International Journal of Industrial Engineering, 13(1), 99-107, 2006.

REPRESENTATION AND PERFORMANCE ANALYSIS OF MANUFACTURING CELL BASED ON GENERALIZED STOCHASTIC PETRI NET

YoonHo Seo[1], Taioun Kim[2*], ByungNam Kim[3], and DongMok Sheen[4]

[1]Korea University, Industrial & Systems Information Engineering, Seoul, Korea
[2*]Kyungsung University, Industrial Engineering, Busan, Korea
[3]Pukyong National University, Systems Management and Engineering, Busan, Korea
[4]Ulsan University, Transportation Systems Engineering, Ulsan, Korea
Email: yoonhoseo@korea.ac.kr; twkim@ks.ac.kr; kimbn@pknu.ac.kr; dmsheen@mail.ulsan.ac.kr
* Corresponding author

To achieve high flexibility and agility for rapidly changing customer's demand, Petri net has been adopted as a modeling and performance analysis tool. The Purpose of this paper is to propose modeling and performance analysis schemes of flexible manufacturing line using a generalized stochastic Petri net. The manufacturing line can be represented using workflows which are composed of bill of material and processes. Bill of process shows the precedence of processes and the relationship among the manufacturing and assembly operations. An algorithm generating a Petri net from bill of material database is proposed. The scheme of generalized stochastic Petri net utilizing both immediate and exponential distributed transitions are adopted to model a manufacturing cell with flexible machines, assembly lines and buffers. Performance analyses are performed based on qualitative and quantitative properties. For the qualitative analysis, behavioral and structural analyses are applied. Quantitative analyses are conducted using a Petri net simulator.

Significance: This paper demonstrates an approach to represent the process of manufacturing line using BOP and Petri net. The Petri net model can be generated automatically from BOM data. Performance analysis schemes are proposed based behavioral, structural and quantitative properties. Through the interactive modeling and analysis process, this technique enables a quick response to customer needs and will enhance the feasibility and productivity of manufacturing system design.

Keywords: Petri net, bill of process, bill of material, flexible manufacturing line, performance analysis

(Received 20 December 2003; Accepted in revised form 27 March 2005)

1. INTRODUCTION

As manufacturing paradigm has shifted from mass production to mass customization, variety and customer satisfaction based on flexibility and quick response are highly required. The modeling of business process and manufacturing cell plays an important role in engineering and manufacturing control system. Process models support the representation of manufacturing system and their decomposition into manageable pieces. Complex manufacturing line can be represented by bill of material (BOM) and their processes which correspond to bill of process (BOP). BOP shows the information of BOM and manufacturing process. It provides the relationship of assembly and disassembly and the precedence of processes. Different approaches have been tried for modeling and analysis of manufacturing system. As many companies have equipped with ERP system, manufacturing systems are required to be connected with the ERP system. Thus, manufacturing execution system (MES) was proposed for FMS control under ERP system. An IDEF0 model of an order handling shop floor having an FMS line is developed to identify functional requirements of MES, and then a two tier MES architecture satisfying the functional requirement is proposed (Choi and Kim, 2002). Structured analysis and design technique (SADT) is a standard tool used in design of flexible manufacturing system. The representation using SADT is mapped into Petri net model by Santarek and Buseif (1998). SADT/IDEF0 represents activity oriented modeling approach which are composed of input, output, condition and control. The inputs are those items which are transformed into outputs by the activity. Conditions and rules describe how the activity is performed. The mechanism represents resources for the activity. SADT/IDEF0 models ensure consistency at each level of decomposition, but they are static and do not show directly logical and time dependencies between them.

Workflow has been used for representing and implementing business processes. Aalst (1999) has proposed the architecture for role and functionality of workflow processes. In Europe, CIM open system architecture (CIMOSA) and workflow architecture were adopted for an enterprise engineering application in a low volume microsystems production (Dickerhof *et al.* 1999; Ortiz *et al.* 1999).

ISSN 1072-4761

To achieve high flexibility and agility for rapidly changing customer's demand, Petri net has been adopted as a modeling and performance analysis tool. Intensive research results of Petri net are introduced in the areas of FMS control, scheduling, planning areas (Zhou, 1995; Desochers and Al-Jaar, 1995). Petri-net-based tools and methods with a variety of problems associated with the design and implementation of FMS are introduced in Zhou and Venkatesh, (1999). It contains FMS modeling, simulation, performance evaluation of push and pull paradigms in flexible automation, and real-time Petri nets for discrete event control. Timed Petri net is used for modeling behavioral analysis and performance evaluation on an automotive assembly plant (Bayhan and Tuncel, 2002). A generalized stochastic Petri net (GSPN) has been used as a performance analysis tool for flexible manufacturing line (Kim *et al.* 2003). Chiola *et al.* (1993) has proposed the class of Petri net obtained by eliminating timing from GSPN models while preserving the qualitative behavior.

Most of the previous works concentrate on either workflow process representation, or Petri net-based modeling, or performance analysis. They do not deal with the whole process starting from BOM to the performance evaluation of flexible manufacturing cell.

The Purpose of this paper is to propose modeling and performance analysis schemes of manufacturing line using a generalized stochastic Petri net. Section 2 introduces BOM, BOP, Petri net modeling primitives and an automated Petri net generation algorithm from BOM database. Section 3 proposes Petri net-based modeling for flexible manufacturing line. Section 4 explores performance analysis based on qualitative and quantitative properties. Finally, section 5 concludes.

2. BOM, BOP AND GENERATION OF PETRI NET MODEL

Petri net provide a uniform environment for modeling, design and formal analysis of flexible and agile manufacturing system. In the manufacturing organization, basic information about products and production process are given in the BOM. BOM describes the component structure of a product in a tabular format within a relational database system. Starting from BOM data, BOP and Petri net are generated, and then performance analysis can be performed for the prior evaluation of manufacturing line. This process is conceptualized in Figure 1.

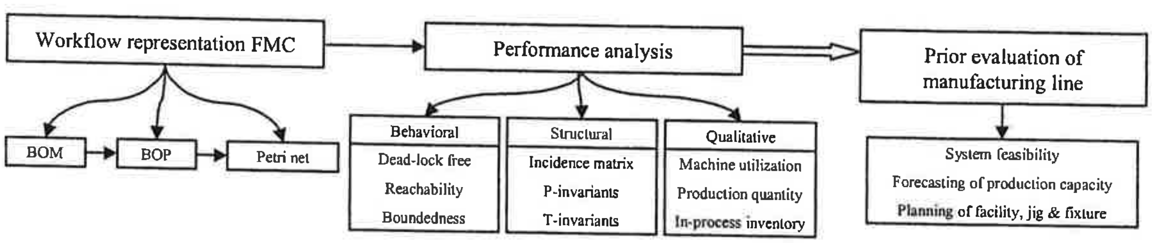


Figure 1. System architecture starting from workflow representation to performance analysis

Designed manufacturing line can be represented by BOM and their processes which correspond to BOP. A process is a collection of tasks, conditions, subprocesses, and their relationships with one another. BOP shows the information of BOM and manufacturing process. It provides the relationship of assembly and disassembly and the precedence of processes. Figure 2 shows the mapping between BOM and BOP.

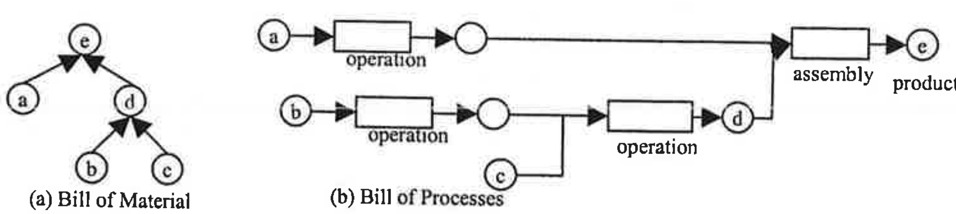


Figure 2. Relationship between BOM and BOP

The BOP model can be transformed into Petri net model because their structures are very similar in nature. Table 1 compares the relationship between BOP and Petri net model. Conditions and causal dependencies in the BOP are depicted using places in the Petri net model. Event and task in the BOP are transformed into transition in the Petri net. Also, case/state and sequential graph can be represented using marking and marking graph respectively.

Table 1. Relationship between BOP and Petri net model

BOP model	Petri net model
Condition/causal dependency (OR-split, OR-join)	Place
Event/task (AND-split, AND-join)	Transition
Case/state	Marking
Sequential case graph	Marking graph

Manufacturing line or business process takes several routings such as sequential, parallel, selective and iterative. They can be represented using Petri net primitives shown in Figure 3. From the figure, (a) and (b) show basic and sequential operations. The second task is performed as the result of the first. In a Petri net, this form of routing is modeled by linking the two tasks using a place. Cases (c) and (d) show parallel routing where two places link to a common transition. Parallel routing happens when more than one task is carried out at the same time or in any order. Parallel operation starts with AND-split in which two tasks happen following a transition. In the AND-join, a transition happens only when both conditions are fulfilled. Case (e) represents join/merge where several information/control sources lead to common condition or status represented by a place. An example in the manufacturing process corresponds to producing identical parts and sending them to a common station for further processing. Case (f) shows the choice operation in the dispatching problem where two or more machines are ready for a coming raw material. The decision is made based on priority rule, probability or frequency. Case (g) shows cyclic operation. A typical example for this case is robot's repetitive task of loading and unloading of a specific part. Also this cycle operation can describe an idle operation in a machine shop where a machine is waiting for work for the incoming task. Case (h) represents mutual exclusive situation in which a common resource is shared by two tasks.

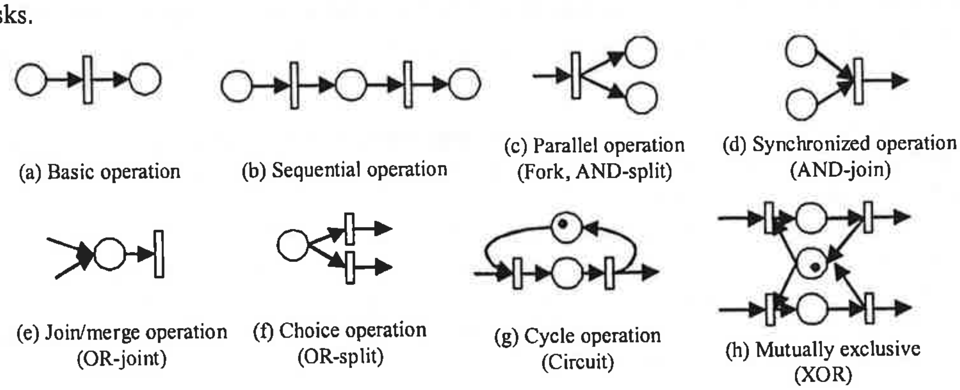


Figure 3. Modeling primitives used for Petri net design

In order to introduce a Petri net generation algorithm, formal definitions for BOM and Petri net are given.

Definition 1. BOM is defined using four tuples.

BOM = (C, A, D, R), where

- C is a finite set of components.
- $A \in C$, a set of parent components of the present node. When $A = \varnothing$, it corresponds to a root node.
- $D \in C$, a set of child components of the present node. When $D = \varnothing$, it corresponds to a leaf node.
- R represents the relationship between the present node and its child nodes, {AND, OR} $\in R$, where relationship between parts are AND/OR in the BOM. All child nodes are assumed to be AND relationship if OR relationship is not specified explicitly.

Definition 2. Formally, a stochastic Petri net (PN) is a bipartite directed graph with six tuples defined as:

PN = (P, T, I, O, F, m), where

- $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of places.
- $T = \{t_1, t_2, \ldots, t_m\}$ is a finite set of transitions.
- $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.
- $I \subseteq (PxT)$: a set of arcs, where I is an input mapping $P \times T \rightarrow \{0, 1\}$ corresponding to the set of directed arcs from P to T.
- $O \subseteq (TxP)$: a set of arcs where O is an output mapping $T \times P \rightarrow \{0, 1\}$ corresponding to the set of directed arcs from T to P.
- F: distribution function to assign a firing time to transition T.

m: $P \rightarrow \{0,1,2,3,\ldots\}$ is the marking of place, where m_0 specifies the initial marking.

GSPN contains immediate transition and transitions with exponentially distributed firing times. Markings in which at least one immediate transition is enabled are called vanishing markings since immediate transitions fire in zero time. Markings in which only exponential transitions are enabled are called tangible markings. A Petri net that contains tangible and vanishing markings is equivalent to an embedded Markov chain. If we remove the vanishing markings, reduced embedded Markov chain is generated.

Firing the transitions in a GSPN depends on whether we are examining tangible or vanishing markings. In the case of tangible markings, any enabled transition can fire next. The actual transition that fires depends on the firing rates on the enabled exponential transitions. For a vanishing marking, only the enabled immediate transitions are allowed to fire. Also, enabled and concurrent transitions fire immediately. In case of conflicting enabled immediate transitions, only one is allowed to fire at a time according to a predefined probability distribution.

According to the above definition about BOM and Petri net, an algorithm generating Petri net automatically is proposed as the following.

Based on the part 'a' in the BOM, in(a) and out(a) is defined as the following.

in(a) = $\{x \in C | (x, a) \in R\}$,

out(a) = $\{x \in C | (a, x) \in R\}$.

Step 1. (Initiation stage) Find a component without a child node (leaf node). If more than one component are found, choose the one which is found for the first time in the database. Suppose the selected component be 'a'. Create an initial PN = $\{P, T, I, O, F, m\}$ where P = {in(a), out(a)}, T = {process(a)}, I = {PxT: (in(a) → process(a)), O = TxP: process(a) → out(a)}, F is the firing distribution function of transition T, and m is the marking at in(a). Go to Step 2.

Step 2. If $C \cap D \neq \varnothing$ (When there are another leaf nodes), create PN = $\{P, T, I, O, F, m\}$ for all the remaining leaf nodes. Go to Step 3.

Step 3. If $C \cap P = \varnothing$ (When there are not any other parent component for the present one, it is a root component.), go to Step 5. Otherwise choose a component 'b' $\in C$ whose child node exists among the already processed set, ($C \cap D \in a_i$, where a_i is a set of already processed component).

Step 4. If the selected component $b \in C$ is an intermediate node, create revised PN = $\{P', T', I', O', D, m\}$, where P' = {out(a_i), out(b)}, T' = {process(b)}, I' = {(out(a_i) → x)} O' = {x → out(b)}. In this Step, decide whether the relationship between the lower components 'a_i' and present component 'b' is {AND, OR}. Go to Step 3.

Step 5. The generation of PN is finished. If we apply the algorithm for the BOM data shown in Figure 2-(a), the following Petri net is generated automatically as shown in Figure 4.

3. PETRI NET MODELING OF FLEXIBLE MANUFACTURING LINE

Consider a flexible manufacturing line composed of two machines M_1 and M_2, two assembly operations and four buffers in the process. Three Part types, P_1, P_2 and P_3 are input to the system for operations and assembly. Part type P_1 requires machine M_1 in operation 1 and machine M_1 or M_2 in operation 2. Part type P_3 requires machine M_2 in operation 1 and machine M_1 or M_2 in operation 2. After Part type P_1 finishing the operation 1 in machine M_1, P_2 is assembled to P_1, then they are put in the buffer 1 if next available machine is not available. Using machine M_1 or M_2, the subassembly $\{P_1P_2\}$ is processed its second operation. In the meantime, part P_3 proceeds to the operation 1 using machine M_2, then enters buffer 3 if next machine is not available. After finishing its operation 2 in M_1 or M_2, P_3 is processed to be assembled with $\{P_1P_2\}$. As the final output is assembled to a single product $\{P_1P_2P_3\}$, raw materials for P_1, P_2 and P_3 are the same quantity. The process of operation for the flexible manufacturing line is shown in Figure 5.

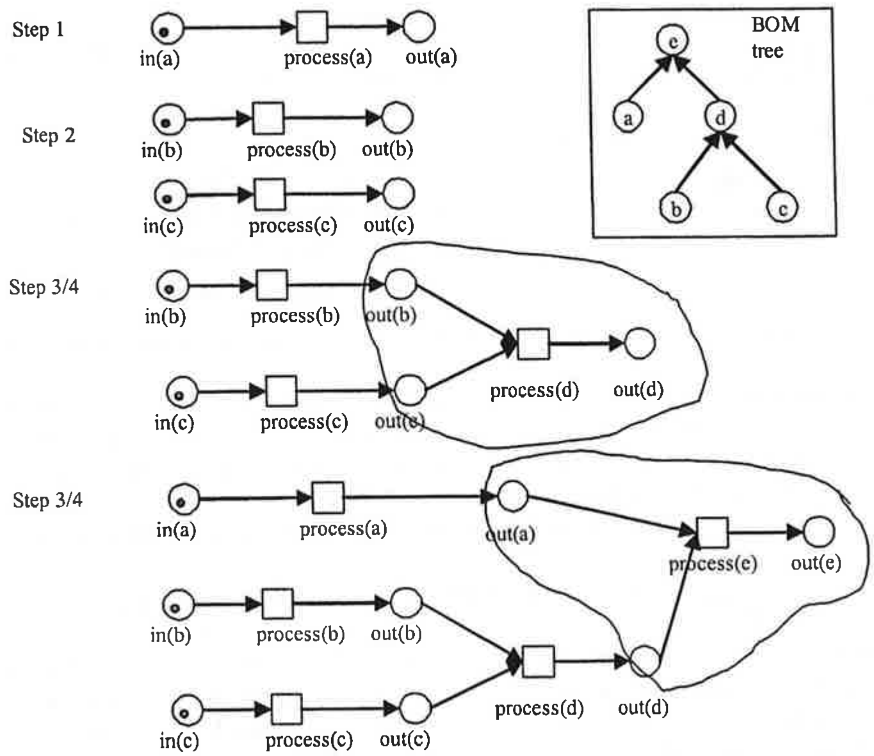


Figure 4. Petri net graph generated using the algorithm

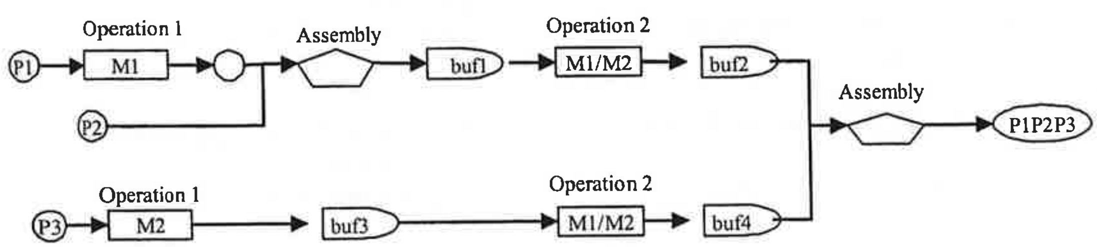
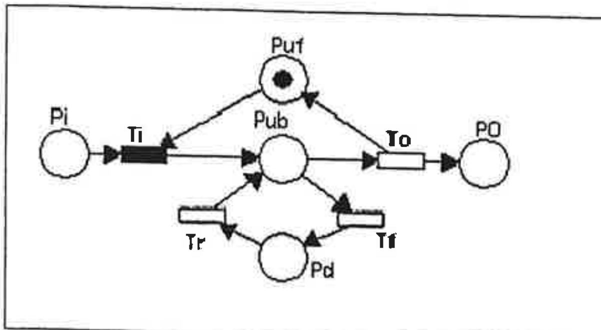


Figure 5. Operation process of flexible manufacturing cell

The given problem with two machines, two operations, two assembly processes and four buffers can be modeled and analyzed using Petri net system. As two machines M_1 and M_2 are assumed to be unreliable machine, the machine can be broken according to failure distribution.

A GSPN model which incorporates both stochastic timed transitions and immediate transitions is adopted in this research. The timed transitions have an exponential distribution with firing rate λ, which fire $1/\lambda$ time unit after they are enabled. The immediate transitions fire as soon as they are enabled. The machine operation, failure and repair rate are assumed to be λ_p, λ_f and λ_r respectively with exponential distribution. The subprocess with failure and repair process during machining operation is shown in Figure 6. In the sub process model, parts enter the machine through immediate transition T_i if place P_{uf} has a token. The status of P_{ub} with a token implies the start of machining operation. If the machine does not fail, the operation will take $1/\lambda_p$ time unit. When failure occurs in the machine, transition T_f fires at a rate $1/\lambda_f$ time unit. The repair occurs at a rate $1/\lambda_r$ time unit. The memoryless property of the exponential distribution implies that next transition can happen either in T_o or T_f.



P_i: place with input
P_{uf}: place with up and free with initial token
P_{ub}: place with up and busy
P_d: place with down (machine failure)
P_o: place with output
T_i: Transition ready for input
T_o: Transition output with processing rate λ_p
T_f: Transition with failure rate λ_f
T_r: Transition with repair rate λ_r

Figure 6. Subprocess with failure and repair process during machining operation

The places and transitions adopted in the model represent the following interpretation shown in Table 2. Only typical cases are explained, and the other remaining places and transitions have similar interpretations. The GSPN model for the given problem is shown in Figure 7. The operation in the unreliable machine follows the subprocess given in Figure 5. In order to be understood more easily, similar indexing schemes are adopted only by changing their subscripts. By putting the same quantity of raw material P_1, P_2 and P_3 as input to the system, a finished assembly from the three parts is produced in the final output.

Table 2. Interpretations of places and transitions

Places		Transitions	
P_{1i}	Part type P_1 is input for operation 1.	T_{1i}	P_1 is supplied to M_1 for the operation 1.
P_{1ub}	M_1 is up and busy for operation 1.	T_{1f}	Machine M_1 fails during operation 1.
P_{1d}	M_1 is down.	T_{1r}	Machine M_1 is repaired.
P_{m1}	Machine M_1 is available.	T_{1o}	Processing in operation 1 is finished.
P_{2ui}	Part P_1 is available for next process (up and idle).	T_{2i}	Part P_1 and P_2 are supplied for assembly operation.
P_{2a}	Part P_1 and P_2 are in assemble process.	T_{2o}	Assembly process is finished (out).
P_{buf1}	Buffer storage 1		
P_{m2}	Machine M_2 is available.		
P_{8a}	Assembly process of part P_1, P_2 and P_3.	T_{8i}	Part P_1, P_2 and P_3 are supplied to assembly process.
P_{8o}	Output of assembly process.	T_{8o}	Assembly process is finished (out).

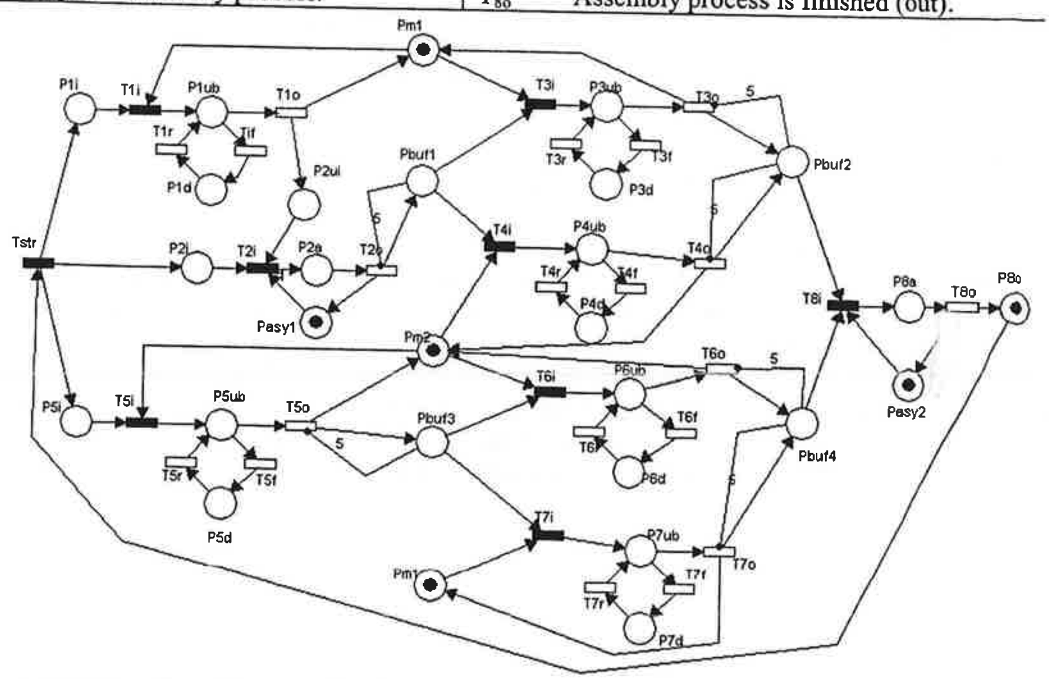


Figure 7. Generalized Stochastic Petri net model with two unreliable flexible machines and four buffers (Transition with black box implies immediate firing time and white box means timed transition.)

4. PERFORMANCE ANALYSIS OF PETRI NET MODEL

In order to analyze a Petri net model, qualitative and quantitative approaches are usually adopted. Behavioral and structural properties of Petri nets correspond to the qualitative properties. Behavioral properties imply that the properties of Petri net are dependent on the initial marking. Some examples of behavioral properties are work-in process, state of the resources, order released etc. Structural properties depend only on the structure of Petri net, and not on the initial marking and the firing policy. In manufacturing system, this is important because they depend only on the layout, not on the way the system is managed. Examples of structural properties are layouts, resources available, management system used etc. Productivity, resource utilization, inventory level and reliability correspond to quantitative properties. Quantitative properties are important for the efficient management purposes.

4.1 Performance analysis based on qualitative properties

The qualitative performance measures are reachability, boundedness, liveness, deadlock, reversibility and persistency. Suppose that m_0 be an initial marking representing the initial status of tokens in the places. Based on the above criteria, the model shown in Figure 6 satisfies all the characteristics. When the transition T_{str} is removed, it is not reversible because after firing all input parts with q_1, q_2 and q_3 tokens from places P_{1i}, P_{2i}, and P_{5i}, no more firing happens.

Structural properties do not depend on the initial marking, but on the topological structure of the net. They will use matrix-based approach. Suppose that an *incidence matrix* be $A = [a_{ij}]$ with an (n x m) of integers. A formula can be derived to test a final marking m_f is reachable from an initial marking m_0.

$$m_f = m_0 + Au \qquad (1)$$

where u is the summation of all u_i, $i = 0,1,\ldots,f\text{-}1$. This (m x 1) column vector, u_i, is called the firing count vector.

An *S*-invariant or *P*-invariant is an (nx1) nonnegative integer vector satisfying

$$x^TA = 0 \qquad (2)$$

From equation (1), mulitfying x^T in both sides,

$$x^Tm_f = x^Tm_0 + x^TAu \qquad (3)$$

Combining equation (2) and (3) yields

$$x^Tm = x^Tm_0 \qquad (4)$$

This implies that the total number of initial tokens in m_0, weighted by the *P*-invariant are constant.

A *T*-invariant is an (mx1) nonnegative integer vector satisfying

$$Ay = 0 \qquad (5)$$

From the equation (1), equation (6) is derived when $y = u$. This implies that when the firing count vector is identical with *T*-invariant, then the final marking is equal to the initial marking.

$$m_f = m_0 \qquad (6)$$

To illustrate the above properties, the GSPN model in Figure 7 is tested. An incidence matrix A from this model is derived. Martinez and Silva (1982) have proposed an efficient algorithm for computing the invariants of Petri net. The important issue is to find the minimal set of invariants which are linearly independent. When the above algorithm is applied to the above incidence matrix A, the minimal *P*-invariants are derived as the following. Suppose that the place vector x is assumed to be

$x = (P_{1i}\ P_{1ub}\ P_{1d}\ P_{m1}\ P_{2i}\ P_{2a}\ P_{bufl}\ P_{3ub}\ P_{3d}\ P_{4ub}\ P_{4d}\ P_{buf2}\ P_{5i}\ P_{5ub}\ P_{5d}\ P_{m2}\ P_{buf3},\ P_{6ub}\ P_{6d}\ P_{7ub}\ P_{7d}\ P_{buf4}\ P_{8a}\ P_{8o})$.

From equation (2), $x^TA = 0$, five *P*-invariants are acquired. Consider the initial marking $m_0 = (q_1\ 0\ 0\ 1\ 0\ q_2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ q_3\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$. Applying equation (4) for the *P*-invariants gives

$$m(p_{4ub}) + m(p_{4d}) + m(p_{5ub}) + m(p_{5d}) + m(p_{m2}) + m(p_{6ub}) + m(p_{6d}) = 1 \qquad (7)$$

$$m(p_{1ub}) + m(p_{1d}) + m(p_{m1}) + m(p_{3ub}) + m(p_{3d}) + m(p_{7ub}) + m(p_{7d}) = 1 \qquad (8)$$

$$m(p_{2i}) + m(p_{2a}) + m(p_{bufl}) + m(p_{3ub}) + m(p_{3d}) + m(p_{4ub}) + m(p_{4d}) + m(p_{buf2}) + m(p_{8a}) + m(p_{8o}) = q_2 \qquad (9)$$

$$m(p_{1i}) + m(p_{1ub}) + m(p_{1d}) + m(p_{2ui}) + m(p_{2a}) + m(p_{bufl}) + m(p_{3ub}) + m(p_{3d}) + m(p_{4ub}) + m(p_{4d}) + m(p_{buf2}) + m(p_{8a}) + m(p_{8o}) = q_1 \qquad (10)$$

$$m(p_{5i}) + m(p_{5ub}) + m(p_{5d}) + m(p_{buf3}) + m(p_{6ub}) + m(p_{6d}) + m(p_{7ub}) + m(p_{7d}) + m(p_{buf4}) + m(p_{8a}) + m(p_{8o}) = q_3 \qquad (11)$$

The *P*-invariants of equation (7) and (8) correspond to the activities of machine 2 and machine 1 respectively. The *P*-invariants of equation (9), (10), and (11) imply that the total number of parts of part type P_2, P_1 and P_3 are q_2, q_1 and q_3 respectively.

Similarly, solving equation (5) gives the minimal *T*-invariants as the following.

$$\{t_{1f}, t_{1r}\}\ \{t_{3f}, t_{3r}\},\ \{t_{4f}\ t_{4r}\},\ \{t_{5f}, t_{5r}\},\ \{t_{6f}, t_{6r}\},\ \{t_{7f}, t_{7r}\} \qquad (12)$$

$$\{t_{str}, t_{1i}, t_{1o}, t_{2i}, t_{2o}, t_{3i}, t_{3o}, t_{5i}\ t_{5o}, t_{7i}, t_{7o}, t_{8i}, t_{8o}\} \qquad (13)$$

{ t_{str}, t_{1i}, t_{1o}, t_{2i}, t_{2o}, t_{4i}, t_{4o}, t_{5i} t_{5o}, t_{6i}, t_{6o}, t_{8i}, t_{8o}} (14)

Equation (12) represents failure and repair operations, in which the system returns to the normal process before the failure occurred. Equation (13) and (14) represent production cycles utilizing machines M_1 and M_2 respectively, in which some subprocesses are appeared in both cases because the system is a flexible manufacturing cell sharing machines M_1 and M_2.

4.2 Performance analysis based on quantitative properties

Performance analysis based on quantitative measures is tried using Petri net simulator. For this purpose, the performance measure should be devised first. Average production rate, in-process inventory, and average utilization are adopted as performance measures. The average production rate in the flexible manufacturing line can be computed as

$$P = \text{rate}(T_{8o}) * \text{prob}\{m(P_{8a})=1\}.$$

The average utilization of machine 1 (UM1) and machine 2 (UM2) and two assembly operations are

$$UM1 = \text{prob}\{(m(P_{1ub})=1)+(m(P_{3ub})=1)+(m(P_{7ub})=1)\},$$
$$UM2 = \text{prob}\{(m(P_{4ub})=1)+(m(P_{5ub})=1)+(m(P_{6ub})=1)\},$$
$$\text{Assembly } 1 = \text{prob}\{(m(P_{2a})=1)\},$$
$$\text{Assembly } 2 = \text{prob}\{(m(P_{8a})=1)\}.$$

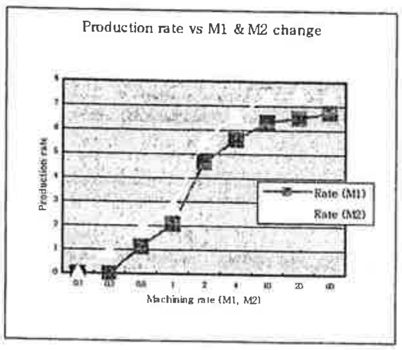
The average in-process inventory is

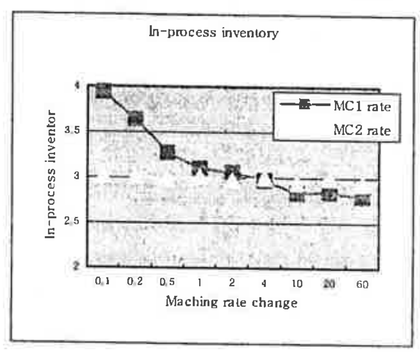
$$n = \{\text{the total number of tokens in the process} - E\,[m(P_{m1}) + m(P_{m2}) + m(P_{asy1}) + m(P_{asy2})\,]\}.$$

The basic parameters adopted for the simulation is summarized as the Table 3. Using these measures, the simulation has been performed in the Petri net simulator. The results are shown in Figure 8. The production rate increases as machining rate increases. The in-process inventory decreases when machining rate of M_1 increases. But, the change of machining rate of M_2 does not affect the in-process inventory because an additional assembly operation is included after the operation 1 of M_1. When the machining rate of M_1 (M_2) increases, the utilization of M_1 (M_2) decreases, while the utilization of M_2 (M_1), assembly 1 and assembly 2 increases.

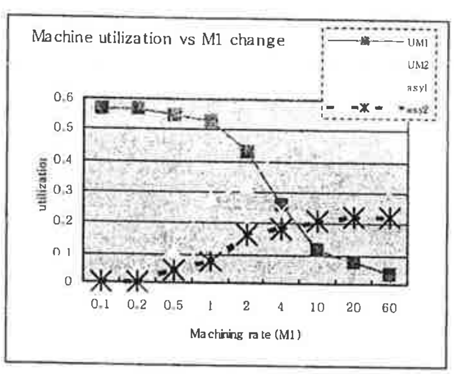
Table 3. Parameters for simulation

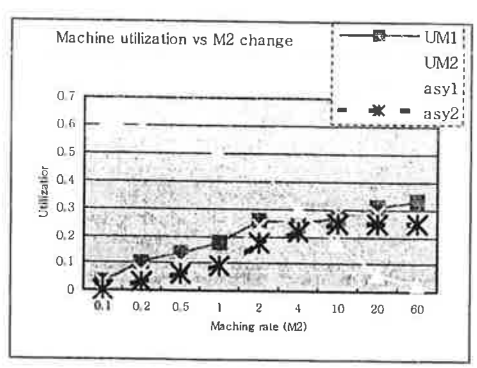
Machine 1			Machine 2			Assembly 1	Assembly 2	Buffer 1 ~ 4
λ_p	λ_f	λ_r	λ_p	λ_f	λ_r			
4	3	5	2	4	6	2	2	5





(a) Production rate versus machining rate change (b) In-process inventory versus machining rate change





(c) Machine utilization versus machining rate of M_1 (d) Machine utilization versus machining rate of M_2

Figure 8. Simulation results for the flexible cell according to the change of machining rate of M_1 and M_2

5. CONCLUSIONS

An algorithm of generating Petri net automatically from BOM data and performance analysis schemes based on qualitative and quantitative properties are proposed. The proposed method was implemented using an example problem from the flexible manufacturing cell. A complex manufacturing system can be decomposed into controllable smaller cells and the decomposed cells can be represented based on BOM and BOP. An algorithm was developed to generate Petri net graph automatically based on BOM. This algorithm can be applied to create Petri net from BOM data used in the legacy system. The proposed Petri net model was evaluated based on behavioral and structural properties. For behavioral properties, reachability, boundedness, liveness, deadlock, reversibility, and persistence were tested. For structural properties, *P*-invariants and *T*-invariants were used based on incidence matrix. Quantitative properties are tested based on production rate, in-process inventory and machine using Petri net simulator.

Two important aspects in manufacturing are regarded as a design and performance evaluation. Considering this, the suggested model can represent a flexible manufacturing cell very efficiently. Also, the model can be tested for system feasibility and performance analysis. Further research areas include developing more performance analysis measures and finding parameters for machining condition and process control. Also, developing statistical model to compare with Petri net will enhance the validity of Petri net modeling.

6. ACKNOWLEDGEMENTS

This work was supported by grant No. (R01-2002-000-00232-0) from the Basic Research Program of the Korea Science & Engineering Foundation.

7. REFERENCES

1 Aalst, W.M.P. van der (1999). On the automatic generation of workflow processes based on product structures. Computers in Industry, 39, 97-111.

2 Bayhan, G. M. and Tuncel, G. (2002). Modeling, behavioral analysis and performance evaluation of an automotive assembly plant using Petri nets. International Journal of Industrial Engineering, 9:3, 238-247.

3 Chiola, G., Marsan, M., Balbo, G. and Conte, G. (1993). Generalized stochastic Petri Nets: A definition at the net level and its implementation. IEEE Transaction on Software Engineering, 19:2, 89-107.

4 Choi, B.K. and Kim, B.H. (2002). MES architecture for FMS compatible to ERP, International Journal of Computer Integrated Manufacturing, 15:3, 274-284.

5 Desrochers, A. and Al-Jaar, R. (1995). Applications of Petri Nets in Manufacturing Systems, IEEE Press.

6 Dickerhof, M., Didic, M.M. and Mampel, U. (1999). Workflow and CIMOSA-background and case study, Computers and Industry, 40, 197-205.

7 Kim, T., Seo, Y. and Sheen, D. (2003). Modeling and performance analysis of manufacturing line using Petri nets. In Proceedings of 8th Annual International Conference on Industrial Engineering, Las Vegas, Nevada, November 10-12, 1174-1179.

8 Martinez, J. and Silva, M. (1982). A simple and fast algorithm to obtain all invariants of a generalized Petri net. in Application and Theory of Petri nets (C. Girault and W. Reisig, eds.), New York: Spring-Verlag, 301-311.

9 Ortiz, A., Lario, F., Ros, L. and Hawa, M. (1999). Building a production planning process with an approach based on CIMOSA and workflow management systems. Computers in Industry, 40,207-219.

10 Santarek, K. and Buseif, I.M. (1998). Modeling and design of flexible manufacturing systems using SADT and Petri nets tools. Journal of Materials Processing Technology, 76, 212-218.

11 Zhou, M.C. (1995), Petri Nets in Flexible and Agile Automation, Boston, Kluwer.

12 Zhou, M.C. and Venkatesh, K. (1999), Modeling, Simulation, and Control of flexible Manufacturing Systems: A Petri Net Approach, River Edge, NJ, World Scientific.