (p_3, t_3)\}; \\
\Sigma &= \{D, E, B, R\}; \\
V &= \{x, y, z, r\}; \\
E &= \begin{cases} x & \text{if arc} = (p_1, t_1) \\ y & \text{if arc} = (p_2, t_1) \\ r & \text{if arc} = (p_3, t_1); \\ \vdots & \\ z & \text{if arc} = \begin{cases} (p_6, t_4) \\ (t_4, p_7) \end{cases} \end{cases}; \\
E^T &= \text{case } z, u, r \text{ of } \begin{cases} d_1, e_1, m_1 \rightarrow b_1 @ + 2.4 \\ d_1, e_1, m_2 \rightarrow b_1 @ + 2.0 \\ d_2, e_2, m_1 \rightarrow b_2 @ + 2.7 \\ d_2, e_2, m_2 \rightarrow b_2 @ + 2.3 \end{cases} \text{ if arc} = (t_1, p_6); \\
\alpha(p) &= \begin{cases} \{d_1, d_2\} & \text{if } p = p_1 \\ \{e_1, e_2\} & \text{if } p = p_2 \\ \{m_1, m_2\} & \text{if } p = p_3 \\ \{m_1\} & \text{if } p = p_4; \\ \{m_2\} & \text{if } p = p_5 \\ \{b_1, b_2\} & \text{if } p = \begin{cases} p_6 \\ p_7 \end{cases} \end{cases}; \\
\tau &= \{2.4, 2.0, 2.7, 2.3\}; \text{ and} \\
\mu_0 &= \{0, 0, 0, m_1, m_2, 0, (b_1, b_2)\}. \\
(3) \text{ CTPN}_d \\
P &= P^B \cup P^I \cup P^M \cup P^{GM} \\
&= \{\{p_1, p_4, p_9, p_{12}\}, \{p_3, p_8, p_{11}\}, \{p_2, p_6, p_7, p_{10}\}, \{p_5\}\}; \\
T &= T^L \cup T^T \cup T^R \\
&= \{\{t_1, t_3, t_7\}, \{t_2, t_6, t_8\}, \{t_4, t_5\}\}; \\
A &= \left\{ (p_1, t_1), (p_2, t_1), (t_1, p_3), (p_3, t_2), (t_2, p_2), (t_2, p_4), (p_4, t_3), (p_5, t_3), (t_3, p_8), \right. \\
&\quad \left. (p_8, t_6), (t_6, p_6), (t_6, p_7), (p_6, t_4), (p_7, t_5), (t_4, p_5), (t_5, p_5), \dots, (t_8, p_{12}) \right\}; \\
h &= \{(p_5, t_4), (p_5, t_5)\}; \\
\Sigma &= \{D \cdot 0, D \cdot 1, D \cdot 2, D, R\}; \\
V &= \{x, y, z, r, u\};
\end{aligned}$$

$$E = \begin{cases} x & \text{if } arc = (p_1, t_1) \\ m_1 & \text{if } arc = (p_2, t_1) \\ y & \text{if } arc = (p_3, t_2) \\ \vdots & \\ u & \text{if } arc = \begin{cases} (p_{11}, t_8) \\ (t_8, p_{12}) \end{cases} \end{cases};$$

$$E^T = \begin{cases} \text{case } x, r \text{ of } \begin{cases} d_1 \cdot 0, m_1 \rightarrow d_1 \cdot 1 @ + 3.7 \\ d_2 \cdot 0, m_1 \rightarrow d_2 \cdot 1 @ + 4.0 \end{cases} & \text{if } arc = (t_1, p_3) \\ \text{case } y, r \text{ of } \begin{cases} d_1 \cdot 1, m_2 \rightarrow d_1 \cdot 2 @ + 1.8 \\ d_1 \cdot 1, m_3 \rightarrow d_1 \cdot 2 @ + 1.6 \\ d_2 \cdot 1, m_2 \rightarrow d_2 \cdot 2 @ + 2.0 \\ d_2 \cdot 1, m_3 \rightarrow d_2 \cdot 2 @ + 1.7 \end{cases} & \text{if } arc = (t_3, p_8); \\ \text{case } z, r \text{ of } \begin{cases} d_1 \cdot 2, m_4 \rightarrow d_1 @ + 4.0 \\ d_2 \cdot 2, m_4 \rightarrow d_2 @ + 3.8 \end{cases} & \text{if } arc = (t_7, p_{11}) \end{cases};$$

$$\alpha(p) = \begin{cases} \{d_1 \cdot 0, d_2 \cdot 0\} & \text{if } p = p_1 \\ \{m_1\} & \text{if } p = p_2 \\ \{d_1 \cdot 1, d_2 \cdot 1\} & \text{if } p = \begin{cases} p_3 \\ p_4 \end{cases}; \\ \vdots \\ \{m_4\} & \text{if } p = p_{10} \\ \{d_1, d_2\} & \text{if } p = \begin{cases} p_{11} \\ p_{12} \end{cases} \end{cases};$$

$$\tau = \{3.7, 4.0, 1.8, 1.6, 2.0, 1.7, 4.0, 3.8\}; \text{ and}$$

$$\mu_0 = \{\{d_1 \cdot 0, d_2 \cdot 0\}, m_1, 0, 0, 0, m_2, m_3, 0, 0, m_4, 0, 0\}.$$

## APPENDIX B

The net system in Figure 6 can be formally described as follows:

$$SysN = (SN, SP^B, \beta, A^R, \gamma)$$

$$SN = \{CTPN\_A, CTPN\_b, CTPN\_d\};$$

$$SP^B = \{p_6 @ CTPN\_A, p_1 @ CTPN\_b\};$$

$$\beta(CTPN) = \begin{cases} p_6 @ CTPN\_A & \text{if } CTPN = CTPN\_b \\ p_1 @ CTPN\_b & \text{if } CTPN = CTPN\_d \end{cases};$$

$$A^R = \{(t_3, p_6) @ CTPN\_A, (t_1, p_1) @ CTPN\_b\}; \text{ and}$$

$$\gamma(t) = \begin{cases} t_3 @ CTPN\_A & \text{if } t = t_4 @ CTPN\_b \\ t_1 @ CTPN\_b & \text{if } t = t_8 @ CTPN\_d \end{cases}.$$