'91 KACC 1991. 10. 22~24

Discrete Event Systems Modeling and Scheduling of Flexible Manufacturing Systems

Hiroyuki Tamura and Itsuo Hatono

Department of Precision Engineering
Osaka University
2-1 Yamada-oka, Suita, Osaka 565, Japan

Fax: +81-6-878-3819
e-mail: tamura@prec.osaka-u.ac.jp
hatono@prec.osaka-u.ac.jp

ABSTRACT

In this paper we describe Flexible Manufacturing Systems (FMS) using Petri nets, since Petri nets provide a powerful tool for modeling dynamical behavior of discrete concurrent processes. We deal with off-line and on-line rule-based scheduling of FMS. The role of the rule-base is to generate appropriate priority rule for resolving conflicts, that is, for selecting one of enabled transitions to be fired in a conflict set of the Petri nets. This corresponds to select a part type to be processed in the FMS.

Towards developing more Intelligent Manufacturing Systems (IMS) we propose a conceptual framework of a futuristic intelligent scheduling system.

Key words. Discrete event system; Petri net; FMS (Flexible Manufacturing System); rule-based scheduling; intelligent scheduling; IMS (Intelligent Manufacturing Systems).

1. INTRODUCTION

The main objective of Flexible Manufacturing Systems (FMS) is to adapt to various changes of external factors such as social/economic environment of production, customers various and varying preferences and demands, and so forth. Highly computerized FMS have been developed to meet the requirement of time-varying-variety/time-varying volume production. The present and the future manufacturing systems are asked to take into account technology assessment for maintaining global natural environment. In such circumstance of manufacturing, job schedulings are the important tasks for achieving various objectives of FMS.

This paper summarizes our recent works [1,2,3,4,5] and their extensions on modeling and scheduling methodologies for FMS as typical discrete event dynamic systems [6]. The events in FMS are such as loading, processing, transporting, unloading, and so forth. The behavior of FMS is so complex that it is difficult to analyze it theoretically. Thus, simulation techniques for FMS [7,8] are necessary for design, performance evaluation, production planning, and production scheduling of FMS.

We use Petri nets for modeling and simulating FMS. Timed Petri nets [9] are used for the purpose of off-line scheduling, and timed-stochastic Petri nets [10,11] for on-line scheduling. We construct a rule base to generate a priority rule whenever we need to resolve conflicts, that is, to select one of enabled transitions in a conflict set of the Petri net. In FMS this corresponds to select a part type to be processed.

Towards developing more Intelligent Manufacturing Systems (IMS) we describe a conceptual framework of a futuristic intelligent scheduling system.

2. MODELING OF FMS USING TIMED/STOCHASTIC PETRI Nets

In this section we summarize [1,3], where we deal with an FMS with fixed routings so that sequences of machines to process are given for each part type.

2.1 Deterministic Modeling Using Timed Petri Nets

Let $P=\{p_1,p_2,\dots,p_n\}$ and $T=\{t_1,t_2,\dots,t_m\}$ be finite sets of places and transitions, respectively. The input and output functions relate the transitions and places. The input function is a mapping from a transition t_i to a set of input

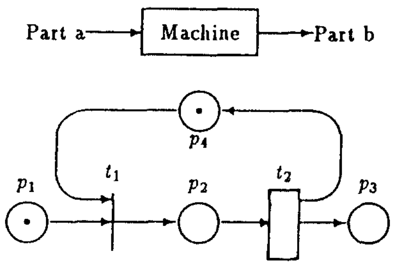


Fig. 1. Petri net model of an FMS element.

places $I(t_i)$ of the transicion. The output function O is a mapping from a transition t_i to a set of output places $O(t_i)$ of the transition. In timed Petri nets [9], the transitions are classified into two types: immediate transiions that can fire immediately after they become enable, and timed transtions that can fire some fixed time after they become enable. This time is called firing time.

Let μ be a marking and $L(p_j)$ be the upper bound of the number of tokens that the place p_j can contain. Then, the firing conditions of enabled transitions are defined as follows:

Firing Condition: A transition t_i is enabled when a marking μ holds the following two conditions

1. $\mu(p_j) > 0$ for all $p_j \in I(t_i)$
2. $\mu(p_j) < L(p_j)$ for all $p_j \in O(t_i)$,

where

$$L(p_j)=\begin{cases} \text{buffer capacity, if } p_j \text{ is a buffer place} \\ 1, \quad \text{otherwise.} \end{cases}$$

Firing a transition will change the marking according to the following evolution rule.

Evolution Rule: When a transition t_i is enabled at time τ with a marking μ, firing t_i results in a new marking μ' as

1. $\mu'(p_j)=\mu(p_j)-1$ for all $p_j \in I(t_i)$
2. $\mu'(p_j)=\mu(p_j)+1$ for all $p_j \in O(t_i)$.

An FMS is regarded as a discrete event dynamic system whose elements are such as loading, processing, unloading, and so forth.

Figure 1 shows a simplest possible example of a Petri net representation for an FMS element. In Fig. 1 transitions t_1 and t_2 represent events of operational start and operational end, respectively, where t_1 denotes an immediate transition and t_2 denotes a timed transition. Places p_1 and p_3 represent buffers for loading and unloading, respectively. Places p_2 and p_4 are referred to as processing and waiting places, respectively. A token in place p_2 represents the state that a part is under processing by the machine. A token in place p_4 represents the state that the machine is waiting for a part to be processed. Tokens in places p_1 and p_3, respectively, represent parts in each buffer. An FMS is modeled by the combinations of this kind of Petri net representations, and we can simulate the state of the FMS in each time step by tracing the distribution of tokens in the timed Petri net.

2.2 Stochastic Modeling Using Timed-Stochastic Petri Nets

In timed-stochastic Petri nets [10,11], the firing time of a timed transition is a random variable with an appropriate probability distribution. By using such stochastic Petri nets we can model an FMS under uncertainty as a discrete event dynamic system whose elements are such as loading, processing, unloading, occurrence of machine failures, repairing of machine tools, and so forth.

When we deal with an FMS under uncertainty, the FMS model is difficult to simulate by using one Petri net model because of the increase of the number of places and transitions. To cope with this difficulty, hierarchical stochastic Petri nets are used for modeling FMS under uncertainty, where an FMS model is partitioned into several submodels, such as a transporting level model, a processing level model, a control level model, and so forth.

The transporting level model describes a flow of workpieces and mutual exclusion control of machine tools or transports (automated guided vehicles, AGV). The processing level model describes the processing of a workpiece with machine tools, repairing of a troubled machine (machine tool or transport), and so forth. The control level model describes the data processing on computers which control machine tools and transports. To reduce the complexity of the model of such hierarchical systems, the hierarchical structure of the system must be described in stochastic Petri nets.

Figure 2 shows an example of hierarchical stochastic Petri net representation for an FMS where one part type is processed with one machine tool. We introduce the following conditions in the hierarchical stochastic Petri net in Fig. 2.

1) The firing time of each timed transition in Fig. 2(b) is obtained using a random number associated with the following probability distribution.

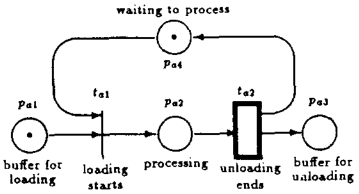
t_{b2}: a normal distribution with mean of the processing time determined in the process planning and variance σ^2.

t_{b4}: an exponential distribution with mean $1/\gamma$, where γ denotes failure rate.

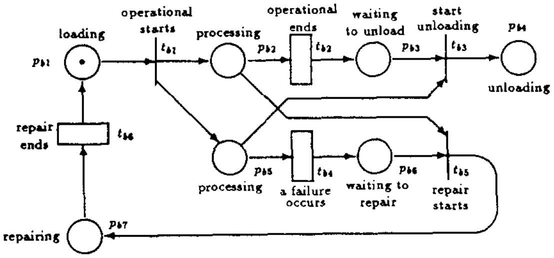
t_{b6}: an exponential distribution with mean r, where r denotes mean repair time.

If the negative firing time is obtained, it is assumed to be zero.

2) In Fig. 2(a), we call the transition t_{a2} the hierarchical timed transition and the firing time of t_{a2} is obtained as the time spent by a token to move from place p_{b1} to p_{b4} in Fig. 2(b).



(a) An example of transporting level model (Discrete-time stochastic Petri nets).



(b) An example of processing level model (Continuous-time Stochastic Petri nets).

Fig. 2. A hierarchical stochastic Petri net model.

A token in places p_{a1}, p_{a2}, p_{a3} in Fig. 2(a) and p_{b1}, p_{b2}, p_{b3}, p_{b4} in Fig. 2(b) represents the state of a workpiece under operation. Therefore, the processing level model shown in Fig. 2(b) plays a submodel of the transporting level model shown in Fig. 2(a). Moreover, all the processing level model of any process in FMS, can be represented by the stochastic Petri nets that has the same connective relations of places and transitions as shown in Fig. 2(b) with different parameter values of mean and variance of probability distributions associated with firing times. Consequently, even when an FMS with many part types and processes is modeled, we need only one processing level model. This will simplify the overall model enormously.

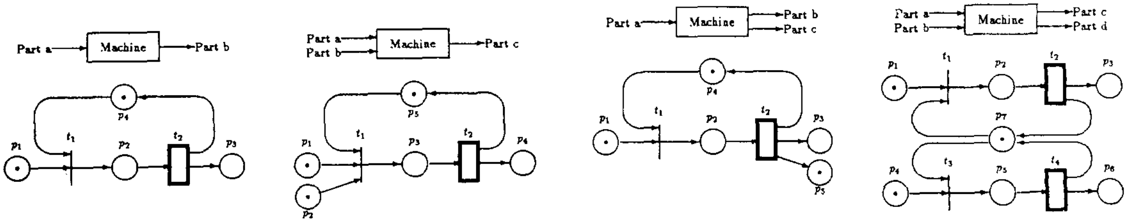
Figure 3 shows typical examples of transporting level models of four basic FMS elements. Each element contains a hierarchical timed transition as shown by a thick white box, and hence contains a processing level model as shown in Fig. 2(b) as its submodel. These four basic stochastic Petri net representations are combined to model a transporting level model of an FMS.

2.3 Conflicting Transitions

In Fig. 3(d), transitions t_1 and t_3 are enabled. If transition t_1 fires then transition t_3 becomes unabled and vise versa. We identify a set of transitions in such situation as follows:

Conflicting Transitions: A set of enabled transition is conflicted at time τ, if, for every pair of transitions t_i and t_j in the set, t_j becomes unabled when t_i is fired at time τ, and vice versa.

In the timed/stochastic Petri net model of an FMS with fixed routings, conflicted sets of



(a) A transfer line. (b) An assembly line. (c) A disassembly line. (d) A transfer line with two part types.

Fig. 3. Transporting level models of four basic FMS elements.

enabled transitions are disjoint and uniquely identified as sets of operational-start transitions with the common waiting places. To resolve conflicts, we need to select one of enabled transitions to be fired in each conflicted set. Since each transition in a conflicted set corresponds to each part type, we can resolve conflicts by applying priority rules to select a part type. An appropriate priority rule for resolving a conflict may change over time and may depend on the different conflicted set.

3. RULE-BASED FMS SCHEDULING

This section summarizes [1,4], where we deal with an off-line and on-line rule-based scheduling system for generating appropriate priority rules to select a transition to be fired from a set of conflicting transitions according to the states of the timed/stochastic Petri net model of an FMS.

3.1 Constructing a Rule Base

For generating an appropriate priority rule, we construct a hierarchically structured rule base. Each rule in the rule base fas the following form:

if P_i then Q_i.

A condition P_i is given by a predicate which describes the states of the Petri net such as throughput, remaining number of processes, remaining time up to the due date, remaining processing time, occupation ratio in each buffer, and so forth. An action statement Q_i is either a priority rule or a group number of the rules to be checked next.

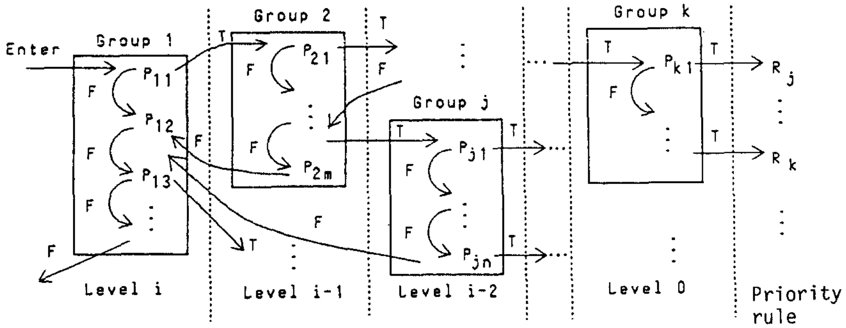
A hierarchical structure of the rule base is shown in Fig. 4 [1]. The procedure to check whether each predicate P_{ij} in group i is true or false is as follows: In the first stage, the first predicate P_{11} in group 1 is checked. If it is true then the predicate P_{21} is examined next as shown by a directed arc T; otherwise P_{12} in group 1 is examined as shown by a directed arc F, where arc T and F denote true and false, respectively. In general, after checking a predicate P_{ij} in group i in this hierarchical rule base, if it is true then the first predicate in the group of lower level followed by an arc T is checked; otherwise the next predicate in the same group followed by an arc F is checked. If the last predicate in a group is false, then a predicate in the group of the upper level followed by an arc F is checked. If a predicate in the group of level 0 is true then the corresponding priority rule is selected for resolving conflicts of enabled transitions.

Job scheduling is carried out under various objectives such as minimizing completion time, flow time, number of tardy jobs, maximizing production rate, and so forth. We need to find an appropriate rule base depending upon the different scheduling objectives. An example of the rule base for on-line scheduling with the major objective JIT (Just-in-Time) can be found in [4].

3.2 FMS Scheduling System

A rule-based scheduling system consists of three subsystems such as an FMS modeling system, an FMS simulator, and a rule interpreter, besides the rule base and the data base. The FMS modeling system constructs a timed Petri net model of an FMS with fixed routings taking into account the machining informations.

The FMS simulator conducts simulation run



P_{ij}: Predicate

Fig. 4. Hierarchical structure of a rule base.

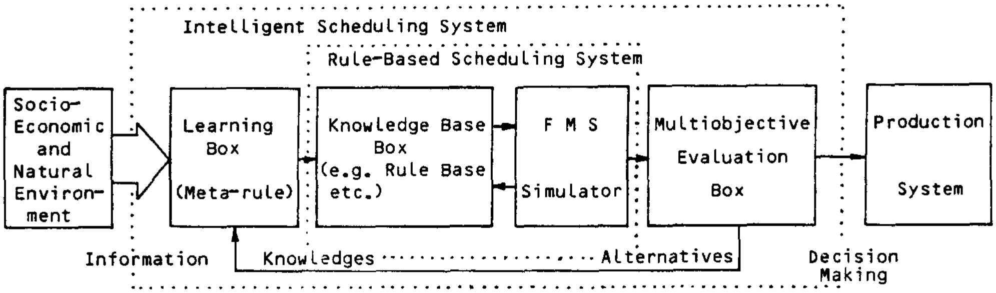


Fig. 5. A conceptual framework of an intelligent scheduling system.

for each time step as follows:

Step 1. Search enabled transition at time t.

Step 2. If there exist conflicting enabled transitions, then call the rule interpreter; otherwise fire all the enabled transitions.

Step 3. If the present time t is the final time for simulation, then stop; otherwise set t+1 t and go to Step 1.

The rule interpreter generates a priority rule from the rule base when the FMS simulator detects conflict among enabled transitions. According to the priority rule generated, one enabled transition is chosen to be fired and the conflict is resolved.

4. TOWARDS INTELLIGENT SCHEDULING

A conceptual block diagram of a futuristic intelligent scheduling system is shown in Fig. 5 [12]. This system consists of four blocks as

1) Learning box
2) Knowledge base box
3) FMS simulator
4) Multiobjective evaluation box.

The "learning box" is to behave as a meta-rule which generates an appropriate rule base depending upon the time-varying multiattribute utility function [2,13] identified in the "multiobjective evaluation box". The multiattribute utility function identified represents the preference of the decision maker who manages the manufacturing system concerned. The preference may vary in time depending upon the socio-economic and natural environment of manufacturing, preference of the customers, the demand for various products, and so forth.

The "knowledge base box" and "FMS simulator" generate production schedules as discussed in the previous sections, where the rule base generated by the "learning box" is implemented in the "knowledge base box".

In the "multiobjective evaluation box" the production schedule generated by the "knowledge base box" and "FMS simulator" is evaluated whether the schedule meets the objective of the decision maker or not. If not, this information is fed back to the "learning box" to modify the meta-rule for generating an appropriate rule base.

Realization of an effective "learning box" is the central problem to obtain an intelligent scheduling system of an FMS. As an introductory research towards this direction, a self-tuning mechanism of rules in rule-based scheduling of FMS is being developed [14]. This mechanism can adjust some heuristic parameters in the rule base automatically analyzing the scheduling results obtained previously.

Another approach towards intelligent scheduling for realizing scheduling objectives with high variety, we propose an application of fuzzy inference [5]. In this approach we devide the knowledge base into 3 modules; schedule evaluation module, scheduling policy determining module, and dispatching module, each of which is described by fuzzy rules. Depending upon the change of scheduling objectives user can easily correct the knowledge base by revising the membership functions of the fuzzy rules in the schedule evaluation module.

5. CONCLUDING REMARKS

In this paper we have summarized our recent works on modeling and scheduling methodologies for FMS which are the typical discrete event dynamic systems. It is shown that timed/stochastic Petri nets provide a powerful tool for modeling such systems.

In flexible manufacturing JIT (Just-in-Time) production and TQC (Total Quality Control) are quite important [15,16] in which the in-process inventory and the inventory of the final products are tried to decrease as much as possible. At the same time, saving the necessary resources and energy as much as possible and production of high quality products and services with high variety are expected.

For production of the next generation, we should take into account the technology and environmental assessment of the concerning production in the global sense. For the environmental assessment of production, only the industrial and factory wastes have been taken into account. This consideration is not enough in future. We need to take into account the environmental assessment of consumption and household wastes as well before we start to produce the concerning products. In other words we need to develop new products which are gentle for our globe, that is, we are obliged to realize "sustainable development". For this purpose, Intelligent Manufacturing Systems (IMS) including intelligent job schedulings are highly expected to contribute.

REFERENCES

[1] Y. Nakamura, I.Hatono, Y.Kohara, K. Yamagata and H. Tamura, "FMS Scheduling Using Timed Petri Net and Rule Base," Proc. 2nd USA-Japan Symp. on Flexible Automation, Minneapolis, July 18-20, 1988

[2] H. Tamura, K. Matsubayashi and Y. Nakamura, "Modeling Multiattribute Preferences for Evaluating Production Schedules in Flexible Manufacturing Systems," Proc. 2nd USA-Japan Symp. on Flexible Automation, Minneapolis, July 18-20, 1988.

[3] I. Hatono, N.Katoh, K.Yamagata and H.Tamura, "Modeling of FMS under Uncertainty Using Stochastic Petri Nets," Proc. Int. Workshop on the Petri Nets and Performance Models, pp. 122-129, Kyoto, Japan, Dec. 11-13, 1989.

[4] I. Hatono, K. Yamagata and H. Tamura, "Modeling and On-Line Scheduling of Flexible Manufacturing Systems Using Stochastic Petri Nets," IEEE Trans. Software Eng., Vol. 17, No. 2, pp. 126-132, 1991.

[5] I.Hatono, T.Suzuka, K.Yamagata and H.Tamura, "Scheduling of Flexible Manufacturing Systems Using Fuzzy Inference - Realization of Various Scheduling Objectives -," (in Japanese) J. Japan Soc. for Fuzzy Theory and Systems, Vol. 3, No. 2, pp. 339-346, 1991.

[6] P. Varaiya and A. B. Kurzhanski, eds., "Discrete Event Systems - Models and Applications -," Lecture Notes in Control and Information Sciences 103, Springer, Berlin, 1987.

[7] S.S. Panwalker and W. Iskander, "A Survey of Scheduling Rules," Operations Research, Vol. 25, No. 1, pp. 45-61, 1977.

[8] J.H. Blackstone Jr., D. T. Phillips and G.L. Hogg, "A State-of-the-Art Survey of Dispatching Rules for Manufacturing Job Shop Operations," Int. J. Production Research, Vol. 20, No. 1, pp. 27-45, 1982.

[9] J. Sifakis, "Use of Petri Nets for Performance Evaluation," In H.Beilner et al. eds.: Proc. 3rd Int. Symp. on Modelling and Performance Evaluation of Computer Systems, North-Holland, Amsterdam, 1977.

[10] M.A. Marsan, G. Conte and G. Balbo, "A Class of Generalized Stochastic Petri Nets for Performance Evaluation of Multiprocessor Systems," ACM Trans. Computer Systems, Vol. 2, No. 2, pp. 93-122, 1984.

[11] F. Archetti and A. Sciomachen, "Representation, Analysis and Simulation of Manufacturing Systems by Petri Net Based Models," In [6], 1987.

[12] H. Tamura, K. Yamagata and I. Hatono, "Decision Making for Flexible Manufacturing - OR and/or AI Approaches in Scheduling -," Syst. Anal. Model. Simul., Vol. 6, No. 5, pp. 363-371, 1989.

[13] J.R. Canada and W.G. Sullivan, "Economic and Multiattribute Evaluation of Advanced Manufacturing Systems," Prentice Hall, Englewood Cliffs, N.J., 1989.

[14] I. Hatono, K. Tachibana, K. Yamagata and H. Tamura, "Rule-based Scheduling of Flexible Manufacturing Systems with Self-tuning Mechanism," (in Japanese), Proc. 34th Conf. of ISCIE, pp.493-494, Kyoto, May 16-18, 1990.

[15] Y. Monden, "Toyota Production System," IIE, 1983.

[16] M. Ebrahimpour and R. J. Schonberger, "The Japanese Just-in-Time/Total Quality Control Production System, Int. J. Production Research, Vol. 22, No. 3, pp. 421-430, 1984.