6-2009

Modelling reconfigurable manufacturing systems with coloured timed Petri nets

Lianfeng ZHANG
University of Groningen

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

Citation

ZHANG, Lianfeng and RODRIGUES, Brian. Modelling reconfigurable manufacturing systems with coloured timed Petri nets. (2009). *International Journal of Production Research*. 47, (16), 4569-4591. Research Collection Lee Kong Chian School Of Business.
Available at: https://ink.library.smu.edu.sg/lkcsb_research/1823

MODELING RECONFIGURABLE MANUFACTURING SYSTEMS WITH COLORED TIMED PETRI NETS

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2007-IJPR-0765.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	22-Jan-2008
Complete List of Authors:	Zhang, Linda; University of Groningen, Operations Rodrigues, Brian; Singapore Management University, Business School
Keywords:	MANUFACTURING MANAGEMENT, MANUFACTURING SYSTEMS
Keywords (user):	reconfigurable manufacturing systems, Petri net modeling

MODELING RECONFIGURABLE MANUFACTURING SYSTEMS WITH COLORED TIMED PETRI NETS

Lianfeng Zhang*† and Brian Rodrigues‡

†Department of Operations, University of Groningen,
Landleven 5, 9747 AD Groningen, The Netherlands
‡Lee Kong Chian School of Business, Singapore Management University
50 Stamford Road, Singapore 178899

ABSTRACT

Reconfigurable manufacturing systems (RMSs) have been acknowledged as a promising means of providing manufacturing companies with the required production capacities and capabilities. This is accomplished through reconfiguring system elements over time for a diverse set of individualized products often required in small quantities and with short delivery lead times. Recognizing the importance of dynamic modeling and visualization in decision-making support in RMSs and the limitations of current research, we propose in this work to model RMSs with Petri net (PN) techniques focusing on the process of reconfiguring system elements while considering constraints and system performance. In view of the modeling challenges, including variety handling, production variation accommodation, machine selection, and constraint satisfaction, we develop a new formalism of colored timed PNs. In conjunction with colored tokens and timing in colored and timed PNs, we also define a reconfiguration mechanism to meet modeling challenges. An application case from an electronics company producing mobile phone vibration motors is presented. Also reported are system analysis and application results, which show how the proposed formalism can be used in the reconfiguration decision making process.

KEYWORD

Reconfigurable manufacturing systems, colored PNs, timed PNs, reconfiguration mechanism.

1. INTRODUCTION

With increasing global competition, traditional manufacturing systems, e.g., dedicated manufacturing systems, mass production systems, flexible manufacturing systems, have become inadequate in supporting the rapid production of customized products with low costs and high profitability (Koren and Ulsoy, 1997). In response to

---

* Corresponding author. Email: L.Zhang@rug.nl

the limitations of such systems and the fast changing environments, reconfigurable manufacturing systems (RMSs) have been put forward as a promising means for manufacturing companies to produce products while meeting individualized customer requirements (Koren et al., 1999). Since companies must provide the needed capacities and capabilities for fulfilling diverse products which are often required in small quantities and with short delivery lead times, a major concern in RMSs is the quick reconfiguration of existing system elements - manufacturing resources such as machines, the associated tools, fixtures and setups - to provide changing production requirements (Mehrabi et al., 2000). Current research has approached RMSs from different aspects and delivered a number of analytical models (e.g., Abdi and Labib, 2004b; Qiu et al., 2005; Spicer et al., 2005; Youssef and El Maraghy, 2006). On one hand, such models provide insight into RMS planning, design and operations. On the other hand, the complexity involved in the model formulation tends to limit understanding. Moreover, the implicit assumptions underpinning these models, which often contradict the counterparts in the real world, make model implementation difficult.

While most researchers focus on problem formulation and solution development statically, limited attempts have been made to explore decision-making support based on dynamic and visual modeling techniques. Recognizing the importance of dynamic modeling and visualization in decision-making support in RMSs and the limitations of current research, we propose to model RMSs with focus on the process of reconfiguring manufacturing resources from a number of alternatives for given products while considering constraints and system performance. This has the potential to help companies make decisions in reconfiguring manufacturing resources to fulfill fast changing production requirements. Along with the fundamental issues in RMSs, we first highlight several challenges in RMS modeling as follows:

1.1 Challenges in modeling RMSs

(i) Variety handling

Diverse custom-specific products are involved in an RMS. Although current design practice, e.g., platform-based product family design, has brought similarities into end-products, the large number of individualized products inevitably lead 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 a high variety of material items and end-products in system models. Further, in spite of the inclusion of high product variety, a compact and representative model is needed to facilitate users' understanding, interpretation and for easy communication. Consequently, this highlights the importance of designing a proper modeling formalism to cope with high product variety.

(ii) Process variation accommodation

The large number of individual products and the resulting material items are associated with diverse design specifications, i.e., design parameters along with the specific value instances. In turn, changes in design specifications lead to many changeovers in production processes of both material items and end-products. Such changeovers are reflected as variations in machines, operations and operations sequences. With an attempt to provide companies with decision support, e.g., in selecting proper machines (including the associated tools, fixtures and setups), a system model should be able to capture and reflect these variations. Accordingly, the modeling formalism must provide proper mechanisms to accommodate changes.

(iii) Machine selection

In RMSs, a number of production 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 an end-product. Moreover, it is common that different machines are able to perform operations on same material items to complete same jobs, where, in most cases, these operations incur different cycle times. Similarly, only one machine is used to process material items at one time. On the other hand, companies are often forced to produce various products in concurrent time periods using same sets of manufacturing resources. This highlights the importance of selecting proper machines and processes to produce diverse end-products. The selection will need to contribute to the improvement of system performance attributes, e.g., throughput, machine utilization, quality. Hence, a modeling formalism should facilitate decision making in selecting machines and processes.

(iv) Constraint satisfaction

In RMSs, many restrictions or constraints can be observed, especially in the production process-planning phase. These constraints are inherent in the selection of machines and operations. For example, if a machine can perform operations only on alloy steel, it would be inappropriate to adopt it to perform operations on aluminum. The constraints are also associated with the specific design of product items, the capabilities of machines, and the availabilities of material items and machines. It is fundamental to cope with such constrains in modeling RMSs in order to build viable models. Accordingly, a modeling formalism should provide the ability to address the constraints in RMSs.

1.2 Strategy for solution

In view of their executability and graphical representation, PNs have been well recognized as a powerful modeling, simulation and evaluation tool for complex flows and processes (Peterson, 1981). Since PN models are graphical and derived from the logical sequence of systems, they are easy to understand and to communicate. Many extensions have been made to PNs to enhance the modeling power, among which coloured Petri nets (CPNs, Jensen 1995) and timed Petri nets (TPNs, Ramachandani, 1974) are of particular interest in this work. CPNs are able to provide a concise, flexible and manageable representation of large manufacturing systems by attaching a variety of colors to tokens (Jensen, 1995). By including timing, TPNs can capture the physical behaviors of systems by assuming specific durations for various system activities.

With decision making support as an objective, this paper applies PN techniques to model RMSs. A new formalism of colored timed PNs (CTPNs) is proposed to cope with the modeling challenges. The basic concepts of CPNs and TPNs are adopted and further extended to define elements in the proposed modeling formalism. Colored tokens are used to represent various objects, e.g., raw materials, parts, WIP, assemblies, machines. Variety handling is accomplished by attaching specific data to colored tokens. 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 rest of the paper is structured as follows: Section 2 provides a review of literature in RMSs, PNs and their applications in manufacturing. The basic PNs and the proposed formalism are introduced in Section 3, following which details of modeling RMSs in terms of material items, machines, cycle times and operations using the formalism are presented in Section 4. Section 5 applies the formalism to model an RMS for producing vibration motors for mobile phones in an electronics company. System analysis and application results are presented as well. Concluding remarks together with future research possibilities are provided in Section 6.

2. RELATED WORK

2.1 RMSs development support

In their keynote paper, Koren et al. (1999) define an RMS and further discuss the key characteristics: modularity, integrability, customization, convertibility, scalability, and diagnosability. Since then many researchers have reported their work in planning, design and reconfiguration issues in RMSs. Two streams can be observed: process family identification and RMS design support. Some methods have been developed to cope with the identification of product families in accordance with proper manufacturing systems. Yigit and Allahverdi (2003) address the planning of modular product variety to be provided in an RMS using an integer nonlinear program. In a design strategy for RMSs, Abdi and Labib (2003) present a model based on analytical hierarchical process (AHP) to assist in the selection of the right manufacturing systems in conjunction with product families. As an extension of their previous work, Abdi and Labib (2004a) discuss a reconfiguration link between markets and manufacturing for grouping products into families and further assigning the product families to manufacturing systems. Again, AHP is used to structure the decision making process. Similarly, Galan et al. (2007) develop a systematic approach based on AHP and AVLC (average linkage clustering algorithm) to forming product families for a given RMS.

In the stream of RMS design support, research efforts have delivered methods to facilitate decision making in RMSs design. Abdi and Labib (2004b) address the evaluation of economic and manufacturing feasibility before costly implementation of an RMS design. Qiu et al. (2005) present an approach based on a non-cooperative game theoretic technique to address resource sharing in RMSs. Aiming at decision making in both the initial design and the reconfiguration stage, Youssef and ElMaraghy (2006) detail a genetic algorithm-based model to determine the number and arrangement of machines, machine types and operations assignment in different aspects of RMS configuration. With focus on design issues at the machine level in RMSs, Spicer et al. (2005) present an architecture for scalable machines in order to address one of the key characteristics of RMSs - scalability. Similarly, Katz (2007) introduces a series of principles for designing reconfigurable machines. However, the design approach aims at designing machines to be used in high-volume production lines rather than small volume production that are typical when RMSs are used.

To summarize, in spite of the many research efforts in addressing different issues in RMSs design and operations, the literature review suggests that support for decision making in RMSs based on dynamic modeling and visual representation is scarce. This is especially true with respect to the stage of reconfiguring manufacturing resources in response to the diverse production requirements of individualized products.

2.2 PNs applications in manufacturing

PNs are a family of operational formalism providing a framework for manufacturing systems design and operations. They have emerged as a promising approach to modeling manufacturing systems. As a graphical tool, PNs serve as a visual modeling technique and as a communication aid for describing models. As a mathematical tool, PNs can be exploited to perform qualitative and quantitative analysis of systems being modeled.

Moore and Gupta (1996) survey PNs applications in manufacturing and present a comprehensive review of PN models of flexible and automated manufacturing systems. Focusing on real time control and performance evaluation, Reddy et al. (1993) present an algorithm for qualitative and quantitative analysis of TPN models in manufacturing systems. Ravi Raju and Krishnaiah Chetty (1993) discuss a PN-based methodology for modeling and simulating AGVs in flexible manufacturing systems. Recognizing the need of expanding the power of TPNs due to the randomness and the number of variables involved, they introduce extended TPNs in their methodology. Liu et al. (2002) propose a workflow modeling language-based CPN called WFCP-nets (workflow based on coloured Petri nets) and apply it to the product development workflow. Chin et al. (2006) put forward methodologies based on the integration of IDEF 0 (Integrated DEFinition 0) and CPNs for modeling and simulating complicated manufacturing processes. Yu et al. (2003) present a KTCOPN (knowledge-based timed colored object-oriented PN) for modeling reconfigurable assembly systems (RASs). As claimed by the authors, by combining knowledge and object-oriented methods into timed colored PNs, the characteristics of RASs can be fully expressed. Researchers, e.g., Dotoli and Fanti (2004), Nandula and Dutta (2000), Jiang et al.

(2000), have also applied colored PNs, timed PNs and colored timed PNs to manufacturing systems modeling, analysis and control. The observation on available PN models is that most researchers have adopted the basic ideas of colored and timed PNs and further extended them to accommodate different modeling requirements of their own problem domains. Similarly, in this work, bearing in mind the modeling difficulties we develop a new formalism of CTPNs, which integrates the principles of CTNs and TPNs, for modeling RMSs.

3. MODELING FORMALISM BASED ON COLORED TIMED PETRI NETS

3.1 Basic PNs

As shown in Figure 1, a basic PN model can be represented as a directed bipartite graph involving two types of nodes: places and transitions, represented by circles and bars, respectively and linked by arrowed arcs. An arc can only connect two nodes of different types, i.e., either from a place to a transition or from a transition to a place, but not nodes of the same type. The places that are connected directly to/from a transition are called input/output places of the transition. For example, places p_2 and p_3 are input places of transition t_2; p_4 is an output place of t_2. The solid dots in p_2 and p_3 are tokens. In basic PNs, tokens are black and carry no specific data or information pertaining to a variety of individual objects. They simply play the role of counters with respect to hosting places.

<<<<<<<<<<<<<<<<<<<<<Insert Figure 1 Here>>>>>>>>>>>>>>>>>>

In manufacturing-related PN models, places usually represent conditions or manufacturing resources, e.g., machines, buffers. Together with tokens, they indicate the status of manufacturing resources or the availability of material items. Transitions usually relate to events, processes, or operations. The movement of tokens between places is controlled by transitions. Furthermore, the distribution of tokens in places defines the state of a system, such as items in a buffer, number of free servers, and availability of machines.

Definition 1: A *Petri Net* is a tuple $PN = (P, T, I, O, M_0)$, where

$P = \{p_i\}_m$ is a finite nonempty set of places;

$T = \{t_j\}_n$, $P \cap T = \phi$ is a finite nonempty set of transitions;

$I\left({}^{\bullet}t\right): P \mapsto N$ and $O\left(t^{\bullet}\right): P \mapsto N$ are input/output arc transfer functions that define the numbers of tokens in the set of input/output places, ${}^{\bullet}t / t^{\bullet}$, of a transition, t, respectively; and N is the set of nonnegative integers;

$M: P \mapsto N$ is a marking function that defines the distribution of tokens in all places when the system is in certain state; and M_0 is the initial marking of the system model.

The dynamic behavior of a PN model is described by markings. When a system changes from one state to another state, the marking of the system model changes accordingly. Two rules describing the dynamic behavior of a PN model are as follows:

(1) *Enabling rule*: A transition $t \in T$ is *enabled* in a marking M iff each of its input places holds a "sufficient" number of tokens, i.e., iff $M(p) \geq I\left({}^{\bullet}t\right), \forall p \in {}^{\bullet}t$.

(2) *Firing rule*: When a transition $t \in T$ fires in M, it creates M' by removing a certain number of tokens from its input places and depositing a certain number of tokens in its output places: $M'(p) = M(p) - I\left({}^{\bullet}t\right) + O\left(t^{\bullet}\right), \forall p \in P$.

3.2 Colored timed PNs

Unlike the identical black tokens carrying no information in the basic PNs, colors are introduced into PNs in order to build a compact and representative model of a complex system (Jensen, 1995). The colors essentially are specific data values pertaining to the objects represented by tokens. The data value may be of a complex type, e.g., a record where the first field is a real number, the second a text string, and the third is a list of integer pairs. Since each colored token is uniquely defined by a color, and vice versa, there is a one-to-one correspondence between colors and colored tokens. Hereafter, colors and colored tokens will be used interchangeably without causing any confusion.

Attempting to analyze the performance of a system model, Ramachandani (1974) introduces time delays into PNs, resulting in TPN models. In most TPN models, a global clock is defined to time system operations. 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 output transitions, i.e., it remains unavailable, and after the elapse of which the token becomes available and can be used to fire transitions. Time delays can be attached to places, transitions, or arc expressions (Desrochers and Al-Jaar, 1995; Jesen, 1995). When time delays are associated with places or transitions, the corresponding processes or operations bear the same time durations in spite of the fact that different colored tokens are present. Consequently, the PN models constructed cannot reflect the real situation, where different cycle times are required for a machine to process different material items. Zuberek (1991) concludes that the nature of the time parameter can be deterministic or stochastic. For a comprehensive review of time representations in PN models, interested readers are referred to Bowden (2000).

Unlike traditional manufacturing systems, in RMSs multiple machines possess the capability to carry out different operations on same material items for same jobs. In most cases, such machines take different cycle times to complete operations. To capture and model such characteristics, a type of special places is defined in the formalism to represent the class concepts of machines that can carry out same jobs. In conjunction with machine class concept places, inhibitor arcs are introduced to keep more than one machine from accessing same material items. To cope with the difficulties in modeling diverse cycle times associated with multiple machines and same jobs, arc expression functions, rather than input/output arc transfer functions in basic PNs, are introduced. Further, in response to the limitations of associating time delays with places and transitions, we define time delays in arc expressions. Thus the proposed formalism is able to capture and model different cycle times associated with same machines but different material items. Time delays can be obtained from a process platform of a process family in relation to a product family (Zhang et al., 2007). Figure 2 shows the graphical formalism of the proposed CTPNs.

<<<<<<<<<<<<<<<<<<<<<<Insert Figure 2 Here>>>>>>>>>>>>>>>>>>

Definition 2: A *Colored Timed Petri Net* is a tuple $CTPN = (P, T, \Sigma, C, E, h, d, M_0)$, where

(i) P is a set of places satisfying the relation

$$P = P^R \cup P^O \cup P^C \text{, where}$$

$P^R / P^O / P^C$, $P^R \cap P^O = \phi, P^R \cap P^C = \phi$ and $P^O \cap P^C = \phi$ are three finite non-empty disjoint sets of places.

A $p \in P^R$ denotes either a buffer or a machine and $p \in P^O$ indicates that a machine is working on material item(s); a $p \in P^C$ represents a machine class concept;

(ii) T, $P \cap T = \phi$ is a finite nonempty set of transitions such that

$$T = T^L \cup T^T \cup T^R \text{, where}$$

$T^L/T^T/T^R$, $T^L \cap T^T = \phi, T^L \cap T^R = \phi$ and $T^T \cap T^R = \phi$ are three finite nonempty disjoint sets of logical/timed/reconfigurable transitions, respectively.

Logical transitions are introduced to capture the logic of a system running. Thus, their firing indicates the satisfaction of preconditions of operations. The typical precondition is the presence of material items and machines to be used. Reconfigurable transitions are defined to model the situations, where multiple machines can perform same jobs and only one is used eventually. Their firing leads to the reconfiguration of proper machines. Timed transitions are defined to represent operations, thus 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, $C(p)$:

$$C(p) = \{o(c_{pi})' c_{pi}\}_I \text{, where}$$

$o(c_{pi})$ is the occurrence multiplicity of color c_{pi}.

$C(p)$ represents either material items or machines if $p \in P^R$; a machine working on material items if $p \in P^O$; or specific machines if $p \in P^C$;

(v) $h, h \subseteq P^C \times T^R$ is a set of inhibitor arcs that (1) connect machine class concept places to reconfigurable transitions only and (2) assumes two values: 1 and 0. $h(p,t) = 1, \forall p \in P^C, t \in T^R$ 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, $(t, p), \forall p \in P^O, t \in T^L$, to a timed arc expression:

$$E^T : T^L \times P^O \mapsto \vee\left(\wedge\, o(c_{p_m j})' c_{p_m j} \to o(c_{ps})' c_{ps} @ + d\right), \forall p_m \in {}^{\bullet}t, c_{p_m j} \in C(p_m), c_{ps} \in C(p) \ ,$$

where $\vee$ represents *Exclusive OR (XOR)*; $\wedge$ *AND*; and $\to$ "*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 the 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. Conforming to common practice, the occurrence of such output tokens is 1.

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:

$$E^U : \neg T^L \times \neg P^O \mapsto \vee o(c_{pt})' c_{pt}, \forall c_{pt} \in C(p), \text{ where}$$

$\vee$ 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.

(viii) M is the marking function and M_0 is the initial marking.

M is a combination of three functions:

$$M = (\xi, \rho, \tau), \text{ where}$$

$\xi : P \mapsto \{o(c_{pi})' c_{pi}\} \cup 0, \forall c_{pi} = C(p)$ is a marking function of available tokens;

$\rho : P \mapsto \{o(c_{pj})' c_{pj}\} \cup 0, \forall c_{pj} = C(p)$ 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 different remaining times to these tokens 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 $p, \forall p \in t^{\bullet}$.

Transition firing is considered to be instantaneous. A new local clock is created for every newly created token and the initial value of the clock is determined by the delay in the timed arc expression. When no transition is enabled, the time of the global clock is incremented by the value of the smallest local clock. An unavailable token in a place, where a local clock reaches zero, becomes available and the clock is destroyed.

4. MODELING RMSS WITH CTPNS

RMSs are expected to provide required manufacturing capabilities and capacities by reconfiguring existing machines on shop floors. In general, two types of machines are designed in an RMS. While multifunctional machines have the capability to perform a wide range of operations on different types of materials, specialized machines carry out a limited number of operations on certain material types. With the presence of such machines, it is possible that RMSs can always satisfy diverse production requirements. As with other manufacturing systems, RMSs adopt buffers as the common solutions between operations/processes to free machines. Since production volumes in RMSs are low, it is expected that material items are always available as required. Considering these issues, the formalism is developed with the following assumptions:

- Machines are always available and never breakdown;
- Buffers, be they for raw materials, parts, assemblies, WIP, or end-products, provide the capacities as required;
- Material items are always available; and
- Cycle times are deterministic and can be obtained, e.g., from a process platform.

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

4.1 Material items

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

As shown in Figure 3(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 Figure 3(b) denotes another status of the raw material: being processed by the machine represented by p_2. Since the occurrences of tokens in Figure 3 are 1, by default they are omitted. For illustrative simplicity and clarity, timed and untimed arc expressions and token definitions are provided when necessary rather than in all figures in the following.

<<<<<<<<<<<<<<<<<<<<<Insert Figure 3 Here>>>>>>>>>>>>>>>>>

4.2 Manufacturing resources

Since no considerations are given to machine breakdowns, repair and maintenance, 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 Figure 3(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), no token would appear in the corresponding place. On the contrary, there would be a token in the place representing "machine working on items". Figure 3(b) shows a busy machine represented by the white token in p_3. The data value attached to the white token is also shown in the figure.

4.3 Cycle times

To capture different cycle times in relation to same jobs and different machines, time delays representing cycle times are attached to timed arc expressions. They are located at the end of expressions, as shown in Figure 3. The expression in Figure 3(a) indicates that it will take 5 time units for machine, m_1, to complete the cutting operation on raw material, $a \cdot 1$. During the 5 time units after firing the logical transition, t_1, the token, $a \cdot 2$, created in p_3 is unavailable and represented by a white dot, as shown in Figure 3(b). At the instant of 5 time units, the operation is completed and the token becomes available, as shown in Figure 3(c). Accordingly, the timed transition, t_2, representing the cutting operation, is enabled and fires.

4.4 Operations

Before the occurrence of any operation, input material items and machines 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 completion, the 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 these characteristics, an operation is modeled by several places representing buffers, machines, and machine working on items, as shown in Figure 4.

The buffer places, p_1 and p_4, contain tokens, $a \cdot 3$ (representing the input material item) and $a \cdot 4$ (denoting the produced parent item), respectively. The machine place, p_2, shows the availability of the machine, m. Along with other relevant places, p_3 indicates the operation has not started yet in Figure 4(a); the operation is ongoing in Figure 4(b); and the operation has been completed in Figure 4(c).

<<<<<<<<<<<<<<<<<<<<<Insert Figure 4 Here>>>>>>>>>>>>>>>>

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

(i) Operations with individual machines

In production practice, for producing end-products, input material items go through a series of operations performed by different machines. The starting of the following operations depends on the completion of the previous ones and the availability of the machines to be used. Figure 5 shows an example of two sequential operations along with individual machines. The white token in p_3 indicates the operation is ongoing and has not been completed. Accordingly, the token representing the output parent item is not available in the WIP buffer p_4. As a result, the logical transition, t_3, is not enabled and cannot fire.

<<<<<<<<<<<<<<<<<<<<<Insert Figure 5 Here>>>>>>>>>>>>>>>>

When a parent item is formed by more than one child item, the operations required for producing these child items are often performed simultaneously by different machines. In some situations, such concurrent operations are vital for activity synchronization, the achievement of which affects the WIP inventory, their waiting time, production lead time, and eventually production costs. Figure 6 shows an example of 2 parallel operations with individual machines. The operation performed by the machine (represented by p_2) has been completed, indicated by the tokens in p_2 and p_4 (a WIP buffer). Since the operation performed by the other machine (represented by p_7) is ongoing, which is indicated by the white dot in p_6, the token representing the corresponding output item has not been created in the WIP buffer (represented by p_8). As a result, the logical transition, 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. In RMSs, such synchronization may be achieved through the selection of proper machines, production processes and the use of different scheduling policies.

<<<<<<<<<<<<<<<<<<<<<Insert Figure 6 Here>>>>>>>>>>>>>>>>>

(ii) Operations with shared machines

Figure 7 shows the situations, where operations are required to be performed by common machines. In Figure 7(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 the buffer place p_{j+1}. Since both t_2 and t_{i+1} require p_4 representing the shared machine, a conflict may occur if there is a token in it. To solve such conflicts, the common approach proposed by most researchers is to assign priorities to transitions (Bowden, 2000). In this work, we follow the common approach and assign different priority numbers (1, 2, …, *n*) to the competing logical transitions in connection with timed transitions, with one being the highest priority and *n* being the lowest priority.

<<<<<<<<<<<<<<<<<<<<<Insert Figure 7 Here>>>>>>>>>>>>>>>>>

For example, in Figure 7(a), since operation t_{i+1} depends on operation t_2, the priority number of t_1 will be 1 and that of t_i will be 2. In Figure 7(b), the 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 to solve the potential conflicts. In this situation, the assignment can be made according to the cycle times of represented operations (t_2 and t_4 in this case).

(iii)Operations with alternative machines

Figure 8(a) describes a general case that an operation can be performed by different machines. Both the two machines, m_1 (represented by p_5) and m_2 (represented by p_6), can work on the same item (token $a \cdot 1$ residing in the buffer place 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 introduction of the inhibitor arcs (the two dashed lines from p_4 to t_3 / t_4) limits the number of tokens residing in p_4 to 1 each time.

Essentially, the two reconfigurable transitions (t_3 and t_4), the two inhibitor arcs and the machine class concept place form the reconfiguration mechanism. Along with the preferred scheduling rules, e.g., SPT, LPT, the mechanism controls the selection, and further reconfiguration, of a proper machine to perform the operation. The timed arc expression shows different time delays for the two machines to complete their operations.

<<<<<<<<<<<<<<<<<<<<<<Insert Figure 8 Here>>>>>>>>>>>>>>>>>>

Figures 8(b) and 8(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 to resolve conflicts. In Figure 8(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. The reason for this is that the operation associated with t_i depends on the one associated with t_1. In Figure 8(c), priority numbers are assigned to t_1 and t_5. The priority assignment in this condition can be determined with consideration of the average cycle times associated with the two machines. Further, in conjunction with the preferred scheduling rules, the reconfiguration mechanism in each model controls the reconfiguration of machines.

5. APPLICATION CASE

The proposed modeling formalism has been tested in a company that manufactures a high variety of individualized vibration motors for mobile phones. Based on similarities in design and manufacturing, the company has classified the motor variants into several families. Variations exist in the production processes due to the many differences in design specifications of existing motors and those of new ones to be manufactured. Accordingly, the machines (including the necessary tools, fixtures and setups) on the shop floor have to be frequently reconfigured so as to form proper operations and production processes while using the existing manufacturing resources.

5.1 Model construction

For illustrative simplicity, modeling an RMS of one motor family is described. Figure 9 shows the common product structure of the motor family, where three major assemblies are frameassy (fassy), bracketassy (bassy) and armartureassy (aassy). Each is formed by several manufactured parts and/or purchased components. The weight and rubber holder are optional purchased components and may not be included in all motor variants at the same time. To fulfill diverse individual customer requirements, each of the above item families has a number of variations.

<<<<<<<<<<<<<<<<<<<<<<Insert Figure 9 Here>>>>>>>>>>>>>>>>>>

The reconfigurable manufacturing system includes 2 caulking machines, 1 inserting machine, 1 fusing machine, 2 stamping machines, 2 injection machines, 1 multifunctional machine, 2 pressing machines, 1 workbench, and several buffers. Table 1 shows the machines, the associated operations (described in general terms) and the corresponding 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. Each specific production process in relation to a motor variant starts from the manufacturing of ba (bracket a), bb (bracket b), tl (terminal), f (frame), and c (coil) and goes through a number of manufacturing/assembly operations for producing the three major assemblies, further abassy (WIP of aassy + bassy) and mainbody (WIP of abassy + fassy) and ends at the final assembly of motors. The processes of individual motor variants differ from one another in the specific machines, operations, cycle times, and operations sequences.

<<<<<<<<<<<<<<<<<<<<<<Insert Table 1 Here>>>>>>>>>>>>>>>>>>

By referring to the generic routing of the motor family, the system model, as shown in Figure 10, has been constructed using the proposed formalism. Table 2 shows the places, the represented system elements and the corresponding tokens.

<<<<<<<<<<<<<<<<<<<<<<Insert Figure 10 Here>>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<<Insert Table 2 Here>>>>>>>>>>>>>>>>>>

For illustrative simplicity without loss of generality, the tokens shown in Table 2 are not exhaustive. The colored tokens residing in buffer places are defined based on the corresponding items in each family. For example, in the raw material buffer, p_1, the tokens, $ba_1 \cdot 1$, $ba_2 \cdot 1$ and $ba_3 \cdot 1$, are defined to represent the raw materials of 3 bracket a variants: ba_1, ba_2 and ba_3; the tokens, $bb_1 \cdot 1$, $bb_2 \cdot 1$ and $bb_3 \cdot 1$, the raw materials of 3 bracket b variants: bb_1, bb_2 and bb_3; and the tokens, $tl_1 \cdot 1$, $tl_2 \cdot 1$ and $tl_3 \cdot 1$, the raw materials of three terminal variants: tl_1, tl_2 and tl_3. The tokens in machine places, e.g., p_3, are specified according to the machine names and capabilities, the types of materials that the machines can work on, the tools, fixtures and setups in relation to the operations to be performed. The tokens in places representing “machine working on material items” are defined based on the specific attribute data pertaining to the output parts/WIP/assemblies. For example, the tokens in p_2 (multifunctional machine working on coil raw materials) 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. The time delays in timed arc expressions are determined according to the cycle times involved in the process platform of the motor family. The timed arc expressions specify the possible machines for processing the given material items. Also specified are the cycle times that the machines may take to complete the relevant operations.

For instance, $(c_1 \cdot 1 \wedge w \rightarrow c_1\ @+2) \vee (c_2 \cdot 1 \wedge w \rightarrow c_2\ @+1.5) \vee (c_3 \cdot 1 \wedge w \rightarrow c_3\ @+2.4)$, of 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, the logical transition, t_1, fires immediately. However, the timed transition, t_6, will fire 2 hours later after the firing of t_1. The untimed arc expressions are defined to specify the input and output of transitions. For example, the arc expression, $tl_1 \vee tl_2 \vee tl_3$, of the output arc, (t_7, p_{11}), of the timed transition, t_7, shows the three possible output terminal variants: tl_1, tl_2 and tl_3.

Both the inserting machine (p_{24}) and the fusing machine (p_{25}) can perform the corresponding assembly operations to form aassy and abassy. To accommodate the reconfiguration, a machine class concept place (p_{23}), two reconfigurable transitions (t_{18} and t_{19}), and two inhibitor arcs ((p_{23}, t_{18}) and (p_{23}, t_{19})) are defined. The determination of machines is based on time delays in the timed arc expressions and the preferred schedule policies.

5.2 Model analysis

After construction, the model is analyzed to check (1) whether or not it is correct; and (2) whether or not it logically represents the system operations. Jensen (1995) introduces several methods to verify models with respect to dynamic properties, e.g., boundedness. Among these, P-invariant analysis is of particular interest to most researchers due to its easy-understandability and implementation, which we adopt in this work. Several P-invariants can be identified in the system model in Figure 10. The total number of busy machines and idle machines gives a P-invariant. In other words, in any system state, 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 the material items in buffers and the items being processed by machines. This P-invariant is obtained through mapping the items being processed to the corresponding raw material items.

Deadlock and conflicts have a major impact on the logical operations of system models (Jiang et al., 1999). Wang (1996) describes different types of conflicts that may occur in a system model. In conjunction with the priority numbers assigned to the competing logical transitions, the definition of colored tokens and arc expressions have solved the possible conflicts in the model in Figure 10. Wang and Wu (1998) generalize a Deadlock Detection algorithm (DDA) based on the work of D'Souza (1994) and David and Alla (1992) for deadlock analysis. By applying their DDA procedure to the system model in Figure 10, a feasible firing sequence is obtained, as shown in Figure 11. It leads the searching of DDA to reach the goal state, $M_{26} = \left(0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, 0, 0, 0, m_1\right)$, from the initial state,

$$M_0 = \begin{pmatrix} c_1 \cdot 1 + ba_1 \cdot 1 + bb_1 \cdot 1 + tl_1 \cdot 1 + f_1 \cdot 1, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, \\ 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0 \end{pmatrix}.$$

Thus, we conclude that the system model is deadlock-free.

<<<<<<<<<<<<<<<<<<<<*Insert Figure 11 Here*>>>>>>>>>>>>>>>>>>

5.3 Application results

The production performance considered in the application case is the makespan. An optimal firing sequence (with respect to the minimum accumulated processing time) of transitions in the system model in Figure 10 results the determination of proper machines for given products. In the mean time, it provides the schedule of the machines for producing products while leading to nearly minimum makespan. In the application, we have modified the PN-based heuristic search method proposed by Lee and DiCesare (1994) in conjunction with SPT for finding the near optimal firing sequence. The firing sequence for a specific motor variant, m_1, has obtained, as shown in Table 3. In the table, firing time is the model time that a transition fires. The machines along with the corresponding schedule determined by the firing sequence are shown in the Gantt chart in Figure 12. The application also indicates that different scheduling rules generate different schedules. This provides companies with greater flexibility in the sense that they can incorporate different scheduling rules that are appropriate in their particular manufacturing environments.

<<<<<<<<<<<<<<<<<<<<<Insert Table 3 Here>>>>>>>>>>>>>>>>>

<<<<<<<<<<<<<<<<<<<<<Insert Figure 12 Here>>>>>>>>>>>>>>>>>

6. CONCLUSIONS

Production requirements in terms of capacities and capabilities vary from time to time with the diverse array of individualized products in small quantities. RMSs have been accepted by both academia and industry alike as a promising means of providing companies with the required capacities and capabilities, when required. One of the major concerns in RMSs is the quick reconfiguration of existing manufacturing resources in response to different production requirements of end-products while considering constraints and production performance. Recognizing the importance of dynamic modeling and visualization in decision making support in system reconfiguration and the lack of research, we proposed to model RMSs with focus on the process of reconfiguring manufacturing resources based on PN techniques.

The fundamental issues in RMSs have raised several challenges in modeling RMSs. They include variety handling, production variation accommodation, machine selection, and constraint satisfaction. To meet the modeling challenges, we introduced a new formalism based on 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 results of application case have proven the potential of the proposed formalism to model RMSs.

While the proposed formalism is able to provide a starting point for RMS modeling, it does not consider machine breakdowns. This, in turn, provides an opportunity to extend this study. In future research, a more comprehensive modeling formalism may be developed to capture machine breakdowns and the corresponding impact on the system models to be constructed. The resulting models can be expected to reflect real situations more closely by taking into account machine availability. In addition, further research can be made to study the impact of different timing concepts, e.g., time representation, time generation, in conjunction with the development of extended formalisms on the final results to be obtained.

Process platforms have been recognized as being able to provide companies with well-structured mechanisms to generate potential production processes for diverse products while eliminating unnecessary production variations (Zhang, 2007). Using such processes, companies are expected to be able to determine optimal configurations to be used on the shop floor. In this respect, another direction of further research may consider the integration of process platforms and modeling formalism, from which system models can be generated automatically. The ultimate goal is to achieve production automation with respect to production process planning and system reconfiguration, towards which this work has contributed.

REFERENCES

Abdi, M.R. and Labib, A.W., 2003, A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP): A case study, *International Journal of Production Research*, 42(15), pp. 3055-3076.

Abdi, M.R. and Labib, A.W., 2004a, Trouping and selecting products: The design key of reconfigurable manufacturing systems (RMSs), *International Journal of Production Research*, 42(3), pp. 521-546.

Abdi, M.R. and Labib, A.W., 2004b, Feasibility study of the tactical design justification for reconfigurable manufacturing systems using the fuzzy analytical hierarchical process, *International Journal of Production Research*, 42(15), pp. 3055-3076.

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

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

David, R. and Alla, H., 1992, *Petri Nets and Grafcet-Tools for Modeling Discrete Event Systems*. Prentice-Hall, Englewood cliffs, NJ.

Desrochers, A.A. and Al-Jaar, R.Y., 1995, *Applications of Petri Nets in Manufacturing Systems*. New York: IEEE Press.

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

D'Souza, K.A., 1994, A control model for detecting deadlocks in an automated machining cell, *Computers & Industrial Engineering*, 26, pp. 133-139.

Galan, R., Racero, J., Eguia, I. and Garcia, J.M., 2007, A systematic approach for product families formation in reconfigurable manufacturing systems, *Robotics & Computer-Integrated Manufacturing,* 23(5), pp. 489-502.

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

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

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

Katz, R., 2007, Design principles of reconfigurable machines, *International Journal of Advance Manufacturing Technology,* 34 (5-6), pp. 430-439.

Koren, Y. and Ulsoy, A.G., 1997, Reconfigurable manufacturing systems. Engineering Research Center for Reconfigurable Machining Systems (ERC/RMS) Report 1, The University of Michigan, Ann Arbor.

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

Lee, D.Y. and DiCesare, F., 1994, Scheduling flexible manufacturing systems using Petri nets and heuristic search, *IEEE Transactions on Robotics and Automation*, 10, pp. 123-132.

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

Mehrabi, M.G., Ulsoy, A.G. and Koren, Y., 2000, Reconfigurable manufacturing systems: Key to future manufacturing, *Journal of Intelligent Manufacturing*, 11, pp. 403-419.

Moore, K.E. and Gupta, S.M., 1996, Petri net models of flexible and automated manufacturing systems: A survey, *International Journal of Production Research*, 34(11), pp. 3001-3035.

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

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

Qiu, R.G., McDonnell, P., Joshi, S. and Russell, D.W., 2005, A heuristic game theoretic approach to resource sharing in reconfigurable manufacturing, *International Journal of Production Research*, 25(1/2), pp. 78-87.

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

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

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

Spicer, P., Yip-Hoi, D. and Koren, Y., 2005, Scalable reconfigurable equipment design principles, *International Journal of Production Research*, 43(22), pp. 4839-4852.

Wang, L.C., 1996, Object-oriented Petri nets for modeling and analysis of automated manufacturing systems, *Computer Integrated Manufacturing Systems*, 26(2), pp. 111–125.

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

Yigit, A.S. and Allahverdi, A., 2003, Optimal selection of module instances for modular products in reconfigurable manufacturing systems, *International Journal of Production Research*, 41(17), pp. 4063-4074.

Youssef, A.M.A. and ElMaraghy, H.A., 2006, Modeling and optimization of multiple-aspect RMS configurations, *International Journal of Production Research*, 44(22), pp. 4929-4958.

Yu, J.F., Yin, Y.H., Sheng, X.J. and Chen, Z.N., 2003, Modeling strategies for reconfigurable assembly systems, *Assembly Automation*, 23(3), pp. 266-272.

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

Zhang, L., Jiao, J. and Helo, P., 2007, Process platform representation based on unified modeling language, *International Journal of Production Research*, 45(2), pp. 323-350.

Zuberek, W.M., 1991, Timed Petri nets: Definitions, properties, and applications, *Microelectronics and Reliability*, 31(4), pp. 627-644.

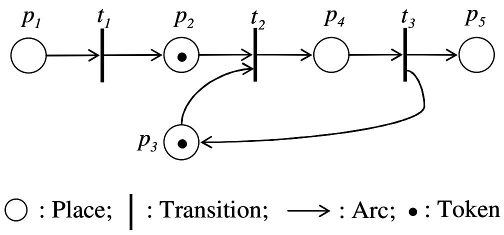


Figure 1: A PN model

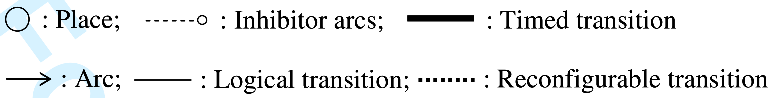


Figure 2: Graphical formalism of colored timed Petri nets

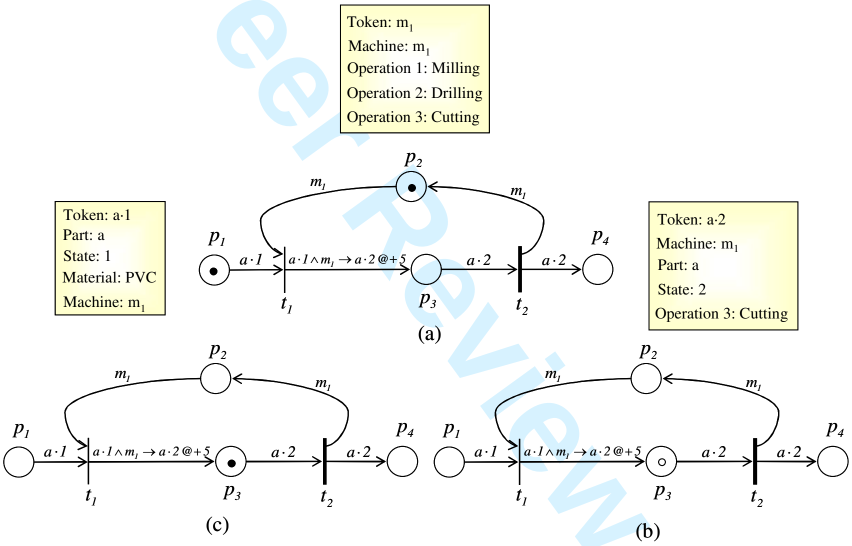


Figure 3: Modeling RMSs using CTPN-based formalism

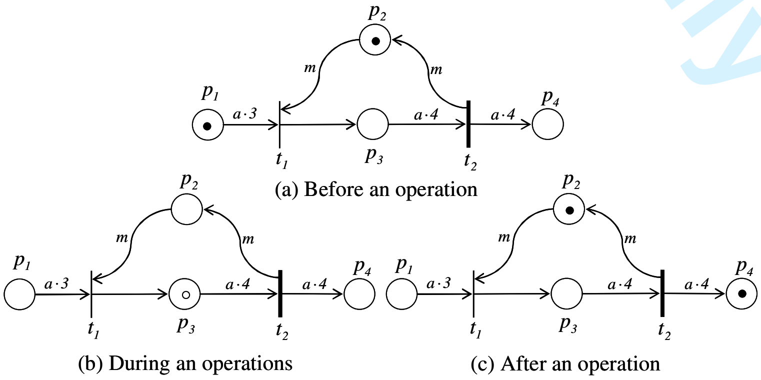


Figure 4: Modeling operations in RMSs

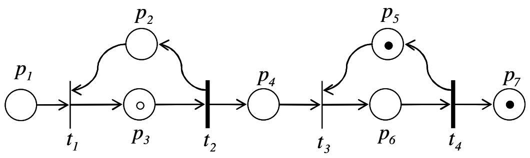


Figure 5: Sequential operations with individual machines

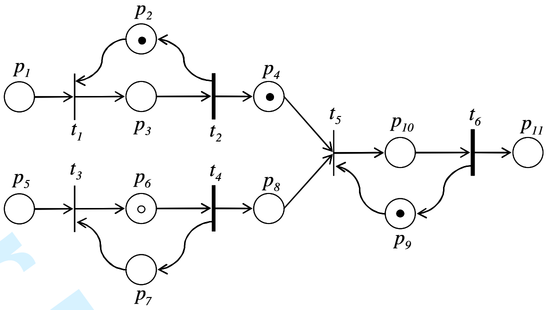
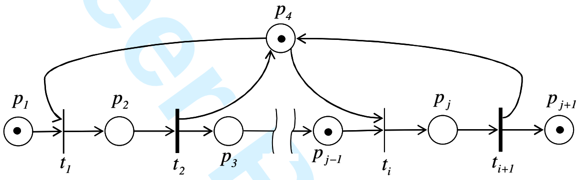
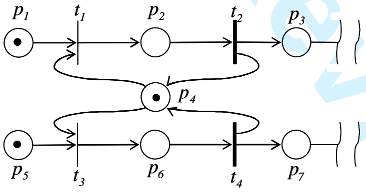


Figure 6: Parallel operations with individual machines

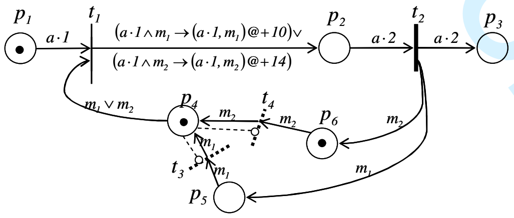


(a) Sequential operations with common machines

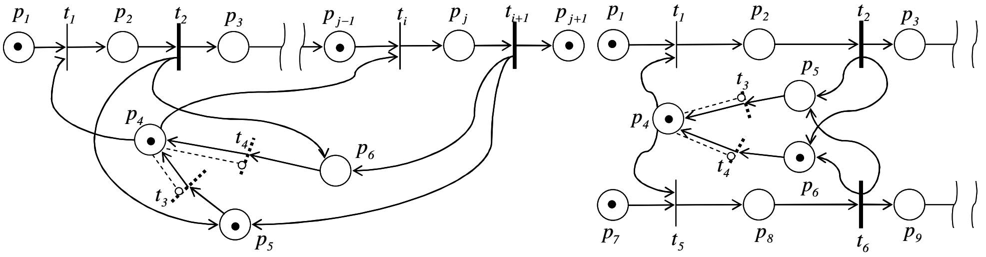


(b) Parallel operations with common machines

Figure 7: Operations with shared machines



(a) Alternative machines for one operation



(b) Alternative machines for sequential operations (c) Alternative machines for parallel operations

Figure 8: Operations with alternative machines

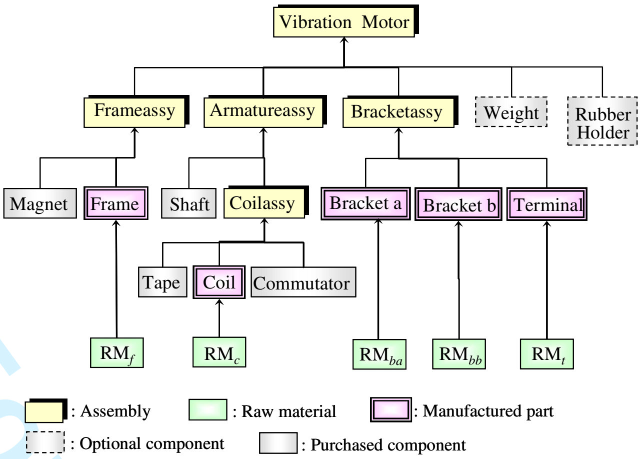


Figure 9: The common product structure of the motor family

$$M_0 = (c_1 \cdot 1 + ba_1 \cdot 1 + bb_1 \cdot 1 + tl_1 \cdot 1 + f_1 \cdot 1, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_1 = (ba_1 \cdot 1 + bb_1 \cdot 1 + tl_1 \cdot 1 + f_1 \cdot 1, c_1, 0, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_2 = (bb_1 \cdot 1 + tl_1 \cdot 1 + f_1 \cdot 1, c_1, 0, 0, ba_1, i, 0, 2's, 0, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_3 = (tl_1 \cdot 1 + f_1 \cdot 1, c_1, 0, 0, ba_1, 0, bb_1, 2's, 0, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_4 = (tl_1 \cdot 1, c_1, 0, 0, ba_1, 0, bb_1, s, f_1, 0, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_5 = (tl_1 \cdot 1, 0, w, 0, ba_1, 0, bb_1, s, f_1, c_1, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_6 = (0, 0, 0, tl_1, ba_1, 0, bb_1, s, f_1, c_1, 0, 0, t_1 + cm_1 + sf_1 + mg_1, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_7 = (0, 0, 0, tl_1, ba_1, 0, bb_1, s, f_1, 0, 0, 0, sf_1 + mg_1, ca_1, 0, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_8 = (0, 0, 0, tl_1, ba_1, 0, bb_1, 2's, 0, 0, 0, f_1, sf_1 + mg_1, ca_1, 0, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_9 = (0, 0, 0, tl_1, ba_1, i, 0, 2's, 0, 0, bb_1, f_1, sf_1 + mg_1, ca_1, 0, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{10} = (0, 0, 0, tl_1, 0, 2'i, 0, 2's, 0, 0, ba_1 + bb_1, f_1, sf_1 + mg_1, ca_1, 0, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{11} = (0, 0, 0, tl_1, 0, 2'i, 0, 2's, 0, 0, ba_1 + bb_1, 0, sf_1, ca_1, 0, 0, p, fa_1, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{12} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, tl_1 + ba_1 + bb_1, 0, sf_1, ca_1, 0, 0, p, fa_1, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{13} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, sf_1, ca_1, 0, BA_1, 0, fa_1, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{14} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, sf_1, 0, b, BA_1, 0, fa_1, ca_1, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{15} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, sf_1, 0, b, BA_1, p, 0, ca_1, 0, fa_1, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{16} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, sf_1, 0, b, BA_1, p, 0, ca_1, 0, fa_1, 0, in, 0, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{17} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, BA_1, p, 0, 0, 0, fa_1, aa_1, 0, 0, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{18} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, BA_1, p, 0, 0, 0, fa_1, aa_1, fu, 0, 0, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{19} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, BA_1, fa_1, aa_1, fu, 0, 0, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{20} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, BA_1, fa_1, 0, fu, in, 0, aa_1, 0, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{21} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, fa_1, 0, 0, in, 0, 0, aaba_1, 0, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{22} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, fa_1, 0, 0, in, fu, 0, 0, aaba_1, 2'cu, 0, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{23} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, cu, mb_1, wt_1 + rh_2, 0, 0, 0)$$

$$\downarrow$$

$$M_{24} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, wt_1 + rh_2, mb_1, 0, 0)$$

$$\downarrow$$

$$M_{25} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, cu, 0, 0, 0, m_1, 0)$$

$$\downarrow$$

$$M_{26} = (0, 0, w, 0, 0, 2'i, 0, 2's, 0, 0, 0, 0, 0, 0, b, 0, 2'p, 0, 0, 0, 0, 0, 0, in, fu, 0, 0, 0, 2'cu, 0, 0, 0, 0, m_1)$$

Figure 11: A feasible firing sequence leading to the goal state

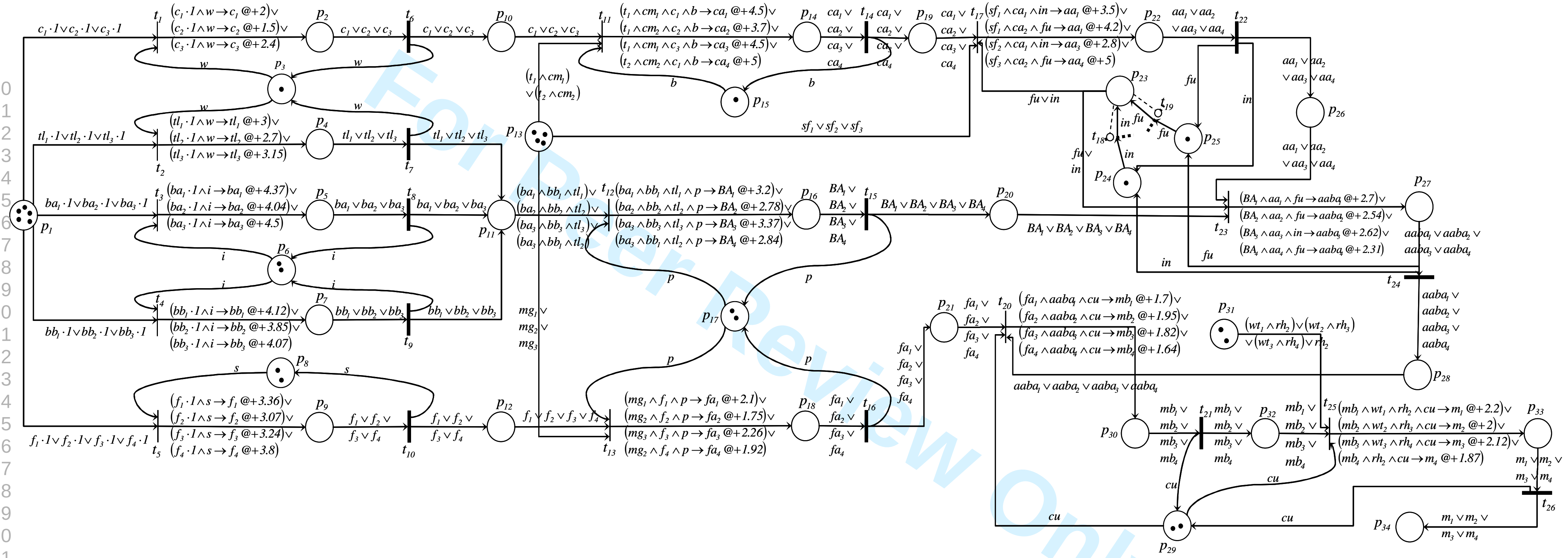
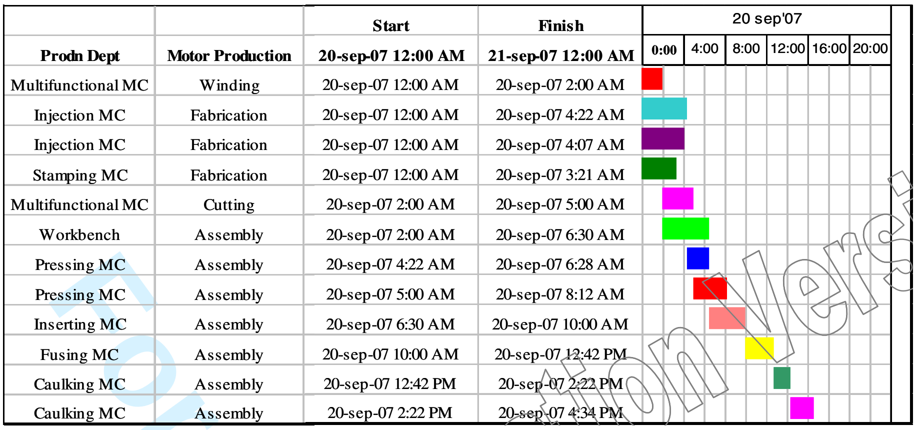


Figure 10: The CTPN model of the RMS



		Start	Finish
Prodn Dept	**Motor Production**	**20-sep-07 12:00 AM**	**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 12: The Gantt chart suggesting machines and operations schedule

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

Table 2: Places, represented system elements and tokens

Places	System Elements	Tokens
p_1	Raw material buffer for bracket a, bracket b, terminal, coil, frame	$ba_1 \cdot 1$, $ba_2 \cdot 1$, $ba_3 \cdot 1$, $bb_1 \cdot 1$, $bb_2 \cdot 1$, $bb_3 \cdot 1$, $tl_1 \cdot 1$, $tl_2 \cdot 1$, $tl_3 \cdot 1$, $c_1 \cdot 1$, $c_2 \cdot 1$, $c_3 \cdot 1$, $f_1 \cdot 1$, $f_2 \cdot 1$, $f_3 \cdot 1$, $f_4 \cdot 1$
p_2	Multifunctional mach processing coil raw materials	c_1, c_2, c_3
p_3	Multifunctional machine	w
p_4	Multifunctional mach processing terminal raw materials	tl_1, tl_2, tl_3
p_5	Injection machine processing bracket a raw materials	ba_1, ba_2, ba_3
p_6	Injection machine	i
p_7	Injection machine processing bracket b raw materials	bb_1, bb_2, bb_3
p_8	Stamping machine	s
p_9	Stamping machine processing frame raw materials	f_1, f_2, f_3, f_4
p_{10}	WIP buffer for coil	c_1, c_2, c_3
p_{11}	WIP buffer for bracket a, bracket b and terminal	ba_1, ba_2, ba_3, bb_1, bb_2, bb_3, tl_1, tl_2, tl_3
p_{12}	WIP buffer for frame	f_1, f_2, f_3, f_4
p_{13}	Raw material buffer for tape, commutator, magnet, shaft	t_1, t_2, cm_1, cm_2, mg_1, mg_2, mg_3, sf_1, sf_2, sf_3
p_{14}	Operator assembling coilassy on workbench	ca_1, ca_2, ca_3, ca_4
p_{15}	Workbench	b
p_{16}	Pressing machine processing bassy	BA_1, BA_2, BA_3, BA_4
p_{17}	Pressing machine	p
p_{18}	Pressing machine processing frameassy	fa_1, fa_2, fa_3, fa_4
p_{19}	WIP buffer for coilassy	ca_1, ca_2, ca_3, ca_4
p_{20}	WIP buffer for bassy	BA_1, BA_2, BA_3, BA_4
p_{21}	WIP buffer for frameassy	fa_1, fa_2, fa_3, fa_4
p_{22}	Inserting (or fusing) machine processing aassy	aa_1, aa_2, aa_3, aa_4
p_{23}	Class concept of inserting & fusing machines	in, fu
p_{24}	Inserting machine	in
p_{25}	Fusing machine	fu
p_{26}	WIP buffer for aassy	aa_1, aa_2, aa_3, aa_4
p_{27}	Inserting (or fusing) mach processing aassy & bassy	$aaba_1$, $aaba_2$, $aaba_3$, $aaba_4$
p_{28}	WIP buffer for abassy	$aaba_1$, $aaba_2$, $aaba_3$, $aaba_4$
p_{29}	Caulking machine	cu
p_{30}	Caulking machine processing mainbodies	mb_1, mb_2, mb_3, mb_4
p_{31}	Raw material buffer for weights and rubber holders	wt_1, wt_2, wt_3, wt_4, rh_1, rh_2, rh_3, rh_4
p_{32}	WIP buffer for mainbodies	mb_1, mb_2, mb_3, mb_4
p_{33}	Caulking machine processing motors	m_1, m_2, m_3, m_4
p_{34}	End-product buffer for motors	m_1, m_2, m_3, m_4

Table 3: The firing sequence leading to the optimal reconfiguration of machines

Fired Transitions	Firing Time	Input Tokens	Created Tokens	Fired Transitions	Firing Time	Input Tokens	Created Tokens
t_1	0	$c_1 \cdot 1$, w	c_1	t_{16}	6:28	fa_1	fa_1, p
t_3	0	$ba_1 \cdot 1$, i	ba_1	t_{18}	6:28	in	in
t_4	0	$bb_1 \cdot 1$, i	bb_1	t_{14}	6:30	ca_1	ca_1, b
t_5	0	$f_1 \cdot 1$, s	f_1	t_{17}	6:30	sf_1, ca_1, in	aa_1
t_6	2’	c_1	c_1, w	t_{19}	6:30	fu	fu
t_2	2’	$tl_1 \cdot 1$, w	tl_1	t_{15}	8:12	BA_1	BA_1, p
t_{11}	2’	c_1, t_1, cm_1, b	ca_1	t_{22}	10:00	aa_1	aa_1, in
t_{10}	3:21	f_1	f_1, s	t_{23}	10:00	BA_1, aa_1, fu	$aaba_1$
t_8	4:07	bb_1	bb_1, i	t_{24}	12:42	$aaba_1$	$aaba_1$, fu
t_9	4:22	ba_1	ba_1, i	t_{20}	12:42	$aaba_1$, fa_1, cu	mb_1
t_{13}	4:22	f_1, mg_1, p	fa_1	t_{21}	14:22	mb_1	mb_1, cu
t_7	5:00	tl_1	tl_1, w	t_{25}	14:22	mb_1, wt_1, rh_2, cu	m_1
t_{12}	5:00	ba_1, bb_1, tl_1, p	BA_1	t_{26}	16:34	m_1	m_1, cu