2009

A Petri Net-based Approach to Reconfigurable Manufacturing Systems Modeling

Linda L. ZHANG
University of Groningen

Brian RODRIGUES
Singapore Management University, brianr@smu.edu.sg

Citation

ZHANG, Linda L. and RODRIGUES, Brian. A Petri Net-based Approach to Reconfigurable Manufacturing Systems Modeling. (2009). *Journal of Systemics, Cybernetics and Informatics*. 7, (1), 18-24.
Available at: https://ink.library.smu.edu.sg/lkcsb_research/3028

A Petri Net-based Approach to Reconfigurable Manufacturing Systems Modeling

Linda L. ZHANG
Department of Operations, University of Groningen
Groningen, 9747 AD, The Netherlands

and

Brian RODRIGUES
Lee Kong Chain School of Business, Singapore Management University
50 Stamford Road, Singapore

ABSTRACT

Reconfigurable manufacturing systems (RMSs) have been used to provide manufacturing companies with the required capacities and capabilities, when needed. Recognizing (1) the importance of dynamic modeling and visualization in decision making support in RMSs and (2) the limitations of the existing studies, we model RMSs based on Petri net (PN) techniques with focus on the process of reconfiguring system elements while considering constraints and system performance. In response to the modeling difficulties identified, a new formalism of colored timed PNs is introduced. In conjunction with colored tokens and timing in colored PNs and timed PNs, we further define a reconfiguration mechanism to meet the modeling difficulties. A case study of an electronics product is reported as an application of the proposed colored timed PNs to RMS modeling.

Keywords: Reconfigurable Manufacturing Systems, Petri Nets, Reconfiguration Mechanism.

1. INTRODUCTION

Recently, reconfigurable manufacturing systems (RMSs) have been acknowledged as a promising means which can assist companies quickly produce diverse individualized products at low costs [1]. A major concern in the use of RMSs is the quick reconfiguration of existing system elements, such as machines, tools, fixtures and setups, to provide changing production requirements. Existing studies have delivered a number of analytical models [2, 3, 4, 5]. On one hand, such models provide certain insight in RMS planning, design and operations; on the other hand, the complexity involved in the model formulation tends to limit understanding. Moreover, the implicit underpinning assumptions, which often contradict the counterparts in the real world, render model implementation difficult.

Identifying the importance of dynamic modeling and visualization in decision making support in RMSs, and in view of the limitations of the existing research, we propose to graphically model RMSs with focus on the process of reconfiguring system elements for given products while considering constraints and system performance. The attempt is to assist companies to make decisions in reconfiguring manufacturing resources to fulfill fast changing production requirements. Along with fundamental issues in RMSs, we first highlight the difficulties in modeling RMSs as follows:

Variety handling. The large number of individualized products in RMSs inevitably leads to a high variety of material items, be they raw materials, parts, WIP (work in process), or assemblies. Since their fulfillment is the central focus of RMSs, it is essential to capture the high variety of material items and end-products in system models. Furthermore, in spite of the inclusion of high product variety, a compact and representative model should be built in order to facilitate users' understanding, consistent interpretation and communication. This underscores the importance of handling high product variety in building system models.

Process variation accommodation. Product variety is associated with diverse design specifications (i.e., design parameters along with specific value instances). In turn, design changes lead to many changeovers in processes of producing material items and end-products. Such changeovers are reflected as variations in machines, operations and operations precedence. In order to provide companies with decision support (e.g., selecting machines), system models should be able to capture and reflect these variations.

Machine selection. In RMSs, a number of processes are feasible to produce one end-product. Such processes relate to different configurations of different and/or same machines. In practice, only one process is adopted to produce a product. It is common that different machines are able to perform operations on same material items for same jobs, and, in most cases, incur different cycle times. Similarly, only one machine is used to process items at one time. In accordance with various products to be produced in same time periods using identical resources, proper machines and processes must be selected from multiple alternatives. This selection should contribute to the improvement of certain system performance attributes (e.g., throughput, machine utilization, quality). Hence, system models should facilitate decision making in selecting machines and processes.

Constraint satisfaction. In RMSs, many restrictions or constraints can be observed. These constraints are inherent in the selection of machines and operations. They are, in fact, associated with items' specific design, machine capabilities, machine and item availabilities. For example, if a machine can perform operations on alloy steel only, it would be inappropriate to use it to execute operations on aluminum. It is fundamental therefore, to deal with these constraints in modeling RMSs in order to build viable models.

Due to executability and graphical representation, Petri nets (PNs) have been well recognized as a powerful modeling, simulation and evaluation tool for complex flows and processes [6]. Many extensions have been made to PNs to enhance their modeling power. Among these, colored Petri nets (CPN) [7] and timed Petri nets (TPN) [8] are of particular interest in this study. CPNs are able to provide a concise, flexible and manageable representation of large manufacturing systems by attaching a variety of colors to tokens. By including timing, TPNs can capture physical behavior of systems by assuming specific durations for various activities.

To achieve this, we apply PN techniques to model RMSs, and to cope with the modeling difficulties, we develop a new formalism of colored time PNs (CTPNs). The basic concepts of CPNs and TPNs are adopted and further extended to define elements in the formalism. Variety handling is accomplished by attaching specific data pertaining to various objects to tokens, resulting colored tokens. A mechanism including reconfigurable transitions, inhibitor arcs and a type of special places are defined to accommodate process changeovers. In conjunction with colored tokens, timing is introduced to address machine selection and constraint satisfaction. We also report a case study of an electronics product to demonstrate the application of CTPNs developed to RMS modeling.

2. MODELING FORMALISM OF CTPNS

In CPNs, colors essentially are specific data values describing objects represented by tokens. Each colored token is uniquely defined by a color. For analyzing the performance of a system model, time delays are introduced into PNs, resulting in TPN models. A time delay is a period of time, before the elapse of which a token after its arrival (atomic arrival) in a place cannot be used by the output transitions (i.e., it remains unavailable), and after the elapse of which the token becomes available and can be used to fire transitions.

In RMSs, multiple machines are able to carry out different operations on same material items for same jobs, and, in most cases, incur different cycle times. To capture and model these characteristics, a type of special places is defined to represent the class concepts of machines that can carry out same jobs. Along with machine class concept places, inhibitor arcs are introduced to keep more than one machine from accessing same material items at one time. To cope with the difficulties in modeling diverse cycle times associated with multiple machines and same jobs, arc expression functions are introduced. Furthermore, in response to limitations of associating time delays with places and transitions [9], we define time delays in arc expressions. Thus CTPNs formalism is able to capture and model different cycle times associated with same machines but with different material items.

Time delays can be obtained from a process platform of a process family in relation to a product family [10]. Compared with randomly generating time delays, determining time delays based on a process platform is advantageous in that the obtained time delays are more close to their counterparts in the real system. Fig. 1 shows the graphical formalism of CTPNs.

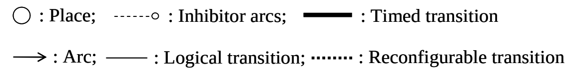

Figure 1. Graphical formalism of colored timed Petri nets

Definition 1: A Colored Timed Petri Net is a tuple $CTPN=(P,T,\Sigma,C,E,h,d,M_0)$, where

(i) $P, P=P^R\cup P^O\cup P^C$ is a set of places with three disjoint finite non-empty subsets. A $p\in P^R$ denotes either a buffer or a machine; a $p\in P^O$ indicates that a machine is working on material item(s); and a $p\in P^C$ represents a machine class concept;

(ii) $T, P\cap T=\phi$ is a set of transitions $T=T^L\cup T^T\cup T^R$, where $T^L/T^T/T^R$ are three disjoint finite nonempty subsets of logical/timed/reconfigurable transitions, respectively.

Logical transitions are introduced to capture the logic of system running. Their firing indicates the satisfaction of preconditions of operations. Reconfigurable transitions are defined to model situations where multiple machines can perform identical jobs and only one is used eventually. Their firing leads to the reconfiguration of proper machines. Timed transitions are to represent operations. Their firing takes a certain time duration. Logical and reconfigurable transitions are untimed. Their firing is atomic, with 0 time delay;

(iii) Σ is a finite nonempty set of color sets or token types, each of which includes a set of individual colors;

(iv) C is a color function that maps a place, p, to a set of colors, $C(p)$: $C(p)=\{o(c_{pi})\,c_{pi}\}_I$, where $o(c_{pi})$ is the occurrence multiplicity of color c_{pi};

(v) h, $h\subseteq P^C\times T^R$ is a set of inhibitor arcs that connect machine class concept places to reconfigurable transitions only, where $h(p,t)=1, \forall p\in P^C, t\in T^R$ indicates that there is a token in the machine class concept place and the associated reconfigurable transition is disabled and cannot fire. When $h(p,t)=0$, no token is in the machine class concept place and the associated reconfigurable transition can fire if it is enabled;

(vi) $d\in\Re^+$ is a set of positive real numbers for time delays of operations;

(vii) E^T is a timed arc expression function that maps an arc, $t\times p, \forall p\in P^O, t\in T^L$, to a timed arc expression: $E^T: T^L\times P^O\mapsto \vee\left(\wedge o(c_{p_m j})\,c_{p_m j}\rightarrow o(c_{ps})\,c_{ps}\,@+d\right)$, $\forall p_m\in{}^\bullet t$, $c_{p_m j}\in C(p_m), c_{ps}\in C(p)$ where $\vee$ represents Exclusive OR (XOR); $\wedge$ AND; and $\rightarrow$ "if-then"; and d a time delay.

A timed arc expression is a set of antecedent-consequent statements with XOR relationships. Each antecedent contains a set of colored tokens with AND relationships. The colored tokens correspond to these residing in input places of the logical transition. The occurrence of each such colored tokens may not be 1. By default, the occurrence of 1 is omitted. The consequent is the colored token to be generated in the working machine place along with the time delay. Conforming to the common practice, the occurrence of such tokens is 1 and thus being omitted.

E^U is an untimed arc expression function that maps an arc, other than $(t,p), \forall p\in P^O, t\in T^L$, to an arc expression without

time elements: $E^U : \neg T^L \times \neg P^O \mapsto \vee o(c_{pt})\, c_{pt}, \forall c_{pt} \in C(p)$, where $\vee$ represents XOR. Untimed arc expressions are defined to specify (1) input tokens for firing any transitions; and (2) output tokens after firing timed and reconfigurable transitions.
(iiiv) M is the marking function and M_0 is the initial marking.

$M = (\xi, \rho, \tau)$ is a combination of three functions: $\xi : P \mapsto \{o(c_{pi})\, c_{pi}\} \cup 0, \forall c_{pi} = C(p)$ is a marking function of available tokens; $\rho : P \mapsto \{o(c_{pi})\, c_{pi}\} \cup 0, \forall c_{pi} = C(p)$ is a marking function of unavailable tokens; τ is the remaining-unavailable-time function that assigns positive real values to a number of local clocks that measure the remaining time for each unavailable token, if any, in a place. If more than one unavailable token with a same color arrives in a place at different model times, τ assigns to these different remaining times according to the time delays in their corresponding arc expressions and the model time when they arrive in the place.

A transition t is enabled in a marking and can fire iff the following rules hold:
(1) Each $p, \forall p \in {}^{\bullet}t$ is marked with a "sufficient" number of colored tokens indicated by the expression on arc (p,t); and
(2) The firing of t does not violate the upper bound on any $p, \forall p \in t^{\bullet}$.

3. MODELING RMSS BASED ON CTPNS

Considering the involved high product variety, machines, diverse operations along with many cycle time instances, we approach the modeling of RMSs from system elements as follows.

Material Items

The introduction of colored tokens in the formalism allows modeling of high product variety while building compact models. Each token represents a specific item. They differ from one another in attribute values that define them.

As shown in Fig. 2(a), place p_1 represents a raw material buffer. The token $a \cdot 1$ in it denotes the raw material of part, a, to be produced. The data that specify the token include: part name (a), the state (1) indicating it is at the status of raw material, type of material (PVC), possible machines (m_1), and others. While the token in p_1 indicates that the raw material is ready to be processed, the white token in p_3 in Fig. 2(b) denotes another status of the raw material: being processed by the machine represented by p_2. Since the occurrences of the tokens in Fig. 2 are 1, by default they are omitted.

Manufacturing Resources

Machines take two statuses in a system model: idle and busy. If a machine is idle and available for the next operation, the corresponding place in the system model contains a token. As shown in Fig. 2(a), at the current system state, one machine represented by p_2 is available as there is a token in it. If a machine is working on material item(s), there would be a token in the place representing "machine processing items". Fig. 2(b) shows a busy machine represented by the white token in p_3.

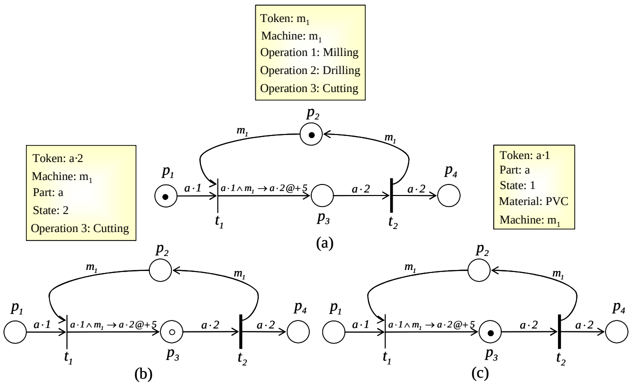

Figure 2. Modeling RMSs using CTPNs-based formalism

Cycle Times

With an attempt to capture different cycle times in relation to same jobs and different machines, time delays representing cycle times are attached to timed arc expressions, as shown in Fig. 2. The expression in Fig. 2(a) indicates that it takes 5 time units for machine, m_1, to complete the cutting operation on raw material, $a \cdot 1$. During 5 units of time after firing t_1, the token, $a \cdot 2$, created in p_3 is unavailable and represented by a white dot, as shown in Fig. 2(b). At the moment of 5 time units, the operation is completed and the token becomes available, as shown in Fig. 2(c). Accordingly, t_2 representing the cutting operation is enabled and can fire.

Operations

Before the occurrence of any operation, the input material items and machine to be used must present. During the operation, both material items and machines are not available for other purposes. After a certain time duration equal to the cycle time, the operation completes. Upon operation completion, input material items have been consumed and a parent item has been generated; the machine is released and waiting for the next task. To capture this characteristics, in this research an operation is modeled by several places representing buffers, machines and machine working on items, as shown in Fig. 3. The buffer places, p_1 and p_4, contain tokens, $a \cdot 3$ (representing the input material item) and $a \cdot 4$ (denoting the parent item), respectively. The machine place, p_2, shows the availability of the machine, m. Along with other relevant places and the residing tokens, p_3 indicates the operation has not started yet in Fig. 3(a); the operation is ongoing in Fig. 3(b); and the operation has been completed in Fig. 3(c).

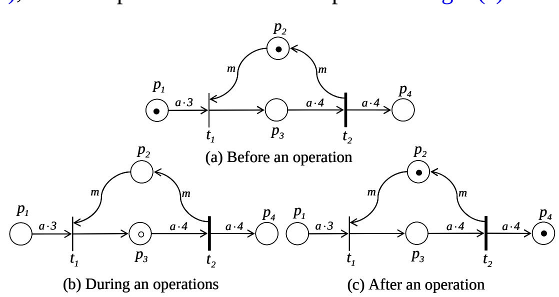

Figure 3. Modeling operations in RMSs

In RMSs, according to relationships among them and machines that perform them, operations can be classified into the following types.

Operations with individual machines: In practice, input material items traverse a series of operations performed by different machines used in producing end-products. The starting of following operations depends on the completion of previous ones and the availability of machines to be used. Fig. 4 shows an example of two sequential operations along with individual machines. Since the operation is ongoing, as indicated by the white token in p_3, the token representing the output parent item is not available in the WIP buffer p_4. As a result, t_3 is not enabled and cannot fire.

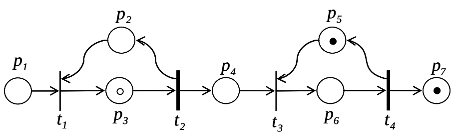

Figure 4. Sequential operations with individual machines

When a parent item is formed by more than one child item, operations required for producing the child items are often simultaneously performed by different machines. In some situations, such concurrent operations are vital for activity synchronization. Fig. 5 shows an example of 2 parallel operations with individual machines. The operation performed by the machine (represented by p_2) has been completed, as indicated by tokens in p_2 and p_4 (a WIP buffer). Since the operation performed by the other machine (represented by p_7) is ongoing, as indicated by the white dot in p_6, the token representing the corresponding output item has not been created in the WIP buffer place p_8. As a result, t_5 is not enabled. Upon the completion of the operation performed by p_7, t_5 fires with the presence of three tokens in p_4, p_8 and p_9, respectively.

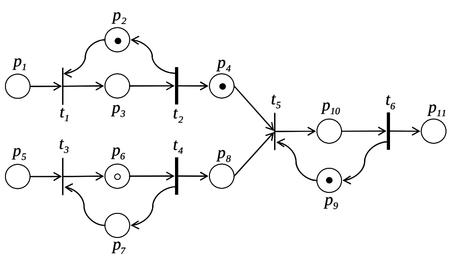

Figure 5. Parallel operations with individual machines

Operations with shared machines: Fig. 6 shows the situations, where operations are required to be performed by common machines. In Fig. 6(a), along with others, two operations, represented by t_2 and t_{i+1}, are for producing a same parent item, represented by the token in p_{j+1}. Since both t_2 and t_{i+1} require p_4 (the shared machine), a conflict may occur if there is a token in it. To solve such conflicts, we adopt the common approach proposed by most researchers: assign priorities to transitions. Different priority numbers (1, 2, …, n) are assigned to transitions, with one being the highest priority and n being the lowest priority.

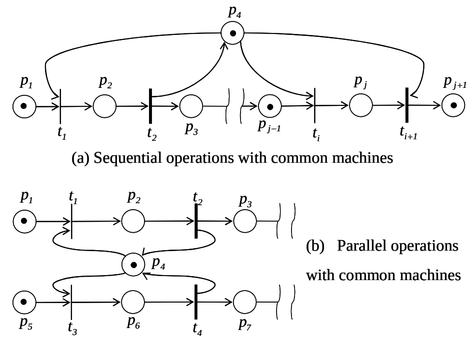

Figure 6. Operations with shared machines

For example, in Fig. 6(a), since t_{i+1} depends on t_2, the priority number of t_1 will be 1 and that of t_i will be 2. In Fig. 6(b), two operations represented by t_2 and t_4 are associated with two different output items, which are two sibling items under a parent item. Similarly, priorities are assigned to the corresponding logical transitions: t_1 and t_3. In this situation, the assignment can be made according to cycle times of the represented operations (t_2 and t_4 in this case).

Operations with alternative machines: Fig. 7(a) describes a general case in which an operation can be performed by different machines. Both machines, m_1 (represented by p_5) and m_2 (represented by p_6), can work on a same item (token $a \cdot 1$ in p_1). It takes m_1 and m_2 10 and 14 time units to complete their operations, respectively. To ensure that only one machine performs the operation, p_4 is incorporated to represent the class concept of the two machines; and thus both m_1 and m_2 are allowed to reside in p_4. The inhibitor arcs (the two dashed lines from p_4 to t_3 and to t_5) limits the number of tokens to reside in p_4 to 1 each time. Essentially, the two reconfigurable transitions (t_3 and t_5), the two inhibitor arcs and the machine class concept place form the reconfiguration mechanism. Along with the preferred scheduling rules, the mechanism controls the selection, and further reconfiguration, of a proper machine to perform the operation.

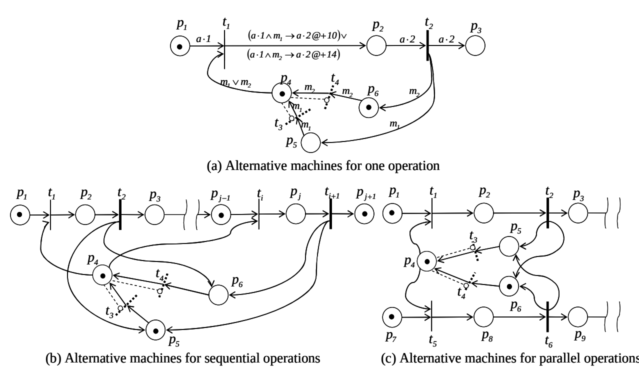

Figure 7. Operations with alternative machines

Figs 7(b) and 7(c) describe two more complicated situations, where multiple alternative machines are shared by more than one operation. When there is a token in p_4 in both models, conflicts may occur. Similarly, priority numbers are assigned to the competing logical transitions. In Fig. 7(b), priority numbers are assigned to t_1 and t_i, with a higher number to t_1 and a lower number to t_i. In Fig. 7(c), priority numbers are assigned to t_1 and t_5. The priority assignment in this condition can be determined by referring to the average cycle times associated with the two machines.

4. CASE STUDY

The proposed CTPNs formalism have been tested in a company that manufactures a high variety of customized vibration motors for mobile phones. Based on design similarities, the company has classified the motors into several families.

Model Construction

Among these facilies, modeling the RMS of one motor family is described. The motor family involves three major assemblies: frameassy, bracketassy and armartureassy, as shown in Fig. 8. Each of the three assemblies are formed by several manufactured parts and/or purchased components.

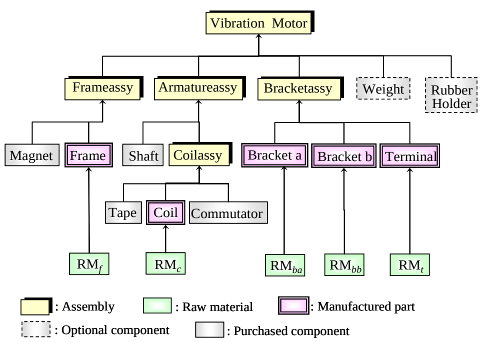

Figure 8. The common product structure of the motor family

Table 1 shows the machines, the associated operations and the output parts/WIP/assemblies. In spite of the variations in production processes of motor variants, a generic routing underpinning the process platform for manufacturing the motor family has been identified. Processes of individual motor variants differ from one another in involved machines, operations, cycle times and operations sequences.

Table 1. Machines, operations and the corresponding output items

Machines (MCs)	Operations	Output Parts/WIP/Assemblies
Multifunctional MC	Cutting	Terminal
	Winding	Coil
Injection MC	Fabrication	Bracket a
	Fabrication	Bracket b
Stamping MC	Fabrication	Frame
Workbench	Assembly	Coilassy
Inserting MC	Assembly	Armatureassy
Fusing MC		Abassy (aassy+bassy)
Pressing MC	Assembly	Frameassy
		Bracketassy
Caulking MC	Assembly	Mainbody (abassy+fassy)
		Vibration motor

By referring to the generic routing of the motor family, the system model has been constructed, as shown in Fig. 9. Table 2 shows the places and the represented system elements. Due to the space issue, the colored tokens are provided in the figure rather than in the table and the table is truncated.

Colored tokens residing in buffer places are defined based on the corresponding items in each family. For example, in p_1, $ba_1 \cdot 1$, $ba_2 \cdot 1$ and $ba_3 \cdot 1$, are defined to represent the raw materials of 3 bracket a variants ba_1, ba_2 and ba_3; $bb_1 \cdot 1$, $bb_2 \cdot 1$ and $bb_3 \cdot 1$ the raw materials of 3 bracket b variants bb_1, bb_2 and bb_3; and $tl_1 \cdot 1$, $tl_2 \cdot 1$ and $tl_3 \cdot 1$ the raw materials of three terminal variants tl_1, tl_2 and tl_3. Tokens in machine places (e.g., p_3) are specified according to machine names, machine capabilities, types of materials that the machine can work on, tools, fixtures and setups in relation to the operations that machines can perform. Tokens in places representing “machine working on material items” are defined based on the specific attribute data of output parts/WIP/assemblies. For example, the tokens in p_2 are defined using the specific data describing the output coil variants: c_1, c_2 and c_3.

The timed and untimed arc expressions are defined by taking into account constraints associated with machine capabilities and the company’s past production practice. Time delays in timed arc expressions are determined according to cycle times involved in the process platform of the motor family.

Table 2. Places and represented system elements

Places	System Elements	Places	System Elements
p_1	Raw material buffer for bracket a & b, terminal, coil, and frame	p_{18}	Pressing machine processing frameassy
p_2	Multifunctional mach processing coil raw materials	p_{19}	WIP buffer for coilassy
p_3	Multifunctional machine	p_{20}	WIP buffer for bassy
…	…	…	…
p_{16}	Pressing machine processing bassy	p_{33}	Caulking machine processing motors
p_{17}	Pressing machine	p_{34}	End-product buffer for motors

For instance, $(c_1 \cdot 1 \wedge w \rightarrow c_1 @{+}2) \vee (c_2 \cdot 1 \wedge w \rightarrow c_2 @{+}1.5) \vee (c_3 \cdot 1 \wedge w \rightarrow c_3 @{+}2.4)$, on arc, (t_1, p_2), specifies that w (the multifunctional machine) can work on the raw materials of the three coil variants; and it takes 2 hours, 1.5 hours and 2.4 hours to complete the relevant operations. With the presence of colored tokens $c_1 \cdot 1$ and w, t_1 fires immediately. However, t_6 will fire 2 hours later after the firing of t_1. Untimed arc expressions are defined to specify the input and output of transitions. For example, $tl_1 \vee tl_2 \vee tl_3$ on the output arc, (t_7, p_{11}), of t_7 shows the three possible output terminal variants: tl_1, tl_2 and tl_3.

Both p_{24} and p_{25} can perform the corresponding assembly operations to form aassy and abassy. To accommodate the reconfiguration, p_{23}, t_{18} and t_{19}, (p_{23}, t_{18}) and (p_{23}, t_{19}) are defined. The determination of machines is based on time delays in timed arc expressions and preferred schedule policies.

System Analysis

In [7], the author introduced several methods to verify models with respect to dynamic properties. Among these, P-invariant analysis is of particular interest to most researchers due to its

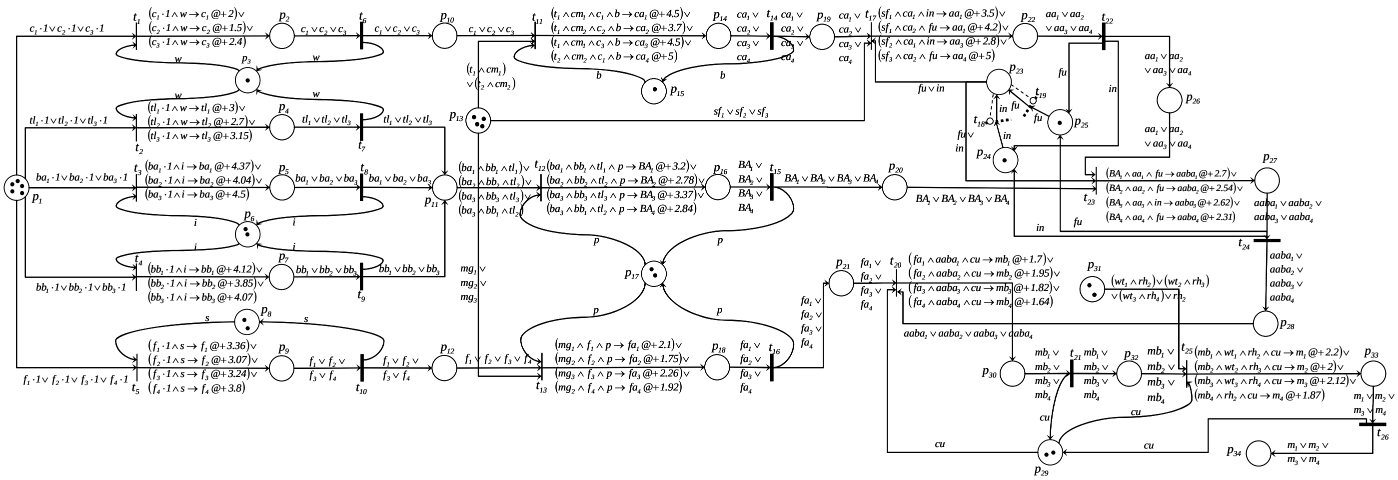

Figure 9. The CTPN model of the RMS

easy-understandability and implementation. Thus, in this research, we adopt P-invariant analysis. Several P-invariants can be identified in the model in Fig. 9. The total number of busy machines and idle machines gives a P-invariant. In other words, in any system states, the total number of tokens appearing in specific machine places, machine class concept places and machine working on material item places is always the same. Another P-invariant relates to material items in buffers and items being processed by machines. This P-invariant is obtained through mapping items being processed to the corresponding raw material items.

Application Results

The production performance considered in the application case is makespan. An optimal firing sequence (with respect to the minimum accumulated processing time) of the transitions in the system model in Fig. 9 results the determination of proper machines. Meanwhile, it provides the schedule of machines for producing products while leading to nearly minimum makespan. In the application, we have modified the PN-based heuristic search method proposed by [11] in conjunction with SPT for finding the near optimal firing sequence. The firing sequence for a motor variant, m_1, has obtained. The machines along with the corresponding schedule determined by the firing sequence are shown in the Gantt chart in Fig. 10.

Prodn Dept	Motor Production	Start 20-sep-07 12:00 AM	Finish 21-sep-07 12:00 AM
Multifunctional MC	Winding	20-sep-07 12:00 AM	20-sep-07 2:00 AM
Injection MC	Fabrication	20-sep-07 12:00 AM	20-sep-07 4:22 AM
Injection MC	Fabrication	20-sep-07 12:00 AM	20-sep-07 4:07 AM
Stamping MC	Fabrication	20-sep-07 12:00 AM	20-sep-07 3:21 AM
Multifunctional MC	Cutting	20-sep-07 2:00 AM	20-sep-07 5:00 AM
Workbench	Assembly	20-sep-07 2:00 AM	20-sep-07 6:30 AM
Pressing MC	Assembly	20-sep-07 4:22 AM	20-sep-07 6:28 AM
Pressing MC	Assembly	20-sep-07 5:00 AM	20-sep-07 8:12 AM
Inserting MC	Assembly	20-sep-07 6:30 AM	20-sep-07 10:00 AM
Fusing MC	Assembly	20-sep-07 10:00 AM	20-sep-07 12:42 PM
Caulking MC	Assembly	20-sep-07 12:42 PM	20-sep-07 2:22 PM
Caulking MC	Assembly	20-sep-07 2:22 PM	20-sep-07 4:34 PM

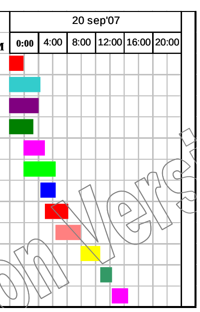

Figure 10. The Gantt chart suggesting machines and operations schedule

5. CONCLUSIONS

In view of the importance of dynamic modeling and visualization in decision making support in system reconfiguration and the lack of research, we propose to model RMSs with focus on the process of reconfiguring manufacturing resources based on PN techniques. To meet the modeling difficulties resulting from the fundamental issues in RMSs, we introduce a new formalism of CTPNs. Variety handling is accomplished by attaching specific data to tokens, which are used to represent various objects. A mechanism including reconfigurable transitions, inhibitor arcs and machine class concept places are defined to accommodate production changeovers. In conjunction with colored tokens, timing is introduced to address the selection of proper machines and constraint satisfaction. The application results have proven the potential of the proposed formalism to model RMSs.

REFERENCES

[1] Y. Koren, U. Heisel, F. Jovance, T. Moriwoki, G. Pritschow, A.G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," **Annals of the CIRP**, Vol. 2, 1999, pp. 1-13.

[2] M.R. Abdi, and A.W. Labib, 2004, "Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process," **Int. J. Prod. Res.**, Vol. 42, 2004, pp. 3055-3076.

[3] R.G. Qiu, P. McDonnell, S. Joshi, and D.W. Russell, 2005, "A heuristic game theoretic approach to resource sharing in reconfigurable manufacturing," **Int. J. Prod. Res.**, Vol. 25, 2005, pp. 78-87.

[4] P. Spicer, D. Yip-Hoi, and Y. Koren, "Scalable reconfigurable equipment design principles," **Int. J. Prod. Res.**, Vol. 43, 2005, pp. 4839-4852.

[5] A.M.A. Youssef, and H.A. ElMaraghy, "Modeling and optimization of multiple-aspect RMS configurations," **Int. J. Prod. Res.**, Vol. 44, 2006, pp. 4929-4958.

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

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

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

[9] F.D.J. Bowden, "A brief survey and synthesis of the roles of time in Petri nets," **Math. Comput. Modeling**, Vol. 31, 2000, pp. 55-68.

[10] L. Zhang, Process Platform-based Production Configuration for Mass Customization. PhD Dissertation,

Division of Systems and Engineering Management, Nanyang Technological University, 2007, Singapore.
[11] D.Y. Lee, and F. DiCesare, “Scheduling flexible manufacturing systems using Petri nets and heuristic search,” **IEEE Trans. Robot. and Autom.**, Vol. 10, 1994, pp. 123-132.