Int J Adv Manuf Technol (2004) 23: 594–605
DOI 10.1007/s00170-003-1682-2

ORIGINAL ARTICLE

H.-P. Hsu · J.-Y. Chen · C.-T. Su

An activity-based predicate/transition net model of operational control planning for a flexible manufacturing system

Received: 11 November 2002 / Accepted: 21 February 2003

Abstract An effective flexible manufacturing system (FMS) relies on a hierarchy of decisions, including the control of the FMS operation. The FMS operation usually is dynamically constrained by the limited resources such as pallets, machines, tools, carts, etc. Most analytical models make many assumptions and oversimplify the complicated decision problems. This study proposes the predicate/transition (Pr/Tr) net, a high level petri net, as a model for operational control planning. Firstly, the activities (modes) and their resources usage in FMS were analysed and aggregated into activity sets. Then, the flow of parts among activities was traced to obtain the "mode transition diagram", and then the Pr/Tr net model was introduced. We incrementally defined the predicates and transitions into this model. Finally, a comprehensive FMS Pr/Tr net model was derived. By implementing it into a rule-based simulation model, it is well suited for FMS operational control planning because of its inclusiveness and high flexibility.

Keywords Flexible Manufacturing System (FMS) Predicate/Transition net · Petri net · Operational control planning · Simulation model

1 Introduction

A flexible manufacturing system (FMS) is a highly automatic manufacturing system. It is a computer controlled production system capable of processing part types of low quantity and high diversity in a flexible manner. Usually, an FMS consists of three elementary components: (1) numerically controlled manufacturing machines (NC, DNC, CNC), including the tools to operate these machines; (2) an automated material handling system (MHS) to move the work pieces through the system; (3) an on-line computer control system to manage the entire FMS, including the NC machines and the MHS. For an FMS, a central computer controls all operations, which may occur simultaneously, asynchronously, or from a parallel base depending upon the constraints of the resource(s) required and the availability. An operating FMS is a dynamic system. Therefore, in order to clearly describe its behaviours, a powerful tool is required.

Models have been used for problem solving for a long time. An easy way to classify models was proposed by Suri [1]. According to the review by Suri, there are basically two kinds of models, a generative model and an evaluative model. Many kinds of different models have been used to illustrate an FMS in the past, including a static allocation model, a queuing network model, a simulation model, a perturbation analysis model, a petri net and an example-based model [2, 3, 4]. Each type of model has both its power and its limits. This paper shows that a simulation model, compared to other methodologies, is a well-suited tool to analyse the performance of different releasing and dispatching policies for an FMS at the operational level.

Some researchers have used a petri net model to analyse an FMS [5, 6, 7, 8, 9]. The major weakness of using an ordinary petri net to model a complicated system is the resulting unmanageable petri net size [10]. As a consequence, other extended petri nets aiming at

---

H.-P. Hsu
Department of Business Administration,
MingHsin University of Science and Technology,
Hsin-Chu, Taiwan ROC

J.-Y. Chen
Department of Computer and Information Engineering,
National Central University, Chung-Li, Taiwan ROC

C.-T. Su (✉)
Department of Industrial Engineering and Management,
National Chiao Tung University,
Hsin-Chu, Taiwan ROC
E-mail: ctsu@cc.nctu.edu.tw

empowering the modelling capability and reducing the size have been proposed by different researchers. For example, Alla et al. [11] used coloured petri net to model and validate an FMS; Gentina and Corbeel [12] proposed a coloured adaptive structure petri net to automatically design a hierarchical control for an FMS and Genrich et al. [13] used a predicate/transition net (Pr/Tr net) to model a generalised system in a resource usage environment. However, a Pr/Tr net has never been applied to an FMS. This paper extends the modelling tool concept of Genrich et al. [13]. A Pr/Tr net model with its graphical capability and higher level of abstraction and aggregation properties approximates an FMS. In addition, the Pr/Tr net model offers rich semantic description compared to an ordinary petri net is derived and proposed.

The primary objective of this paper is to gain an understanding of the operational decision problems of an FMS using a hierarchy view, and then modelling the problems for further assistance in operational control planning decision-making. Firstly, this study reviews and evaluates different decision models. Secondly, FMS operations, activities and resources usage are analysed. A mode transition diagram is developed to describe the transition of the activities. Finally, a comprehensive FMS Pr/Tr net model based on the activities of the mode transition diagram and the Pr/Tr net model are integrated. The resulting comprehensive and flexible model is then applied using a rule-based simulation, which revels it is well suited for operational control planning for an FMS.

2 The FMS decision problem

2.1 The hierarchy level of decision problems

In spite of its flexibility, an FMS is highly constrained by its resources such as pallets, fixtures, carts, machines, etc. This constraint makes the planning for FMS extremely difficult. In this section, there is an analysis of the planning work required for FMS and then different modelling tools are evaluated for decision support purposes in the next section.

In regards to the FMS decision problem, Kalkunte et al. [14] present a four-level hierarchical framework for classifying the decision problems, which relate to the design, justification and operational decisions of an FMS. As shown in Fig. 1, an FMS hierarchy decision structure is depicted. The decision outputs at the upper level become the inputs for the lower level. For Level 1, the Strategic Analysis and Economic Justification decision, is related to whether an FMS to be installed or not. Level 2 is the level at which strategic business plans are coalesced into a specific facility design to achieve the long-term objectives. Level 3 encompasses decisions related to master production scheduling and specific issues related to machine loading problems. Level 4 involves dynamic operational minute-to-minute decisions of the FMS. In relation to this, another classification scheme of FMS decision problems have been proposed by Van Loovern [15], categorising FMS decision problems into Strategic, Tactical, and Operational Planning Levels.

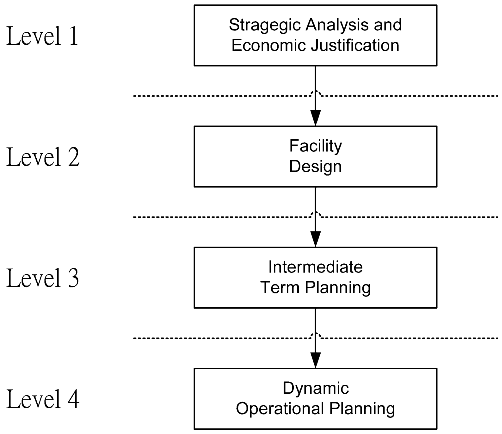

Fig. 1 An FMS hierarchy decision structure

According to the hierarchy decision structure, since the economic justification and facility design were done in the long term planning stage, the decisions remained in levels 3 and 4 impacting the daily operational performance of an FMS in a profound way. Therefore, in this paper, the focus is on level 4, the dynamic operations planning level. A good model for operational control level planning should have the capability to accept the planning outcomes decided in the earlier levels 2 (Facility Design) and 3 (Intermediate Term Planning) as inputs for level 4.

Stecke and Solberg [16] have described in detail operational control planning decisions for FMS. According to their research, different types of decisions that a real-time dispatching system has to make at various points of time can be described as follows:

(1) Select the part to be released into the system.
(2) Select the pallet type to mount a part.
(3) Select the mode of transportation to be used if more than one choice is available.
(4) Select the transportation path to be used to the next workstation.
(5) Select the workstation, among the available list, to perform a requested operation.
(6) Select the part to be processed next, from the input queue at the workstation.
(7) Select the cutting tool to be used to perform an operation.
(8) Select the operator, if necessary, to perform an operation.
(9) Select the next operation to be performed on a part if no predetermined sequences of operation exist.
(10) Select an alternative action under the occurrence of unforeseen situations.

2.2 Factors affecting an FMS operational performance

Many researchers explored dynamic operational problems in an FMS in order to understand its core characteristics. Nof et al. [17] analysed the operational control of the item flow in versatile manufacturing systems. Stecke and Solberg [18] conducted an experimental investigation of operational strategies for a computer-controlled FMS. Denzler and Boe [19] studied the scheduling decision rules for a dedicated FMS. Lie et al. [20] investigated a part type selection problem. Stecke and Kim [21] examined part selection problems. Hutchison et al. [22] proposed an approach for a random job shop flexible manufacturing system. Arzi and Roll [23] studied real time production control of an FMS in order to operate in a customer order environment.

The results of these research studies were inconsistent, but it was shown that system performance depends on what heuristic dispatching rules are used. One control strategy that performs best in an FMS configuration may not be the best for another one. Overall system performance is related to separate, different and unique FMS system elements.

According to the decision framework of an FMS proposed by Kalkunte [14], the factors that affect the system performance for each level are listed in Table 1.

In level 4, a release rule is used to select the next part introduced into an FMS. After a part is entered, further operations will be triggered according to the next task and dispatching rule used. The detailed contents of the dispatching rule used in level 4 includes: selection of pallet type to mount a part, choosing one of the available transportation modes, picking one of available transportation paths to next workstation, choosing one of the parts from the input queue at the work station for machining, selecting the cutting tool, deciding on the operator to perform an operation and finally selecting the next operation for a part if no predetermined sequence of the operation exists. Moreover, decisions about unexpected disturbances, like a machine breakdown, are sometimes necessary. Therefore, the performance function of an FMS can be defined as $PI = F(C, M, Pr, T, Mp, R, D, U)$, where:

PI: FMS performance index
C: System configuration
M: Part mix
Pr: Part ratio
T: Tool assignment
Mp: Machine pooling
R: Release rule
D: Dispatching rule
U: Unexpected disturbance

How to develop an FMS model, which encompasses all of these factors for operational control planning, is quite complex, difficult and important.

Table 1 Factors affecting FMS performance

Level	Factors affecting FMS performance
Level 1	None
Level 2	System configuration
Level 3	Part mix Part ratio Tool assignment Machine pooling
Level 4	Release rule Dispatching rule Unexpected disturbance

2.3 A review of decision models

Several existing decision models have been reviewed before; the power and limits of each model are listed below.

(1) Static allocation model: this is a static and simple model, it ignores all the dynamics, interactions, and various measures of performance, though the static allocation model is easy to implement, it can be too inaccurate and seriously overestimates systems' performance.
(2) Queuing network model: the basic theory of queuing network was developed by Jackson [24], and later on extended by Gordon and Newell [25] and, Buzen [26]. This kind of model accounts for the dynamics, interactions, and uncertainties in the system. But a disadvantage of the queuing network model is that, a set of restrictive assumptions (e.g., exponential processing times, infinite queues) is often required.
(3) Simulation model: this can provide an accurate picture of system performance, but it takes a long time for a model building and data input.
(4) Perturbation analysis: Detailed behaviour of the system is observed—whether through simulation or from the actual system in process for one set of decision parameters. By doing some minor additional calculations while the system is being observed, perturbation analysis can predicate the system behaviour if these decisions are changed. The main disadvantage of this model is that it cannot accurately predicate the effects of large changes in decisions.
(5) Petri net model: this is quite appropriate for modelling dynamic systems. In addiction, it is graphical, readable and easy to understand.

This study shows that system performance depends on the characteristic of each unique FMS, and given the complexities of an FMS, it is clear that analytical, queuing network and perturbation models are not easily adaptable for modelling a unique dynamic FMS. This is especially true because it is necessary to include all the factors in levels 2, 3, and 4 into a model.

Especially, unexpected important factors (like machine breakdown) that impact an FMS performance seriously sometimes needed to be considered. A simulation-based model combined with petri net could be, appropriate for modelling a dynamic manufacturing system. Some other researchers share the same point of view. [27, 28]

---

3 Modelling

3.1 A definition of a Pr/Tr net

A Pr/Tr net consists of the following constituents:

1) A directed graph (P, T, A) where P is the set of predicates ('first-order' places), T is the set of transitions. A is the set of arcs.
2) A structure Σ consisting of some sorts of individual tokens (P_i) together with some operations (OP_j) and relations (R_k), i.e.,

$$\Sigma = \left(P_1, ..., P_i;\ OP_1, ..., OP_j;\ R_1, ..., R_k\right)$$

3) A labelling of all arcs with a formal sum of n attributes of token variables (including the zero-attributes indicating a no-argument token).
4) An inscription on some transitions being a logical formula built from the operations and relations of the structure Σ; variables occurring free in a formula have to occur at an adjacent arc.
5) A marking M of predicates of S with formal sums of n-topples of individual symbols.
6) Firing rule: Each element of T represents a class of possible changes of markings. Such a change, also called transition firing, consists of removing tokens from a subset of predicates and adding them to other subsets according to the expressions labelling the arcs. A transition is enabled whenever, given an assignment of individual tokens to the variables that satisfies the predicate associated with the transition.

An example of Pr/Tr Net is illustrated in Fig. 2a. W, R, U, V are predicates, $<X,Y>$, $<Z>$, $<X>+<Y,Z>$ are labels of formal sum for each arc, t is transition with logical formula $Y<Z$, and predicate tokens $W<a,a>$,$W<a,b>$,$R<c>$ forms the initial markings. In Fig. 2b, t is firable. Figure 2a, after transition t is firing, becomes Fig. 2b.

A Pr/Tr net is a high level petri net. With its core elements defined above, it possesses higher-level abstraction and aggregation properties than ordinary petri net has. A simple Pr/Tr net is illustrated in Fig. 3b to demonstrate its modelling power. As shown in Fig. 3, an ordinary petri net Fig. 3a can be transformed to a concise Pr/Tr net, in Fig. 3b which reduces from 5 places, 2 transitions to 2 places, 1 transition net.

Therefore, it seems plausible that applying this kind of net other than using an ordinary petri net can derive a concise FMS model.

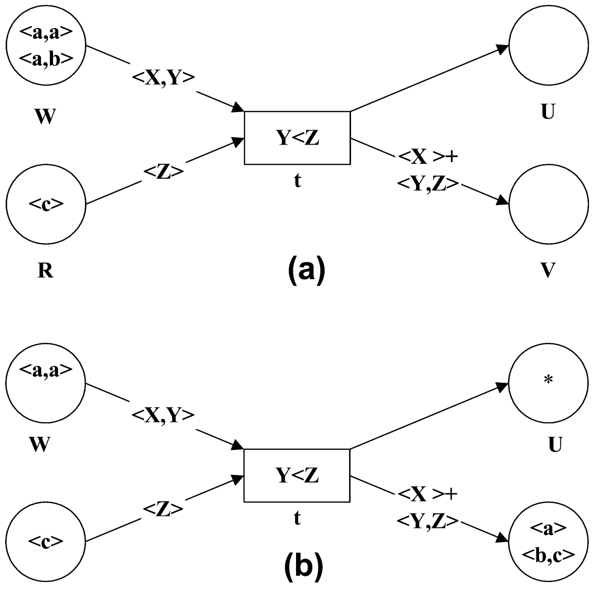

Fig. 2 **a** A Pr/Tr net example **b** A Pr/Tr net example after t is fired

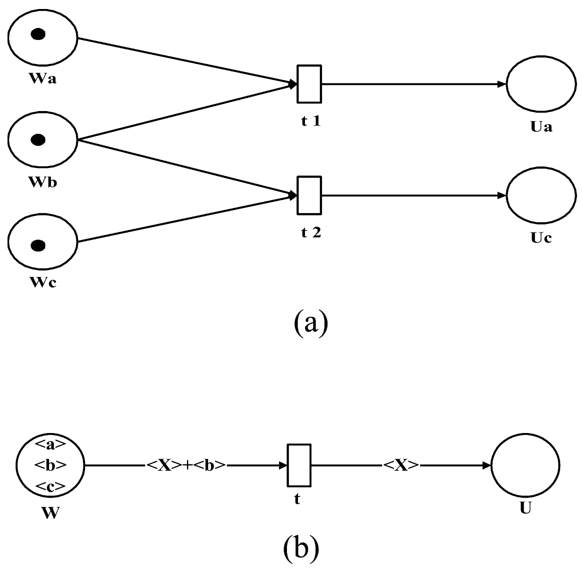

Fig. 3 **a** An ordinary petri net **b** A Pr/Tr net

3.2 The activity definition of an FMS

3.2.1 Activity analysis

In order to understand all activities that occur in an FMS, the processing flow of a part is outlined and an activity mode indicator is assigned to each activity.

1. The part enters FMS from an external storage place or As/Rs (mode 1 activity), if there are some parts waiting at the entrance, then select part by a "release rule".
2. When the part enters the system load-station, the operator sets up pallet and fixture (mode 2 activity).

3. After the part is positioned by the operator, select one of available carts by a "part-select-cart rule", and then shift the part to cart (mode 3 activity).
4. When the part has been shifted to a cart, select one of the available machines which is capable for the next operation by a "part-select-machine" rule and transport the part to the machine selected (mode 4 activity); otherwise, transport the part to the central-buffer (mode 5 activity) for temporary storage. At stage 4, if the part is transported to the in-buffer of the selected machine, then shift the part to the in-buffer by using the machine robot (mode 6 activity). If the part is transported to the central buffer area, then shift the part to buffer using central buffer robot (mode 7 activity).
5. After the part is shifted to machine in-buffer and the machine is free, then load the part to machine (mode 8 activity). If the part is shifted to the central buffer at stage 5, then go to stage 3 and continue stage 7. When a part is loaded on the machine at stage 6 and the operation tool is available, and then do the required machining work (mode 9 activity).
6. If the part's operation work is finished in this machine and the machine out-buffer is available, then unload the part to the machine out-buffer (mode 10 activity). If the operation work is the last machining task for this part, then, after a cart is available and arrives, shift it to the cart and transport it to the system unload-station (mode 11 activity). Otherwise, continue stage 3.
7. When the part is transported to the system unload-station, the system unload-station and the system unload-station robot is available, then shift the part to the system unload-station (mode 12 activity).
8. When the part is shifted to the system unload-station, after it is removed from the pallet and fixture part, the part exits the system (mode 13 activity).

According to the previous operational analysis, the activities which occurred in an FMS are examined and defined by the activity mode indicator and listed in Table 2, a universal activity indicator set, $M=\{1,2,3,4,5,6,7,8,9,10,11,12,13\}$, is acquired.

3.2.2 The mode transition diagram

Based on the flow outline in the preceding section, a mode transition diagram that describes the transition relationship of each activity is shown in Fig. 4. This diagram can detail the FMS behaviours in terms of activity modes and will serve as the basis to apply the Pr/Tr net to an FMS.

3.2.3 The activity set

In this section, an analysis of the resource required and the release for each activity was conducted and defined in Table 3. In Table 3, two functions $F(m)$, $F'(m)$ are used. $F(m)$ is the function that shows the need for resource(s) for a certain activity (mode = m) to perform the operation. $F'(m)$ denotes the function which release(s) the resources back to the FMS after finishing the activity mode = m.

Table 2 The activity mode(m) defined for FMS

Activity mode (m)	Definition
1	Part enters system load-station
2	Part positioned in pallet and fixture
3	Part shifted to cart by system load-station robot
4	Part transported to the machine in-buffer of a machine
5	Part transported to central buffer
6	Part shifted to the machine in-buffer by machine robot
7	Part shifted to the central buffer by central buffer robot
8	Part loaded in the machine by machine robot
9	Machining work performed and completed
10	Finished part shifted to machine out-buffer by machine robot
11	Part transported to the system unload-station
12	Part shifted to the system unload-station
13	Part removed from pallet, fixture and exits the system

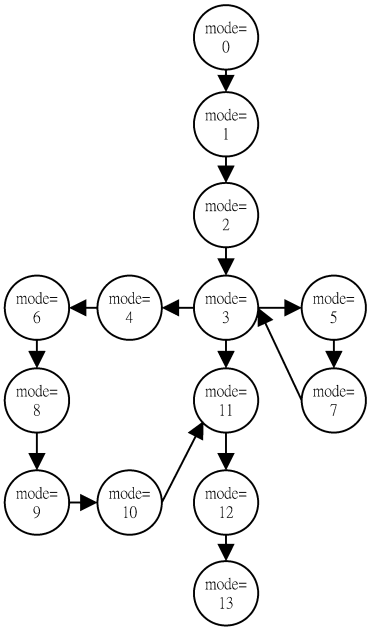

Fig. 4 A mode transition diagram

From Table 3, it can be observed that some activities need resource(s) in order to conduct that activity but some others do not. Some activities release the resource(s) while finishing the operation but some others do not. Activities $m\in\{1,2,3,6,7,8, 9,10,12,13\}$ need resource(s) to support their operations, but for parts going to activity modes 4,5,11 no resource is required to trigger the transportation activity. Because from the activity mode transition diagram a cart is already acquired at the previous activity (activity mode = 3) no additional resource is needed to perform the current transportation activity. After finishing some

Table 3 Resource(s) required and resource(s) release table

Activity mode	Resource(s) required $F(m)$	Resource(s) release $F'(m)$
Mode = 1	System load-station	None
Mode = 2	Operator, pallet, fixture	Operator
Mode = 3	System load-station robot, cart	System load-station robot, or machine out-buffer, or central buffer
Mode = 4	None	None
Mode = 5	None	None
Mode = 6	Machine robot, machine in-buffer	Machine robot, cart
Mode = 7	Central buffer robot, central buffer	Central buffer robot, cart
Mode = 8	Machine robot, machine	Machine robot, machine in-buffer
Mode = 9	Tool	Tool
Mode = 10	Machine robot, machine	Machine robot, machine
Mode = 11	None	None
Mode = 12	System unload-station robot, System unload-station	System unload-station robot, cart
Mode = 13	Operator	Operator, pallet, fixture, system unload-station

activities, $m\in\{2,3,6,7,8, 9,10,12,13\}$, partial or all owned resources must be returned to FMS by function $F'(m)$, but for the activities, $m\in\{1,4,5,11\}$, none are returned. Accordingly, from the resource(s) required and resource(s) release situations, four activity sets can be defined as M_1, M_2, M_3 and M_4.

Resource(s) required activity set: (M_1)

$$M_1 = \{1, 2, 3, 6, 7, 8, 9, 10, 12, 13\}$$

Resource(s) not-required activity set: (M_2)

$$M_2 = \{4, 5, 11\}$$

Resource(s) return activity set: (M_3)

$$M_3 = \{2, 3, 6, 7, 8, 9, 10, 12, 13\}$$

Resource(s) not-return activity set: (M_4)

$$M_4 = \{1, 4, 5, 11\}$$

Note that the relationships

$$M_1 \cup M_2 = M$$

$$M_3 \cup M_4 = M$$

hold and, $M = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13\}$ is the universal activity set.

3.3 The FMS Pr/Tr net model

3.3.1 Modelling by Pr/Tr net

The activity mode, the activity set and the mode transition diagram defined previously will be used to build an FMS Pr/Tr model. First, two predicates, *W* (Want to use resource) and *U* (Using resources), are introduced into the model and transitions with different activity modes are included as shown in Fig. 5a. Next, the activities are aggregated into the activity set M_1 and M_2, and then a concise model is constructed by the activity sets in Fig. 5b. Continuously, introducing another two predicates *F* (Finish using resource) and *R* (Resource available), a refined Pr/Tr net model for FMS is illustrated in Fig. 6.

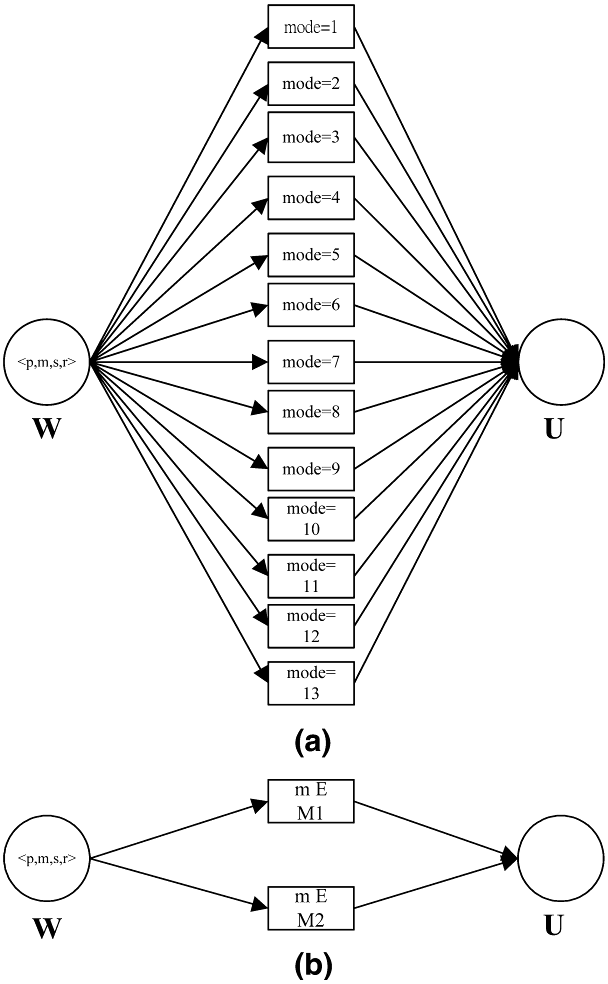

Fig. 5 **a** Transitions with different activity modes **b** A concise model constructed by the activity sets

Additional factors can be used to extend and refine this model. For instance, if a resource, such as a machine, breaks down when the resource is being used to process a part, the part may wait while the machine is being repaired. This creates a new predicate *M* (Repair). Finally, a FMS Pr/Tr Net Model with 7 predicates and 11 transitions is derived as shown in Fig. 7.

3.3.2 A description of the model

The definition of the predicates and transitions in the model in Fig. 7 are tabulated in Tables 4 and 5, respectively; for example, predicate *W* (Want to use

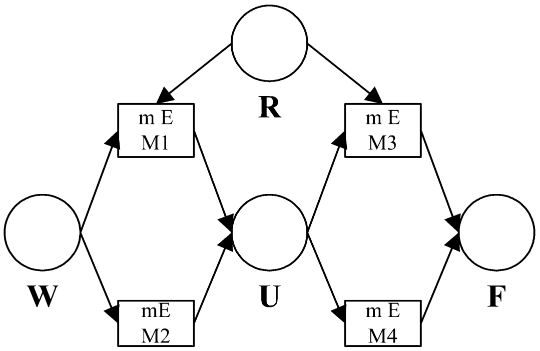

Fig. 6 A refined Pr/Tr net model for FMS

resource) in Table 4 consists of four attributes, *p* (part), *m* (the activity mode), *s* (status of the resource set) and *r* (repair variable when resource or resources break down). Transition T_2 , for instance, move a part from *W* predicate to *U* (Using resource) predicate given that the activity mode belongs to M_1 and the resource set status is "on" ($s=1$). In essence, there are two kinds of predicates in the FMS Pr/Tr Net Model, "activity" predicates (*W*, *U*, *M*) and "state" predicates (*H*, *F*, *E*) (refer to Table 4). An activity predicate may cause a delay in processing of a part. But a state predicate, which only shows state of the part, will not cause any time delay.

A description of the whole operation of the Pr/Tr net model is shown in Fig. 7. Firstly, all parts reside at *H* predicate; all resources such as system load-stations, machines, robots, and carts reside at *R* predicate. The resources constitute the production capacity of the FMS. Secondly, the parts and the resources initially reside at H and R predicates respectively and form the initial markings of the net. When the model begins to

Table 4 Definition of predicates for FMS Pr/Tr net model

Assertion of fact	Predicate	Definition of fact
$H<p>$	*H*	Part *p* appears
$W<p,m,s,r>$	*W*	Part *p* requires activity *m*, *s* is the resources status variable,*r* is repair attribute for the resource breakdown
$U<p,m,s,r>$	*U*	Part *p* is using resource(s) for activity mode = *m*
$F<p,m,s,r>$	*F*	Part *p* activity mode =*m* finished
$R<r>$	*R*	Resource *r* appears
$M<p,m,s,r>$	*M*	Part *p* is in the repaired state, because of a resource(s) availability breakdown
E	*E*	Part *p* all operations completed and exits the FMS system

Table 5 Definition of transitions for FMS Pr/Tr net model

Transition	Definition of transition
T_1	Transits a part to *W* predicate
T_2	Transits a part to *U* predicate (resource(s) required)
T_3	Transits a part to *U* predicate (resource(s) not required)
T_4	Transits a part to *F* predicate (resource(s) return not required)
T_5	Transits a part to *F* predicate (resource(s) return required)
T_6	Transits a resource to *M* predicate
T_7	Transits a part to *H* predicate
T_8	Transits a part to *E* predicate
T_9	Transits a part to *M* predicate
T_{10}	Transits a part to *U* predicate
T_{11}	Transits a resource to *R* predicate

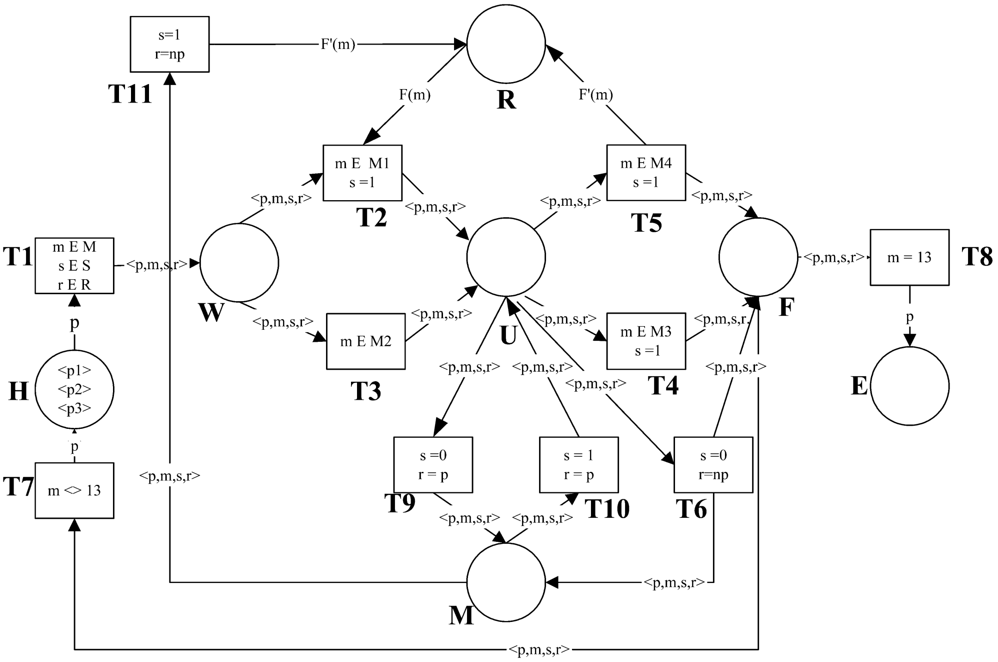

Fig. 7 A FMS Pr/Tr Net Model with seven predicates and 11 transitions

run, all parts (or "tokens") are transited (by transition $T1$) to predicate W with activity mode attribute ($m=1$). At W, if a part, selected by a heuristic rule, acquires all the needed resources allocated by function $F(m)$, transition $T2$ will be fired and the part is transited to U predicate for performing activity m. Resources will be returned to R predicate after this activity is completed. During the usage of resources at U predicate, if some resource(s) breakdown occurs, the part might release the resource(s) and the resource(s) is then transited to M predicate. The part, depending on the repair rule, either stays at M with the broken resource or flows to the F predicate. Then, the part is checked at F predicate to make sure all operations are completed. If yes, the part will be transited to E (Exit) predicate; otherwise, the part will be transited back to H predicate and recycled again for the next activity. When all parts reach E predicate (all parts are finished), the model then concludes.

3.3.3 The definition of transitions and events

An activity creates a pair of events, the beginning event and the ending event, for the activity. An activity predicate may cause a time delay in processing a part. Corresponding to three "activity" predicates W, U, M as defined previously section: WT denotes waiting time of the resource availability, UT denotes the using time and MT denote the repair time for recovery. The time variables WU, BU, FU are used to denote WU(want to use time), BU(beginning use time) and FU(finishing use time) of an activity. The relationship between the activity and the events for predicate U is depicted in Fig. 8. Figure 8a illustrates the relationship on a time axis, while Fig. 8b is in Pr/Tr net notation. Every transition firing means an event occurred in the FMS Pr/Tr Net Model. By defining the events, the FMS Pr/Tr Net can be transformed into a discrete event simulation model.

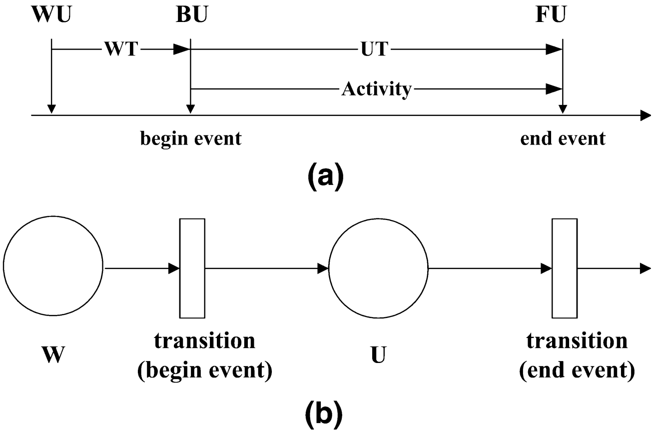

Fig. 8 **a** An activity and event representation on a time axis **b** An activity and event representation in the Pr/Tr Net

There are 11 aggregate transitions and 13 activity modes in the model as shown in the previous section. The symbols

$E(Ti, m)$, for $i = 1$ to 11, $m = 1$ to 13,

are used to denote the event which is associated with transition i and activity mode m. For example $E(T_2,3)$ signifies the beginning event of a part being shifted to a cart by a system load-station robot (see Table 2 and Fig. 7), also, $E(T_5,3)$ denotes the ending event. And $E(T_i,m)$, $i\in\{2,3,10\}$ are beginning events, $i\in\{4,5,6,9\}$ are ending events of an activity. A detailed definition of each event is tabulated in appendix A.

3.3.4 Model flexibility and application

The derived comprehensive model integrates the FMS Pr/Tr net and the mode transition diagram. This model can be used for modelling a different FMS with varying capacity without changing its basic architecture. The system load-station, system unload-station, pallets, fixtures, tools, carts, 0robots, buffers and so on, basically constitute the FMS capacity which can be initially represented by symbolic tokens in this model. Unlimited distinct tokens can be put into the model as needed without reconstituting the structure of the model, which often happens in an ordinary petri net. Simulation scenarios for studying level 4 can be designed. For example, problems relating to release rule, dispatching rule and unexpected machine breakdown can be examined. Below, the lists of the possible and extensible application domains are indicated.

(1) System configuration study: many resource tokens can be put into the model as needed; so, the outcomes can be studied by changing the quantity of the system configuration related tokens. For example, system load-station, system unload-station, pallets, fixtures, tools, carts, robots, machine in-buffer, machine out-buffer, and central buffer, etc.
(2) Intermediate-term problem: given the configuration of an FMS, the effects of part mix, lot size, routing design, tool assignment to a machine, etc. can be analysed.
(3) Dynamic control policy: the impact of different release rules and dispatching rules to the system or the impact of a resource breakdown can be determined.
(4) Dynamic scheduling planning: a schedule plan for each resource within this system by the simulation method can be created.

This model can examine most of the decision problems mentioned by Stecke or Kalkunte, especially the unexpected machine breakdown occurrences factor, which is not easily formulated by using other analytical models.

4 Implementation

4.1 The algorithm for running the FMS Pr/Tr net model

An algorithm for the FMS Pr/Tr net model can be described as follows:

Step 1. Set all parts at predicate place *H* and set all resources at *R* predicate place.

Step 2. Transit all parts to predicate place *W*, all parts at predicate place *W* requesting to perform next activity mode *m*, select a part by a release or dispatching rule and the resource(s) is available.

Step 3. After the activity, transit part to predicate place *F*

Step 4. Parts at predicate place *F* will be processed according to following conditions:

- 4.1 If a part at predicate place *F* with the activity mode 13 (all operations of a part are finished), then transit it to predicate place *E*. If all parts are at predicate place *E*, then stop.
- 4.2 If a part at this predicate place and the activity mode of this part is not mode 13, then transit it to predicate place *H*.

Step 5. Go back to step 2

4.2 The implementation for the mode transition diagram

In step 2 of the algorithm in the previous section defined, whenever a part finishes an activity in FMS, tokens will be transited to *W* predicate for the next activity; the next activity is decided by the mode transition diagram. Figure 9 illustrates a portion of the mode transition diagram. A part token $< p1 >$ with activity mode 1 is represented by a assertion of fact, "mode1(p1)", in the PROLOG system.

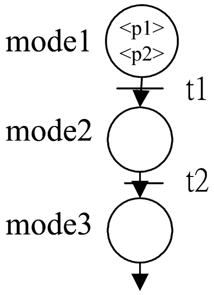

Fig. 9 Transforming the mode transition diagram into the PROLOG language

Example rules for transitions $t1$ and $t2$, which derived by transforming the mode transition diagram into the PROLOG language in Fig. 9, are illustrated in the following section.

```
Transition t1:
modetransition(X):-mode1(Part),
retract(mode1(Part)),
asserta(mode2(Part)).
Transition t2:
modetransition(X):- mode2(Part),
retract(mode2(Part)),
asserta(mode3(Part)).
```

In Transition $t1$, firing a rule simulates a triggering of a transition and reasoning for the next activity, this includes clauses of unmarking the old token (retract mode1(Part)) in the database and inserting another assertion of fact (mode2(Part)) which specifies the next activity to be performed.

4.3 An implementation for the FMS Pr/Tr net model

The implementation of the FMS Pr/Tr net model is simple and straightforward with the algorithm and concept of first order predicate logic. The 11 rules

Table 6 Rules for transitions for FMS Pr/Tr net model

Transition	Rule number	Rule contents
T_1	Rule 1	W(P,M,S,R,WU_Time):−H(Part), modetransition(Part), m∈M, SET(S), S∈{0,1}, SET(R), R∈{np,p}, WU_Time = TIMER(TNOW)
T_2	Rule 2	U(P,M,S,R,BU_Time):-W(P,M,S,R,WU_Time), R(F(M)) S = 1, M ∈ M1, BU_Time = WU_Time + WT
T_3	Rule 3	U(P,M,S,R,BU_Time):−W(P,M,S,R,WU_Time), M∈M2, BU_Time = WU_Time + WT
T_4	Rule 4	F(P,M,S,R,FU_Time):−U(P,M,S,R,WU_Time), S = 1, M∈M3, FU_Time = BU_Time + UT
T_5	Rule 5	R(F′(M)):−U(P,M,S,R,WU_Time), M∈M4, S = 1. F(P,M,S,R,BU_Time):−U(P,M,S,R,WU_Time), M∈M4, S = 1
T_6	Rule 6	M(P,M,S,R):−U(P,M,S,R,BU_Time), S = 0, R = np. F(P,M,S,R,FU_Time):−U(P,M,S,R,BU_Time), S = 0, R = np, FU_Time = BU_Time + (BD_Time-BU_Time)
T_7	Rule 7	H(P):−F(P,M,S,R,FU_Time), M < > 13
T_8	Rule 8	E(P):−F(P,M,S,R)
T_9	Rule 9	M(P,M,S,R):−U(P,M,S,R), S = 0, R = p
T_{10}	Rule 10	U(P,M,S,R):−M(P,M,S,R), S = 1, R = p
T_{11}	Rule 11	R(F′(M)):−M(P,M,S,R), S = 1, R = np

corresponding to 11 transitions in the FMS Pr/Tr net model are derived in Table 6. But, with the concept of simulation, another important factor, "time", must be included. All related time variables, such as *WU_Time*, *BU_Time*, *FU_Time* all of which have been defined in section 4.3.3 except for *BD_Time* used in rule 6, which signifies time for an unexpected breakdown.

Corresponding to Transition 1, Rule 1 is designed to transit a part from H to W predicate, in Rule 1, the modetransition(Part) clause was called to get the next activity mode m, $m \in M$; SET(S) used to indicate resource(s) status for this activity, $S \in \{0,1\}$, and SET(R), indicates if a resource breakdown occurs whether not to preempt resource(s) ($R = p$) or not ($R = np$), so $R \in \{np, p\}$. For triggering transition 2, preconditions $W(P, M, S, R, WU\_Time)$, $R(F(m))$, $S = 1, M \in M1$ for Rule 2 must all be satisfied. $R(F(m))$ can be specified and states the resource(s) which are needed for triggering activity m. Similarly, $R(F'(m))$ indicates the resources returned back to the system in Transition 5.

In this program scheme, there are $H(p1)$, $H(p2)$, $H(p3)$ as part tokens, and $R(m1)$, $R(m2)$, $R(m3)$ as resource tokens which form the initial markings. Combining with other system configuration data and the 11 rules, these tokens can drive the model toward the goal.

System performance measures can be established, such as mean throughput time, utilisation of resources, job tardiness or makespan for the results of different heuristic rules that were chosen for each run. The help for operational control decision planning could be reached.

5 Conclusions

This paper focused on FMS decisions using a hierarchical perspective. There are so many factors that may affect the system performance and it seems quite difficult to formulate all of these factors into one analytical model. A petri net is good for modelling a dynamic system, but it becomes unmanageable if the system is too complex and large. A coloured petri net gives colours to tokens to empower its modelling capability but it does not provide semantic meaning for the net. Therefore a higher-level petri net, the Pr/Tr net, is used as a comprehensive modelling tool for FMS operational control planning. By changing the parameters, an examination can be made of the factors that might impact an FMS and provide assistance for operational control decision-making.

There may be an FMS whose operational activities are different from the one introduced here. However, it is quite easy to modify and adapt. The mode transition diagram only needs to be changed in order to adapt to the specific system without changing the FMS Pr/Tr net architecture. The adaptive capability is extensive.

In addition to net properties, the Pr/Tr net includes first-order predicate logic to treat individual tokens and their changing properties and relations [29]. Interestingly enough, Giordana and Saitta [30] used a Pr/Tr net to model production rules. Also, Murata and Zhang [31] applied a Pr/Tr net model to parallel interpretation of logic programs. All these studies indicated the feasibility of implementation of the model by logic languages. The implementation of this FMS Pr/Tr net model was made possible by the use of a well-known logic language—PROLOG.

A decision support system, which was comprised of three components: the UI (User Interface) module; the FMS Pr/Tr net simulation module and the database module, was developed. This system provides an excellent basic foundation for studying sets of heuristic rules for complex operational control planning for an FMS.

Appendix A. Events list for FMS Pr/Tr net model

$E(T_1,m)$	Events of part want to perform an activity mode $= m$ ($m \in [1...13]$)
$E(T_2,m)$	Begin events of a part perform an activitity mode $= m$ ($m \in [1...13]$)
$E(T_2,1)$	Begin event of a part enters system load-station
$E(T_2,2)$	Begin event of a part sets up pallet and fixture
$E(T_2,3)$	Begin event of a part shifted to cart
$E(T_2,4)$	Begin event of a part transported to machine buffer
$E(T_2,5)$	Begin event of a part transported to central buffer
$E(T_2,6)$	Begin event of a part shifted down to machine buffer
$E(T_2,7)$	Begin event of a part shifted down to central buffer
$E(T_2,8)$	Begin event of a part loaded on machine
$E(T_2,9)$	Begin event of a part machining
$E(T_2,10)$	Begin event of a part unloaded from machine
$E(T_2,11)$	Begin event of a part transported to system unload-station
$E(T_2,12)$	Begin event of a part shifted down to system unload-station
$E(T_2,13)$	Begin event of a part positioned pallet, fixture and exit the system
$E(T_3,m)$	Begin events of a part performs a activity mode $= m$
$E(T_3,4)$	Begin event of a part transported to machine in-buffer
$E(T_3,5)$	Begin event of a part transported to central buffer
$E(T_3,11)$	Begin event of a part transported to system unload-station
$E(T_4,m)$	End event of a part finishes activity mode $= m$
$E(T_4,1)$	End event of a part enters system load-station
$E(T_4,2)$	End event of a part positioned in pallet and fixture
$E(T_4,4)$	End event of a part transported to machine in-buffer
$E(T_4,5)$	End event of a part transported to central buffer

Appendix (Contd.)

$E(T_4,11)$	End event of a part transported to system unload-station
$E(T_5,m)$	End event of a part finishes activity mode $= m$
$E(T_5,3)$	End event of a part shifted to cart
$E(T_5,4)$	End event of a part transported to machine in-buffer
$E(T_5,6)$	End event of a part shifted to machine in-buffer
$E(T_5,7)$	End event of a part shifted to central buffer
$E(T_5,8)$	End event of a part loaded on machine
$E(T_5,9)$	End event of a part machining
$E(T_5,10)$	End event of a part unloaded from machine
$E(T_5,12)$	End event of a part shifted to system unload-station
$E(T_5,13)$	End event of a part removed pallet and fixture
$E(T_6,m)$	Events of part returns of resource(s) for mode m ($m\in$ [1...13])
$E(T_6,1)$	Event return resource(s) to the system when resource(s) for activity 1 breaks down
$E(T_6,2)$	Event return resource(s) to the system when resource(s) for activity 2 breaks down
$E(T_6,3)$	Event return resource(s) to the system when resource(s) for activity 3 breaks down
$E(T_6,4)$	Event return resource(s) to the system when resource(s) for activity 4 breaks down
$E(T_6,5)$	Event return resource(s) to the system when resource(s) for activity 5 breaks down
$E(T_6,6)$	Event return resource(s) to the system when resource(s) for activity 6 breaks down
$E(T_6,7)$	Event return resource(s) to the system when resource(s) for activity 7 breaks down
$E(T_6,8)$	Event return resource(s) to the system when resource(s) for activity 8 breaks down
$E(T_6,9)$	Event return resource(s) to the system when resource(s) for activity 9 breaks down
$E(T_6,10)$	Event return resource(s) to the system when resource(s) for activity 10 breaks down
$E(T_6,11)$	Event return resource(s) to the system when resource(s) for activity 11 breaks down
$E(T_6,12)$	Event return resource(s) to the system when resource(s) for activity 12 breaks down
$E(T_6,13)$	Event return resource(s) to the system when resource(s) for activity 13 breaks down
$E(T_8,13)$	Event of a part exits system
$E(T_9,m)$	Begin events for repairing resource(s), activity mode $=m$ ($m\in$[1...13])
$E(T_{10},m)$	End events for repairing resource(s), activity mode $=m$ ($m\in$[1...13])

References

1. Suri R (1983) An overview of evaluative models for Flexible Manufacturing Systems. Ann Oper Res 8–15
2. Solberg J (1977) A mathematical model of computerized manufacturing systems. In: Proceedings of the 4th International Conference on Production Research, Tokyo, Japan, August 1977
3. Suri R, Hildbrant RR (1984) Modeling flexible manufacturing systems using mean-value analysis. J Manufact Sys 3(1):27–38
4. Shukla CS, Chen FF (1996) The state-of-the-art in intelligent real-time FMS control: a comprehensive survey. J Intellig Manufact 7:441–455
5. Narahari Y, Viswanadham N (1985) A petri net approach to the modeling and analysis of flexible manufacturing. Ann Oper Res 3:449–472
6. Valette R, Courvoisier M, Demmou H, Bigou JM, Desclaux C (1985) Putting petri nets to work for controlling flexible manufacturing systems. In: Proceedings of the International Symposium on Circuit Systems, Kyoto, Japan, 1985
7. Lee DY, Uzsoy R, Martin-Vega LA (1994) Scheduling FMS using petri nets and heuristic search. IEEE Trans Rob Automat 10:123–132
8. Reyes A, Yu G, Kelleher SL (2002) Integrating petri nets and hybrid heuristic search for the scheduling of FMS. Comp Indust 47:123–128
9. Reyes A, Yu H, Kelleher G (2000) Advanced scheduling methodologies for flexible manufacturing systems using petri nets and heuristic search. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2000), San Francisco, CA, April 2000
10. Murata T (1989) Petri nets: properties, analysis and application. Proceedings IEEE 77(4)
11. Gentina JC, Corbeel D (1987) Colored adaptive structured petri nets: A tool for the automatic synthesis of hierarchical control of FMS. In: Proceedings of the International Conference of Robotics Automation, Raleigh, NC, April 1987
12. Alla H, Ladet P (1986) Colored petri nets: a tool for model validation and simulation of FMS. In: Flexible manufacturing systems: methods and studies. North-Holland, The Netherlands
13. Genrich HJ, Lautenbach K (1986) System modelling with a high-level petri net. Theor Comp Sci 13:109–136
14. Kalkunte MV et al. (1986) Flexible manufacturing systems: a review of modeling approaches for design, justification and operation. In: Manufacturing systems: method and studies, pp 3–25
15. Van Looveren AJ et al.(1986) A review of planning models. In: Modelling and design of flexible manufacturing systems, pp 3–31
16. Stecke KE, Solberg JJ (1981) Loading and control policies for a flexible manufacturing system. Int J Prod Res 19(5):481–490
17. Nof SY et al. (1979) Operational control of item flow in versatile manufacturing systems. Int J Prod Res 17(5):479–489
18. Solberg JR (1977) An overview of evaluative models for flexible manufacturing systems. In: Proceedings of the 4th International Conference on Production Research, Tokyo, Japan, August 1977
19. Denzler DR, Boe WJ (1987) Experimental investigatigation of flexible manufacturing system scheduling decision rules. Int J Prod Res 25(7):979–994
20. Liu CY, Liu LL, Yih TY (1995) A framework for part type selection and scheduling in FMS environment. Int J Comp Integ Manufact 8:102–118
21. Steck KE, Kem I (1991) A flexible approach to part type selection in flexible flow systems using part mix ratios. Int J Prod Res 29:53–75
22. Hutchison J, Leoong, Snyder K, Ward D (1991) Scheduling approaches for random job shop flexible manufacturing systems. Int J Prod Res 29(5):1053–1067
23. Arzi Y, Roll Y (1993) Dispatching procedures for a flexible manufacturing cell in constant production circumstance. Int J Oper Prod Manage 13(11):35–51
24. Jackson J (1957) Networks of waiting lines. Oper Res 5
25. Gordon W, Newell G (1967) Closed queuing systems with exponential servers. Oper Res 15
26. Buzen J (1973) Computational algorithms for closed queuing networks with exponential servers. Comm ACM 16
27. Park T, Lee H, Lee H (2001) FMS design model with multiple objectives using compromise programming. Int J Oper Res 39(15):3513–3528

28. Chan FTS, Chan HK, Kazerooni A (2002) A fuzzy multi-criteria decision-making technique for evaluation of scheduling rules. Int J Adv Manuf Technol 20:103–113
29. Genrich JJ (1987) Predicate/transition nets. Lec Not Comp Sci 205:207–247
30. Giordana A, Saitta L (1985) Modeling production rules by means of predicate/transition networks. Info Sci 35:1–41
31. Murata T, Zhang D (1988) A predicate/transition net model for parallel interpretation of logic programming. IEEE Trans Soft Engin 14(4):481–497