Chung-Hsien Kuo
Graduate Student.

Han-Pang Huang
Professor.

Kuang C. Wei
Ford Research Laboratory

Steve S. H. Tang
Ford Lio-Ho Motor Co.

Robotics Laboratory, Department of
Mechanical Engineering,
National Taiwan University, Taipei, Taiwan
10674, R.O.C.,
TEL/FAX: (886)2-3633875,
e-mail: hphuang@w3.me.ntu.edu.tw

System Modeling and Real-Time Simulator for Highly Model-Mixed Assembly Systems

In recent years, the automobile industry is facing more challenge than ever due to rapid model change requirements and demand fluctuations. Traditional fixed-type production systems are less capable in coping with the rapid change, while flexible production systems can better meet the requirement of low-volume production with model variations. Flexible machines integrated with model-mixed assembly lines form flexible assembly systems (FASs). In this paper, Colored Timed Petri Net (CTPN) is used to model a highly model-mixed automobile assembly system. Based on this CTPN model, a real-time simulator was developed along with different dispatching rules. Besides, the algorithm of balanced budget work standard (BWS) for a small batch production is developed for adjusting the loading balance of the operators in the highly model-mixed assembly lines. The simulator can be used as an effective tool for developing scheduling strategies.

Introduction

As the global marketplace gets more competitive, product life cycle becomes shorter and product variety wider. An efficient FAS which can cope with the market requirements is highly desirable [14]. Various modeling tools are often used to analyze the performance of the production system. In order to achieve high productivity and quality, several modeling approaches have been used to analyze the efficiency of manufacturing systems, e.g. queuing theory, finite-state machines, perturbation analysis, and Petri Net (PN). Among these, PN is a powerful tool for modeling a complex system with concurrent and asynchronous properties [2, 13]. In this study, the Colored Timed Petri Net (CTPN) [9, 10] provides the additional color and time properties for the original PN, and they can describe the attributes of resource and time properties. Therefore, the CTPN is suitable for modeling the complex behaviors in an FAS. Besides, the modular design and object-oriented approach make the system modeling much more easy and clear. Although the probabilistic issue in the system modeling can be dealt with stochastic Petri net (SPN), Molloy [12] addressed that the moderate number of places in SPN can easily be handled is approximate 50. The number of places in the proposed system is more than 50. Therefore, the SPN is not used in this study. Rather, some probabilistic modules are employed and interacted with the CTPN components. For example, the initial inspection result in the paint shop is generated in terms of the first run rate (the ratio of acceptant quality in the first time inspection in one shift) and a sequence of random numbers.

In this approach, the activities in the FAS are imbedded in multi-level structures [4, 11]. More specially, they include machining, sub-assembly, main assembly, plant process and production levels. Macro transitions and communication places are used to construct a hierarchical CTPN model. The macro describes a series of process actions, such as operations in a main assembly line or in a plant process. Independent modules are constructed first, then are grouped into a CTPN module library. The communication signal integrates vertically from the machine level to the production level and horizontally among macro transitions on the same (machine, main assembly, sub-assembly line or plant) level. Using this approach, a graphical

simulator [5] has been developed to simulate the FAS. This approach makes the modeling of the FAS much more easily and clearly. Furthermore, it makes the model extension, maintenance and debugging more convenient.

A high performance FAS depends very much on efficient job dispatching rules in reducing inventory and rendering a just-in-time (JIT) production. In this paper, several dispatching rules are developed by taking into account the balanced budget work standard. Finally, the proposed CTPN model can be simulated along the G2 expert system [3].

A Highly Model-Mixed Assembly Line

In this study, an FAS typically consists of a number of sub-assembly systems such as process actions, material handling devices, material storage and control units, inspection and gauging stations. These machines, equipments and assembly lines are often connected through an automated material delivery system. Production control messages including product information, assembly and control commands are routed via a communication system consisting of a computer, a control unit and a local area network (LAN).

A model-mixed assembly line [8] can assemble multiple types of product on the same assembly line to accommodate rapid product and demand changes. Key elements of the model-mixed assembly line are flexible fixtures, flexible tooling and skilled human operators. In addition, the timely delivery of material is crucial in keeping the line running at the desired speed.

A modular FAS has the following properties [11]:

1. Flexibility.
2. Production efficiency with low cost.
3. Meeting customers' need and delivering the products in a timely fashion.
4. Ability to quickly adapt to production condition changes.
5. Modular system design is easy to extend, correct and maintain.
6. Low work in process (WIP) and low inventory.
7. Ability to cope with rapid market and demand changes.
8. Low floor space requirement.

The following factors should also be considered:

1. Training.

Contributed by the Manufacturing Engineering Division for publication in the JOURNAL OF MANUFACTURING SCIENCE AND ENGINEERING. Manuscript received April 1996; revised April 1998. Associate Technical Editor: T. C. Woo.

Fig. 1 Hierarchical structure of the FAS

2. A real-time dispatcher.
3. Highly stable and reliable equipments.

To examine its structure and functionality, the FAS hierarchy can be decomposed into five levels, as shown in Fig. 1.

1. Production level: This is the highest level in the proposed FAS. It handles the system operations including master production schedule (MPS), system resource management, database, maintenance schedule and system performance measurement. The production level controller can coordinate the activities in the plants of this system. One example system of this level is the vehicle information system (VIS) in this case.

2. Plant process level: There are many main assembly lines in each plant. For example, engine assembly line (EAL) and engine dressing up (EDU) line are two main lines in the engine plant (EP). Several sub-assembly lines provide the material fitting to a specific main assembly line. For example, the header assembly line provides the assembled header to the EAL, and is not a main line in the EP. Therefore, the header assembly line is directly controlled by the EAL rather than the EP. Plant process level controller is the control center of a plant, and it coordinates and controls all main assembly lines in the plant.

3. Main assembly line level: It is the level of the main assembly lines below the plant process level. Main assembly line controller coordinates and controls all sub-assembly lines that provide the materials to this main assembly line.

4. Sub-assembly line level: It is the level of the sub-assembly lines under the main assembly line level. Since there are a lot of machines and equipments in a sub-assembly line, these machines and equipments can be grouped into several modules.

5. Machine level: It consists of machining, inspection and material handling equipments.

In this study, we focus on a particular automobile FAS. The material flow of this system is shown in Fig. 2. The system is composed of four plants, and they are body plant, paint shop, passenger plant and engine plant. The body plant, paint shop and engine plant handle the vehicle bodies, and the engine plant assembles the engines. The EDU line and PBS denote the engine dress-up line and the painted body storage area, respectively. The engine assembly line (EAL) starts from the Block-In station and ends at Engine Buy-Off station right before the EDU line. When the PBS station releases a vehicle body, it also

broadcasts the message to the engine plant. The vehicle body and the engine deck together in the decking station of the passenger plant. Finally, the assembled vehicles are inspected in the car assembly inspection (CAI) station.

Colored Timed Petri Net (CTPN)

The CTPN extends the framework of the original PN by adding color and time attributes to the net. The color attribute [4, 5, 6, 7] is developed to tackle large systems that have many similar or redundant logical structures. The time attribute [4, 5, 7] allows various time-based performance measures to be conducted in the system model. A time-delay can be assigned to either places or transitions to model the time elements in a system.

In a PN, when a token in an immediate place is enabled, the condition of the output transition determines the fate of the token. If the output transition is an immediate transition, then the enabled token fires and escapes from the input place. In the meantime, the transition releases this token to the output place(s). On the other hand, if the output transition is a timed transition, then the token will not be fired until a prescribed time delay is reached. Tokens in a timed place are processed in a similar way.

Notice that the number of arcs between the input place(s) and the output transition(s) is also important to the token. Three types of arcs are used; namely, directed, inhibitor and interrupt arcs. The directed arc connects between a place and a transition, or vice versa. It has no preconditions associated with it. The inhibitor and interrupt arcs are defined as follows. The place connected with the inhibitor arc is called the inhibitor place. When the inhibitor place contains the same color token as the output transition, then the output transition is inhibited to fire. Similarly, the place connected with the interrupt arc is called the interrupt place. When the interrupt place contains the same color token as the output transition, then the firing in the output transition is interrupted and further inhibited. For the immediate transition case, the interrupt and the inhibitor arcs behave the same. For the timed transition case, if the timed transition is firing and the interrupt place has the same color, then it interrupts the firing of this transition and returns the color token to the input place(s) [5, 7, 16, 17].

Macro Transition and Communication Place

Modular design is essential for a compact FAS model. In addition to the five levels mentioned above, each level is broken down into several sub-systems in the similar fashion. For example, the plant level under study consists of a body shop (BS), a paint shop (PS), an engine plant (EP), a passenger vehicle

Fig. 2 Material flow of an automobile FAS

assembly plant (PAP) and a commercial vehicle assembly plant (CAP). To facilitate a modular design, individual FAS modules are developed first for each of the five levels. Then, the modules from different levels are integrated together to form the complete FAS model.

In our approach, macro transitions are often used to formalize a modular design [4, 5, 7, 15, 16, 17]. A macro transition is the combination of a series of transitions, places and arcs. A macro transition may consist of several sub-macro transitions (it is, in essence, a macro transition but in different levels). A system can be composed of many macro transitions. The interconnection between different levels is achieved by communication places. Based on macro transitions and communication places, a hierarchical and modular model of the FAS can be constructed.

The communication and interface between the macro transitions are through four types of communication places; i.e. pitch-up, pitch-down, catch-up and catch-down places. The pitch-down and catch-down places are applied to the higher level modules, and the pitch-up and catch-up places are for the lower level modules. The higher level modules use the pitch-down place to send (or pitch) tokens to the lower level modules, and use the catch-down place to receive (or catch) tokens from the lower level modules. Similarly, the lower level modules use the pitch-up place to send tokens to the higher level modules, and use the catch-up place to catch tokens from the higher level modules [5, 7, 16, 17].

In the framework of the hierarchical design, we can model an FAS modularly. The primary consideration here is to make the model easy to extend, maintain and debug in the future.

A hierarchical CTPN is an ten-tuple, $CTPN = (P_t, T_t, P_0, T_0, P_c, T_m, A, B, F, C)$, where P_t is a set of timed places; T_t is a set of timed transitions; P_0 is a set of immediate places; T_0 is a set of immediate transitions; P_c is a set of communication places; T_m is a set of macro transitions; A is a set of directed arcs; B is a set of inhibitor arcs; F is a set of interrupt arcs; C is the color set of transitions and places.

These elements are further elaborated as follows:

1. Place: $P = \{p_1, p_2, p_3, \dots, p_n\}$ is a finite set of places, $n \geq 1$, including immediate places, timed places and communication places. Notice that the immediate places can describe the conditions or properties (without time factor) of resources; the timed places can describe the time properties of resources; the communication places can describe the communication packages in the hierarchical CTPN.

2. Transition: $T = \{t_1, t_2, t_3, \dots, t_m\}$ is a finite set of transitions, $m \geq 1$, including immediate transitions, timed transitions and macro transitions. Notice that the immediate transitions can describe the behaviors or events (without time factor) of resources; the timed transitions can describe the time properties of resources.

3. Color: $C(p)$ and $C(t)$ represent the color sets of places in P and transitions in T ;

$$C(p_i) = \{a_{i1}, a_{i2}, \dots, a_{iui}\}, \quad u_i = |C(p_i)|; \quad i = 1, 2, \dots, n$$

$$C(t_j) = \{b_{j1}, b_{j2}, \dots, b_{jvj}\}, \quad v_j = |C(t_j)|; \quad j = 1, 2, \dots, m$$

where n and m are non-negative integer; a and b are colors of places and transitions, and $|\cdot|$ denotes the cardinality.

4. Input, output, inhibitor and interrupt functions:

$I(p, t)(a, b) : C(p) \times C(t) \rightarrow N$, is an input function. It describes the mapping relation from the transition t with color b to the place p with color a , where N is a non-negative integer. Similarly,

$O(p, t)(a, b) : C(p) \times C(t) \rightarrow N$, is an output function.

$Inh(p, t)(a, b) : C(p) \times C(t) \rightarrow N$, is an inhibitor function.

$Int(p, t)(a, b) : C(p) \times C(t) \rightarrow N$, is an interrupt function.

5. Marking: $\mu(p) : C(p) \rightarrow N; \forall p \in P$, has n elements. $\mu(p_i)$ is an $(n \times 1)$ vector defined as:

$\mu(p_i) = \sum_{h=1}^{w_i} n_{ih} a_{ih}$, where n_{ih} is the number of colors a_{ih} at this instant; w_i is the total number of colors in place p_i . $\mu(p_i)(a_{ih})$ denotes the number of colors a_{ih} in place p_i . μ_0 is the initial marking.

6. Conditions of enabling and firing: The transition t_j is enabled with respect to the color b_{jk} if

$$\mu(p_i)(a_{ih}) \geq I(p_i, t_j)(a_{ih}, b_{jk}), \quad \forall p_i \in P, a_{ih} \in C(p_i)$$

$$\mu(p_k)(a_{kg}) = 0, \quad \forall Inh(p_k, t_j)(a_{kg}, b_{jk}), \quad \forall p_k \in P, a_{kg} \in C(p_k)$$

$$\mu(p_l)(a_{lf}) = 0, \quad \forall Int(p_l, t_j)(a_{lf}, b_{jk}), \quad \forall p_l \in P, a_{lf} \in C(p_l)$$

After the transition t_j is fired, two outcomes could appear. For an immediate transition the new marking μ' becomes:

$$\mu'(p_i)(a_{ih}) = \mu(p_i)(a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk}) - I(p_i, t_j)(a_{ih}, b_{jk})$$

For a timed transition, the temporary marking μ' is:

$$\mu'(p_i)(a_{ih}) = \mu(p_i)(a_{ih}) - I(p_i, t_j)(a_{ih}, b_{jk})$$

The transition waits for the firing time, then releases the token(s) to the corresponding place(s). Then the new marking μ'' becomes:

$$\mu''(p_i)(a_{ih}) = \mu'(p_i)(a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk})$$

7. Time function: It is simply the time attribute in the places and transitions. For the transition, $f(t(b_{jk})) : T \rightarrow N$, is the time required by the timed transition t associated with the color b_{jk} to complete the firing. For the place, $f(p(a_{ih})) : P \rightarrow N$, is the time delay required in the timed place p associated with the color a_{ih} to release.

8. Reachability set: It consists of all reachable markings from the initial marking.

9. Liveness, deadlock: For any reachable marking, if there exists a transition in this CTPN which is enabled, then the CTPN is live; otherwise, this CTPN is in a deadlock.

The icon definition of CTPN is shown in Fig. 3.

Object-oriented approach can describe the system as a collection of objects, and these objects can interact with one another by using both data structure and behavior. In the object-oriented CTPN model, the operations among all CTPN objects can be easily created, such as message passing among the CTPN objects, the method of each CTPN object, the data abstraction of each CTPN object, and so on. The inheritance of the object-oriented description eases the construction of the CTPN classes. In addition, the efforts of application development will be enormously reduced by using the developed CTPN classes libraries. In the proposed object-oriented model, the components of CTPN are designed as objects. Different classes, such as places, transitions and arcs, can be created as objects with attributes.

Fig. 3 Icon definition of CTPN graph

Fig. 4 The CTPN object hierarchy

For example, a timed place may have the following attributes: a boundedness (the maximum tokens in this place) attribute, a time attribute, a job attribute, a batch attribute, a buffer attribute and a machine attribute. In constructing a CTPN, we can use these classes and define the input and output functions with respect to these classes. This will create a design that is easy to extend, maintain and debug. This approach is implemented on an object-oriented real-time expert system, G2 [3]. The G2 environment provides an inference engine to reason about the objects in the knowledge base. The object hierarchy of this CTPN is shown in Fig. 4.

Modeling an Automobile FAS by Using CTPN

In this study, CTPN is used to model the FAS. The relationships of the physical assembly system and CTPN elements must be linked in terms of their properties. The events of the system are often modeled in terms of immediate transition, such as the event of decking competence of the vehicle body and engine. The process actions with time elapse are often modeled by the timed transitions, such as the hot-testing of an assembled engine. The resources are often modeled by the places, such as the engine buffers in the EP. The hierarchical modules of the FAS can all be modeled by the macro transitions.

The following notions are used in the modeling of an automobile FAS:

1. Immediate place describes the resources of the FAS, such as material, machine, message, control command and buffer.
2. Timed place is used to model a place that is associated with a time-delay, e.g. a conveyor belt on which components stay for a certain time.
3. Communication place describes a place which provides communication between macro transitions, such as that between a plant layer and a main assembly line layer.
4. Immediate transition can model the event occurred in the FAS, e.g. an operator completes one job. On the other hand, it can also model a decision making, such as selecting the next product type for an assembly line under a specific dispatching rule.
5. A timed transition can describe a process that needs time to complete.
6. Macro transition models the aggregation of many processes such as, the plants, main assembly lines or sub-assembly lines.

The following procedure is used in the construction of the model:

1. Analyze the FAS, use the hierarchical construction, and define the plants, main assembly lines, sub-assembly lines and machines in each level.
2. Define the system requirements, resources, decision making and events, then describe them using appropriate places and transitions.
3. Use macro transitions to model the plants, main assembly lines and sub-assembly lines. Modify them in different levels if necessary.
4. Define the processes of each level in terms of material flow or machines.
5. Draw flow charts of the processes in Step 4.
6. Define the interface and communication between different levels and macro transitions.
7. Use CTPN to model the flow charts in Step 5.
8. Use the net properties, such as boundedness, safeness, conservation, liveness, reachability, coverability, conflict, deadlock and firing sequences, to check the validity of the model.

Development of a Balanced Budget Work Standard for a Small Batch Production

The CTPN models in the proposed FAS can describe the activities and behaviors in the FAS. Being flexible and dexterous, human operators are key elements in a highly model-mixed assembly line. The performance of such a system ties closely with the operator's performance. Overloaded operators can have adverse effect on product quality. On the other hand, underloaded operators directly affect the throughput of the system. In a model-mixed assembly line, different product types often have different budget work standard (BWS, it is the standard time measurement for an operator to assemble one product completely). It is crucial to select a proper sequence of model mixes so that budget work standard can be "balanced" as close as possible. For example, the assembly time of a series of processes for an operator is 2.795, 1.487, 1.487 and 2.795 hrs. (hours), respectively. Then a job command sequence of 2.795, 1.487, 2.795 and 1.487 hrs. will be better, or more "balanced" than the command sequence of 1.487, 1.487, 2.795 and 2.795 hrs for the operator since the operator may not be very busy in some interval of time or not very relaxed in some interval of time, and the loading for the operator may be "balanced".

At present, job dispatching is performed manually by a human scheduler in order to balance the budget work standards. The process is subjective, unstable, and depends heavily on personal experience and attitude. Thus, there is a strong need to develop a systematic method and a computer program that can generate a balanced budget work standard for a small batch production. The algorithm of a balanced budget work standard serves to finely tune the dispatching results from the simulation. The fine adjustment can balance the loading of operators, and do no harm to the simulation results.

The following parameters are defined to facilitate the development of a balanced budget work standard for a small batch:

1. Weighed mean budget work standard:

$$\overline{BWS} = \frac{\sum_{i=1}^k W_i BWS_i}{n}$$

where n is the total number of products in the daily schedule; k is the total number of product types in the daily schedule; BWS_i is the budget work standard of product type i ; W_i is the number of products for product type i in the daily schedule.

2. Let M be the number of products in a small batch. The manager selects M out of S products from the sample space. The operator can reset the loading after assembling M products.

Namely, the operator can reset the overloading value to zero after assembling M products.

3. Let S be the number of products in the sample space $S \geq M$. The goal of dispatching is to select M out of S products to form a small batch and satisfy the balanced budget work standard.

A direct search algorithm for selecting M products to achieve a balanced budget work standard is developed as follows:

1. Select M products arbitrarily from S products. The number of possible selection combinations will be C_M^S .
2. Sum up the process time for each combination, i.e. $SUM(j)$, where j is from 1 to C_M^S .
3. Calculate the derivation of these process time from the mean budget work standard for each combination, $Error(j)$, defined as:

$$Error(j) = abs(SUM(j) - \overline{BWS} \times M)$$

where $abs(\cdot)$ denotes the absolute value.

4. If $Error(j)$ equals to zero, this combination, j , meets the balanced budget work standard.

5. If $Error(j)$ is not equal to zero for all combinations, choose the minimum one as $Error(j)$. This value will be recorded and added to the next batch computation to compensate the previous loading.

6. To improve the convergence and reduce the computation time, we can define an acceptable derivation range. If $Error(j)$ falls within this range, we consider this combination meets the balanced budget work standard requirement. If none of the combinations meets the specified range, we can relax this range until one of $Error(j)$ satisfies the range. Again, we may record this value and add it to the next batch computation to compensate the previous loading.

7. Once an acceptable combination is found, the same product type should not be used subsequently. In addition, the first product's type in this combination shall not agree with the last product's type in the previous combination.

8. The values of M , S , acceptable derivation range and the range relaxation are determined by taking into account both the product's assembly budget work standard and the desirable system performance.

Dispatching Rules for the Assembly of a New Engine

In the FAS under study, the vehicle body and engine are built in different plants. These two components are brought together to be assembled on the vehicle assembly line at a specified time. Sometimes, an engine assembly takes longer time than the vehicle trimming line takes. Consequently, the corresponding vehicle body arrives at the Body Decking station without an engine to be assembled with. Instead of stopping the line and waiting for the missing engine, the vehicle body and corresponding engine will be assembled at a later time off-line. Such off-line assembly is very costly. It is thus necessary to develop efficient dispatching rules to ensure that the components will meet at the right assembly station at the desired time and be assembled together on-line.

To develop these dispatching rules, we first need to measure the processing time for different assembly stations. Based on these results, we select several strategic points on the shop floor to send out control messages. Those points are shown in Fig. 2 (big bullet points). The messages to be sent out are stacked in a message queue. Based on proper dispatching rules and the balanced budget work standard, dispatching messages are sent to the engine assembly line to start the assembly of a new engine.

Four dispatching rules are developed as follows:

1. Dispatch of new engine blocks follows the PBS signal:

Fig. 5 Flow chart of the simulation program

The dispatching message queue is formed after receiving the PBS signal. This rule depends on whether the quality in the paint shop is stable and acceptable. If the quality is stable and acceptable in the paint shop, then engine will be assembled in time for the corresponding painted body.

2. Dispatch follows Surface Paint Information System (SIS):

This rule is independent of the original PBS signal. The dispatching messages are formed and sent out according to the surface paint inspection results. The surface paint quality is divided into five levels. Different levels will be handled by different processes. The message contains the following attributes: quality level, body number and body code. These attributes are used to sent dispatching priorities, and to stack them in the message queue.

3. Kanban Dispatch:

Kanban is a card used in the just-in-time (JIT) production system [1]. In this approach, the message queue is the engine types required by the EDU line. The required engines will be retrieved from the engine buffer at the end of engine assembly line. Each time an engine is retrieved by the EDU line, a kanban will be issued upstreamwise to trigger the dispatch of a new engine block.

4. A hybrid Approach—Switching between SIS and PBS signal:

In this approach, we count the number of painted bodied in the tunnel to be the index of switching. The index can switch the production mode between the signal from the PBS and the signal from the surface paint inspection station.

No matter which dispatching rule is used, the command for EAL to assemble a new engine will only be issued once for each engine to avoid duplicated orders.

Application Examples

The proposed CTPN modeling technique and dispatching rules are applied to an automobile factory. The flow chart of the simulation program is shown in Fig. 5. Each module of this simulation program is described as follows:

1. Man Machine Interface (MMI) module: This is a user interface to input the budget work standard, daily schedule, dispatching rule, first run rate, . . . , etc.

3. **Surface Paint Inspection module:** It simulates the surface paint inspection under a specified first run rate set at the MMI. The results are five levels of quality grades assigned to the body numbers.

4. Dispatching Rules module: Four dispatching rules to operate this FAS, and the dispatching command is issued by taking the balanced budget work standard into account.

5. **Paint Shop Assembly after Surface Paint Inspection module:** The portion of paint shop after inspecting the surface paint of the body is modeled as a macro transition. Notice that there are a series of places and transitions in this macro transition.

6. **PBS module:** This module simulates the automatic storage of painted vehicle bodies from the paint shop. It also simulates the retrieval/dispatch of vehicle bodies from the body buffer based on the balanced budget work standard. Notice that the body buffer is a temporary storage of the painted vehicle bodies in the paint shop.

7. Tunnel module: It simulates the activities of painted vehicle bodies inside the tunnel, and we can predict when a body will exit from the tunnel.

8. EAL Assembly module: The activity of the engine assembly line is modeled as a macro transition. There are a series of places and transitions in this macro transition.

9. **Engine Buffer Management module:** This module simulates the storage of engines, which have passed hot test, in the engine buffer. It also simulates the retrieval of engines from the engine buffer according to the request of EDU line.

10. **EDU Line Assembly module:** The activity of the EDU line is modeled as a macro transition. The command for EDU line to start dressing up an engine is from the Passenger Assembly Plant (PAP). Again, there are a series of places and transitions in this macro transition.

11. **Assembly Module before PAP Decking:** The activity of the passenger vehicle assembly line is modeled as a macro transition. This module also sends out the command to the EDU line describing the engine type which needs to be assembled (refer to the EDU Line Assembly module).

Fig. 7 CTPN for paint shop plant level

Fig. 8 CTPN module for the sub-assembly line level

Fig. 6 CTPN module for the production level

Fig. 9 The CTPN module for the PAP plant level

12. **Tow Motor Trolley module:** It simulates the shipping of engines between the PAP and engine plant by a trolley tow car.

13. **Decking Assembly module:** In this module, we will check whether the components are available to be assembled

Fig. 10 The CTPN module for the EDU line main-assembly level

on the vehicle assembly line. This part is modeled as part of PAP plant net.

The CTPN modules (or sub-nets) for different plants and assembly lines are shown in Fig. 6 to Fig. 10. Once the CTPN model is constructed, we can use these sub-nets to develop the software for simulation. In this study, we use a real-time object-oriented expert system G2 to simulate the FAS. The simulation flow chart is shown in Fig. 11. Results of the balanced budget work standard under the conditions of $S = 10$ and $N = 5$, are shown in Table 1. Note that 8Z, AZ, 5G, ..., etc. are the product's types, and "room" is the located position of the product in the automatic storage unit. Finally, we use Fig. 12 and Fig. 13 to compare the performance between the dispatching without and with balanced budget work standard. Clearly, the performance with balanced budget work standard is much better than that without balanced budget work standard.

Conclusions

An object-oriented modeling technique was used to develop a hierarchical and modular CTPN for a highly model-mixed automobile plant. Each object in the CTPN module has a num-

Fig. 11 Simulation flow chart for the firing of CTPN modules

Table 1 Results for the balanced budget work standard at the PBS station

sim-time	body-no	room	body-code
247	"R2500003G"	17	"5G"
251	"R2800002Z"	20	"8Z"
267	"R2500004G"	18	"5G"
271	"R2600003H"	19	"6H"
275	"R2A00002Z"	23	"AZ"
287	"R2700010U"	21	"7U"
291	"R2500005G"	25	"5G"
295	"R2300003Z"	26	"3Z"
299	"R2700011U"	22	"7U"
303	"R2500003Z"	28	"5Z"
307	"R2600004H"	29	"6H"
315	"R2500006G"	31	"5G"
319	"R2800003Z"	33	"8Z"
323	"R2600005H"	35	"6H"
327	"R2700012U"	24	"7U"

Fig. 12 The error of the budget work standard without using balanced BWS at the stand of PBS

Fig. 13 The error of the balanced budget work standard at the stand of PBS

ber of attributes. A complex system can be constructed in this manner using several levels and modules within each level. The material and information flows in the FAS were modeled by the communication places. Based on this model, the shop floor controller and real-time simulator were constructed. Furthermore, dispatching rules were developed to balance the budget work standard for a small batch production. These rules ensure

that the components of a vehicle will be available for assembly at different stations.

The CTPN models, especially the algorithm of the balanced BWS for a small batch production and dispatching rules, of the proposed system have been justified and validated by the Ford Lio-Ho motor company. Based on this research, the bottleneck occurred in the engine plant can be eliminated, the loading of operators can be balanced, and hence the quality of products can be stabilized. Besides, this CTPN models and simulation environment can be used to validate the new developing methodologies before they are actually implemented. The balanced BWS for a small batch production algorithm and the dispatching rules are integrated into the PC-based engine dispatching system (EDS). This system is installed in the engine plant of Ford Lio-Ho, and it carries out the engine dispatching order to the EAL. Currently, the EDS works with the original PBS dispatching system in parallel to operate the EAL smoothly. The dispatching results can satisfy the requirements of the engine plant.

In the future, the material management, shop floor monitoring and control, and virtual manufacture can all be developed in terms of the CTPN models and the EDS.

Acknowledgment

This work was partially supported by the National Science Council, Taiwan, R.O.C., under Grant NSC85-2212-E-002-073, the University Research Program from Ford Motor Company, and Ford Lio-Ho Motor Co. (FLH).

References

- 1 Bauer, A., Bowden, R., Browse, J., Duggan, J., and Lyons, G., *Shop Floor Control System—from Design to Implementation*, London, Chapman & Hall, 1991.
- 2 Desrochers, A.A., and Al-Jaar, R.Y., *Applications of Petri Nets in Manufacturing Systems—Modeling, Control and Performance Analysis*, New York, IEEE Press Marketing, 1995.
- 3 *G2 Reference Manual Version 4.0*, Gensym, 1995.
- 4 Huang, H.P., and Chang, P.C., "Specification, Modeling and Control of a Flexible Manufacturing Cell," *Int. J. Production Research*, Vol. 30, No. 11, pp. 2515–2543, 1992.
- 5 Huang, H.P., and Tseng, Y.H., "Modeling and Graphic Simulator for Integrated Manufacturing Systems," *Intelligent Automation and Soft Computing*, Vol. 1, pp. 183–186, 1994.
- 6 Huber, P., Jensen, K., and Shapiro, R.M., "Hierarchies in Coloured Petri Nets," *Advanced in Petri Nets 1990, Lecture Notes in Computer Science*, Springer-Verlag, pp. 313–341, 1990.
- 7 Kuo, C.H., "Application of Reliability and Equipment Maintenance to Flexible Manufacturing Systems," Master Thesis, Department of Mechanical Engineering, National Taiwan University, 1995.
- 8 Kuo, C.H., Huang, H.P., Wei, K.C., and Tang, S.S.H., "Dispatching and Simulation for Highly Model-Mixed Automotive Plants," *Proc. Int. Conference on Automation Technology*, Taiwan, Vol. 1, pp. 423–430, 1996.
- 9 Kuo, C.H., and Huang, H.P., "Colored Timed Petri Net Based Statistical Process Control and Fault Diagnosis to Flexible Manufacturing Systems," *IEEE Int. Conf. on Robotics and Automation*, New Mexico, Vol. 4, pp. 2741–2746, April, 1997.
- 10 Kuo, C.H., Huang, H.P., Wei, K.C., and Tang, S.S.H., "The Development of PC-Based Highly Model-Mixed Engine Dispatching Systems," *IEEE 6th Int. Conference on Emerging Technology and Factory Automation Proceedings*, Los Angeles, California, pp. 49–54, 1997.
- 11 Lu, S.S., and Huang, H.P., "Modularization and Properties of Flexible Manufacturing Systems," *Advances in Factories of the Future, CIM and Robotics*, M. Cotsaftis and F. Vernadat, eds., Amsterdam, Elsevier, pp. 289–298, 1993.
- 12 Molloy, M.K., "Performance Analysis Using Stochastic Petri Nets," *IEEE Trans. On Computers*, Vol. C-31, pp. 913–917, 1982.
- 13 Peterson, J.L., *Petri Net Theory and The Modeling of Systems*, Englewood Cliffs, N.J., Prentice-Hall Inc., 1981.
- 14 Tang, H.P., Ho, J.K.L., and Tam, H.Y., "Reconfigurable Data-Modules for the Reconfiguration of a Flexible Assembly System," *Intelligent Automation and Soft Computing*, Vol. 1, pp. 231–236, 1994.
- 15 Tsai, C.J., and Fu, L.C., "Modular Approach for Petri-Nets Modeling of Flexible Manufacturing Systems Adaptable to Various Task-Flow Requirement," *Proc. IEEE Int. Conf. on Robotics and Automation*, France, pp. 1043–1048, 1992.
- 16 Tseng, Y.H., "Modularized Modeling and Simulation of a Flexible Manufacturing System," Master Thesis, Department of Mechanical Engineering, National Taiwan University, 1993.
- 17 Yeh, M.C., "Development of Shop Floor Control System for flexible Manufacturing Systems," Master Thesis, Department of Mechanical Engineering, National Taiwan University, 1994.