Division of Research
Graduate School of Business Administration

September 1983

USING PETRI NETS TO REPRESENT PRODUCTION PROCESSES

Working Paper No. 339

Didier Dubois
D.E.R.A./C.E.R.T.
Toulouse

Kathryn E. Stecke
The University of Michigan

Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio, TX
December 14-16, 1983

USING PETRI NETS TO REPRESENT PRODUCTION PROCESSES

DIDIER DUBOIS

Département d'Études et de Recherches en Automatique
Centre d'Études et de Recherches de Toulouse
31055 Toulouse, Cedex FRANCE

KATHRYN E. STECKE

Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan USA

ABSTRACT

A preliminary investigation, and extension of the power, of timed Petri nets to describe, model, and analyze production processes is reported. In particular, insights into real-time control aspects as well as the performance of flexible manufacturing systems are sought. Comparisons with previous investigative models are made. New and general modeling conventions are provided, which extend the realm of Petri net modeling capabilities. Various realistic aspects of manufacturing processes are modeled. Efficient algebraic tools to analyze a certain class of Petri nets are described [1]. We show that, under some structural assumptions, timed Petri net models of manufacturing processes translate into linear equations in a (max,+)- based algebra. Efficient algorithms can solve these equations for the purposes of both performance evaluation and real-time control.

INTRODUCTION

Petri nets are useful to model systems whose behavior can be described as interferences between asynchronous and concurrent processes. To date, Petri nets have mainly been used to describe computer operating system behavior. Particular applications include the descriptions of computer hardware and software [9], communication protocols [3], and queueing networks [4], [6]. In addition, production systems, such as job shops and flexible manufacturing systems, exhibit such interference during the manufacture of part types with different processing sequences. However, to our knowledge, Petri nets have not been used to analyze control problems of production systems. Perhaps the main reason for this is that there is very little literature available on timed Petri nets.

The bulk of the literature focuses on the analysis of qualitative properties of complex, concurrent computer systems, such as the non-existence or detection of deadlocks, the safeness, boundedness, conservation, liveness, and, in particular, reachability of the states of a Petri net. These properties are described in §1.

Ramchandani [8] first introduced timed Petri nets and provided methods to calculate throughput for certain classes of Petri nets. Ramamoorthy and Ho [7] specify an enumeration procedure to find the maximum system performance. Sifakis [10] generalizes the results of [8].

Our motivation to investigate the usefulness of timed Petri nets is the study of planning and control problems of flexible manufacturing systems (FMSs). The main models used to date to analyze performance have been queueing networks [11], [13], perturbation analysis [5], and simulation [12]. Queueing networks provide information about the average system behavior observed over a long time period (steady state). They are useful for qualitative answers to design and planning problems of FMSs [13], but not very useful to study control issues. Perturbation analysis "views a queueing network as a stochastic dynamical system evolving in time and observes the sample realization of its trajectory,"[5] similar to simulation. However, by perturbing one event, and following through an analysis, many useful questions can be answered without repeating a simulation run. Perturbation analysis has been used to answer questions that product form queueing networks can't address. Most often, real-time issues are analyzed using simulation, which is flexible, but timeconsuming and expensive.

Our aim is to demonstrate the usefulness of Petri net models to efficiently analyze problems of real-time control as well as the performance evaluation of production systems. Our main motivation is to use such analyses to gain insight into the real-time behavior of deterministic processes, FMSs in particular. For example, given a finite number of various resources (machines, pallets, fixtures, carts, tools,...), after an initial transient period, we conjecture that a deterministic production system will eventually reach a periodical steady state. Petri net representation can help to capture such behavior, under certain modeling conditions which are provided subsequently. We have found new tools that are available to efficiently analyze a special class of timed Petri nets. These are shown to be equivalent to certain linear, dynamical, discrete-event systems.

In §1, the notation and definitions that are required to represent manufacturing processes by Petri nets are reviewed. New modeling conventions are suggested in §2, which extend the realm of Petri net modeling capabilities. §3 contains examples of Petri nets that model various realistic aspects and components of production processes, such as flowshops, blocked machines, assembly processes, and the FIFO control rule. We model finite buffer situations as well as carts, pallets, and different fixture types. Processing, transportation, set-up, and waiting times can be included. Our examples demonstrate the new modeling capabilities. The algebraic tools that are available to analyze certain types of Petri nets are described in §4. The advantages and limitations of Petri net representations are outlined in §5. A summary of existing results and future research is provided in §6.

1. NOTATION AND DEFINITIONS

A Petri net structure is a four-tuple, $C=(P,T,I,O)$, where $P=\{p_1,p_2,\ldots,p_n\}$ is a finite set of places and $T=\{t_1,\ldots,t_m\}$ is a finite set of transitions. $P\cap T=\emptyset$. $I:T\to 2^P$ is the input function, which defines for any transition, its set of input places. A Petri net graph is a representation of a Petri net structure as a bipartite directed multigraph. In the graph, a circle represents a place and a bar represents a transition.

A marking μ of a Petri net is a function from P to the nonnegative integers N; $\mu:P\to N$. $\mu(p_i)=\mu_i$= number of tokens in place p_i. Tokens reside in the places and

control the firing of transitions. A transition may fire when it is enabled. A transition is enabled when there is at least one token in each of its input places. A transition fires by removing one token from each of its input places and distributing these tokens among its output places. If $X_{I(t_j)}$ and $X_{O(t_j)}$ are characteristic functions of sets $I(t_j)$ and $O(t_j)$, respectively, then the firing of t_j changes the marking μ into μ' such that

$$\mu'(p_i) = \mu(p_i) - X_{I(t_j)}(p_i) + X_{O(t_j)}(p_i), \ \forall\, i. \qquad (1)$$

The state of a Petri net is defined by its marking. The next state function δ of a Petri net with marking μ, denoted (C,μ), is defined for any enabled transition t_j, by $\delta(\mu,t_j)=\mu'$, where μ' satisfies (1). $\delta(\mu,t_j)$ is not defined if t_j is not enabled. μ' is said to be immediately reachable from μ. The execution of a Petri net results in a sequence of markings $(\mu^0,\mu^1,\ldots,\mu^n,\ldots)$ and a sequence of transitions $(t_{j_0},t_{j_1},\ldots,t_{j_n},\ldots)$ such that $\forall\, k \geq 0$, $\delta(\mu^k,t_{j_k})=\mu^{k+1}$.

The following properties are those that are addressed in the Petri net literature. A marking μ' is reachable from μ if there are sequences of markings $(\mu^0,\ldots,\mu^n)$ and transitions $(t_{j_0},\ldots,t_{j_n})$ such that $\mu=\mu^0$, $\mu'=\mu^n$, and μ^{i+1} is immediately reachable from μ^i, $i=0,\ldots,n-1$. The reachability set of a Petri net (C,μ) is the set of all reachable markings from μ and denoted $R(C,\mu)$. It includes μ itself.

A place p_i is said to be safe if $\forall\, \mu' \epsilon R(C,\mu)$, $\mu'(p_i)<1$, i.e., p_i contains at most one token. A Petri net (C,μ) is safe when all of its places are safe. When the upper bound on the number of tokens which can be in a place simultaneously exists (but is not necessarily one), the place is said to be bounded. A bounded Petri net (C,μ) contains only bounded places.

A Petri net (C,μ) is strictly conservative if and only if $\forall\, \mu \epsilon R(C,\mu)$,

$$\sum_{p_i \epsilon P} \mu'(p_i) = \sum_{p_i \epsilon P} \mu(p_i), \qquad (2)$$

i.e., the number of tokens in the Petri net remains a constant. As a consequence, for any transition, the number of input places equals the number of output places. A transition t_j in a Petri net (C,μ) is potentially firable if there exists a $\mu' \epsilon R(C,\mu)$ such that t_j is enabled in μ'. t_j is live if it is potentially firable for any $\mu' \epsilon R(C,\mu)$. A Petri net having only live transitions is live. Petri nets which are not live have deadlocks.

A decision-free Petri net structure is characterized by the existence of a single input arc and a single output arc for each place, i.e.,

$$\forall p_i,\ \exists!(t_j,t_k) \ni\ p_i \epsilon O(t_j) \cap I(t_k). \qquad (3)$$

A directed circuit in a Petri net is a sequence of transitions and places: $t_{j_1},p_{i_1},\ldots,p_{i_{n-1}},t_{j_n}$, such that $t_{j_1}=t_{j_n}$ and $\forall k=1,\ldots,n-1$, $p_{i_k} \epsilon O(t_{j_k}) \cap I(t_{j_{k+1}})$. A Petri net is strongly connected if every pair of places is contained in a directed circuit.

For the subclass of strongly connected, decision-free Petri nets, the following results are known [6]:

1. The number of tokens on a circuit does not change after a transition fires;
2. Liveness is equivalent to the existence of at least one token on each circuit;
3. Safeness is equivalent to the situation where every place is in a circuit containing exactly one token;
4. A marking μ' is reachable from a live marking μ if and only if the number of tokens in each circuit is the same for μ and μ'.

A timed Petri net is a Petri net C with an associated mapping $\tau:T \to [0,+\infty)$. Each transition t_j has a firing time $T_j=\tau(t_j)$. Tokens are moved from the input to the output places of t_j only after time T_j has elapsed since firing began. In a timed Petri net, each firing sequence $(t_{j_1},\ldots,t_{j_n})$ has a minimal duration, which is $\sum_{k=1}^{n} T_{j_k}$. Also, availability times may be associated with tokens, as part of the initial conditions.

2. VARIOUS MODELING PRINCIPLES

New and more general uses of Petri nets to describe production processes are now provided in §2.1. Following these is a comparison with previous modeling conventions in §2.2. The advantages of our suggestions and some limitations of the conventional methods are shown through examples. The power of these modeling conventions will be further displayed in §3.

2.1 Suggested Modeling Principles

The following modeling conventions are suggested to allow both more general applications of Petri nets as well as efficient algorithms to analyze their performance for certain classes of Petri nets.

i) The set of activities (tasks, operations) to be performed in the production system is represented by the set of transitions. Each transition is assigned the corresponding activity duration.

ii) An activity is defined by the simultaneous use of one or more resources. In particular, the number of input places of a particular transition indicate the number of resources which the activity simultaneously requires. Output places specify the activities that are required next and the release of certain resources.

iii) Tokens of various types represent available resources which flow through activities according the system control rule.

2.2 Comparison with Conventional Modeling

Differences, advantages, and limitations of the previous modeling conventions and those suggested in §2.1 are now demonstrated through examples of PERT/CPM charts and queueing networks.

PERT/CPM Charts. In conventionally modeling a PERT chart with a Petri net, an activity is represented by a place, precedence constraints are represented by transitions, and time is not considered [6]. In our Petri net model, both activities and precedence are represented by transitions. Places are reserved to model multiple resources of limited amounts. In addition, our timed Petri nets are more powerful models than PERT/CPM charts since:

i) Circuits of activities are allowed. This generalizes PERT modeling by enabling the activities to be performed repeatedly, if necessary.

ii) Required resources for activities appear explicitly, as tokens, in the representation.

iii) Nondeterministic aspects can be dealt with. For example, the order in which a particular resource performs some tasks may not be totally specified. In this sense, Petri net models

subsume disjunctive graphs.

Queueing Networks. A manufacturing system can be modeled as a queueing network, where machines and transporters are the servers and the parts to be processed are the customers. Suppose that the part route is known and the ordering of customer service is fixed.

Then previous Petri net representations of queueing networks, both untimed [6] and timed [4], have the queues as places, servers as transitions, and customers as tokens. These specifications appear to be arbitrary: alternatively, one may view machines as flowing through parts. More seriously, limited amounts of multiple resources cannot be modeled. For example, a server may require tools to service a customer. Also, blocking effects due to limited queue capacity cannot be modeled.

An alternative Petri net model, following the suggestions provided in §2.1, would represent the service activities as transitions of some duration. Each transition has two input places, for both server and customer tokens. Note that if additional resources are required, such as a robot, buffer, transporter, pallet, fixture, and/or one or more cutting tools, these are modeled with additional places (or different token types). In addition, finite buffers, blocked machines, FIFO queue discipline, and assembly processes can be modeled. Examples of these are provided in §3.

3. MODELING SOME COMPONENTS OF MANUFACTURING PROCESSES

In order to assess the descriptive power of Petri net modeling as applied to production processes, examples of different features are now provided.

3.1 Fixed-Route Flexible Manufacturing System

Consider the FMS of Figure 1, consisting of three different machine tools that process three different part types according to a specific part mix and the specified part routes.

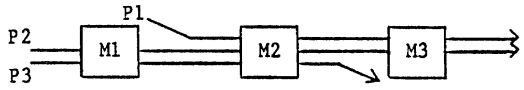

Figure 1. Three-Machine FMS.

Processing times of each operation of each part type on each of the machine tools are given in the following matrix:

	P1	P2	P3
M1		1	5
M2	3	2	3
M3	4	3	

The characteristics of the system that are modeled include the following:

1. The sequence of parts on the machine tools is given in Figure 1;
2. Set-up times on a machine tool for part changeover are negligible (the system is flexible, having automated tool interchange);
3. Transportation times are neglected;
4. For each part type, there is one pallet and one fixture;
5. The product mix is balanced, and can be obtained through a periodic input of parts. Here the sequence 1-2-3 is used.

The pallets ensure a proper positioning of parts on machines. When a pallet leaves the last machine on its processing sequence, it returns to the first machine to carry a new part. (The use of carts to transport parts is modeled in §3.3.)

The Petri net of Figure 2 is obtained. For

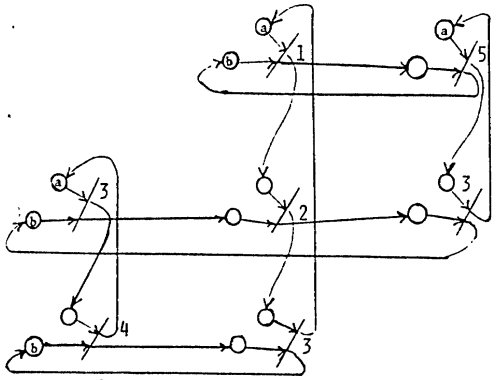

Figure 2. Petri Net of a Three-Machine FMS with Three Part Types.

clarity, the different token types in Figure 2 (machine tools and pallets) have been alphabetically named. Sometimes different token types are alternatively represented by colored tokens. However, they are unnecessary in this instance since the different token types appear in distinct places. Colored Petri nets are sometimes used to simplify large Petri net representations [3]. The Petri net of Figure 2 is decision-free, safe, and live for any feasible initial marking, for example, a marking consisting of three "machine" tokens and three "pallet" tokens. A similar Petri net can be constructed for a more general job shop type of FMS. In addition, adding another pallet for one of the part types can easily be represented. Two ways are suggested:

i) Expand the number of transitions;
ii) Add a token to the appropriate processing sequence circuit.

In the first case, the safeness of the Petri net is preserved, but not in the second case. Both proving the equivalence of these representations, as well as determining their pro's and con's, are matters for further research.

Both set-up and transportation times can easily be introduced as one-place-in, one-place-out transitions that separate processing transitions. These are demonstrated in §3.2. To keep the Petri net of Figure 2 simple, we would value the arcs linking processing transitions to their output places by the required set-up or transportation times. An efficient analysis technique is demonstrated using this example in §4.

3.2 Blocking Phenomena

In the example of §3.1, it is assumed that there is sufficient storage provided between machines so that no machine is blocked when the downstream machine is busy. We now model a three machine system similar to that of Figure 1, but with no buffer storage available. For simplicity of exposition, there are two part types, each having the same processing sequence: M1→M2→M3, but possibly different operation times. There is one pallet for each part type.

To capture the blocking effect, we distinguish between two types of transitions: one type represents the processing of a part on a machine for a fixed duration; the other represents the instantaneous departure of a part from a machine when the next required machine is free. Transportation follows the part release. Between processing and transportation, waiting may occur.

A Petri net model of this situation is displayed

in Figure 3. Processing (t_{ij}), set-up (S_{ij}), and transportation (T_{ij}) times are shown. A decision-free

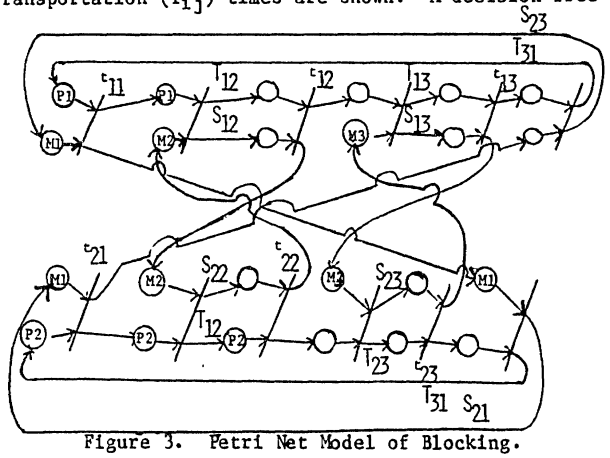

Figure 3. Petri Net Model of Blocking.

Petri net is obtained. Any feasible marking of five tokens guarantees a live and safe Petri net.

Notice that a particular machine set-up can start when transportation begins for the next part to be processed; both activities begin when a part is released from the machine in question. Hence, both release transitions and the corresponding successor processing transitions can be combined into one transition, as shown in Figure 4. Firing time for each combined transition is set equal to the processing time plus the maximum of the appropriate set-up and transportation times. Such reductions in the number of required transitions appear to generalize for systems with limited storage capacity.

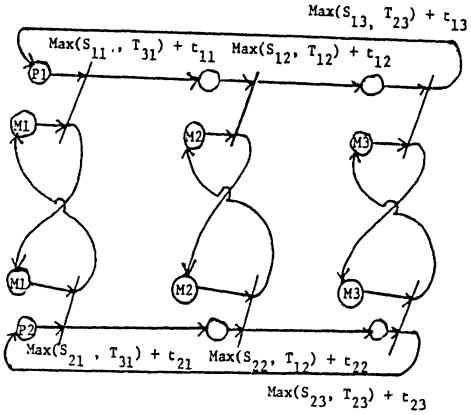

Figure 4. Reduced Petri Net Model of Blocking.

3.3 Assembly Processes

Consider the flexible assembly system of Figure 5, consisting of three different machining centers and a load/unload station (L/UL). There are three part types being machined. Cases (on machine 1) and covers (on machine 2) meet and match on a one-to-one basis at machine 3, for additional operations to be performed on the assembled unit. There are two cart types: C1 begins by loading a case at the L/UL station

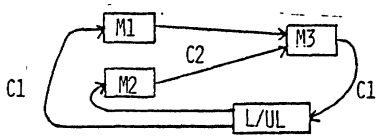

Figure 5. Four-Machine Flexible Assembly System with Two Cart Types.

and automatically transporting the palletized case to M1 in time $TT_{L/UL \to M1}$. Then it travels back to the L/UL to pick up a cover and bring it to M2 in time $TT_{L/UL \to M2}$. Then cart C1 travels to M3, in time $TT_{M2 \to M3}$, to unload the finished, assembled unit, transport it to the L/UL station, where it is taken off and a new case is loaded to begin cart 1's cycle again; C2 begins at machine 1 to unload a case and bring it to machine 3 in time $TT_{M1 \to M3}$. Then C2 travels (empty) to machine 2 to unload a cover and bring it to machine 3 ($TT_{M3 \to M2} + TT_{M2 \to M3}$). Finally C2 returns to the first machine to begin its cycle again.

The Petri net model of this system is provided in Figure 6. Notice that travel times, processing times,

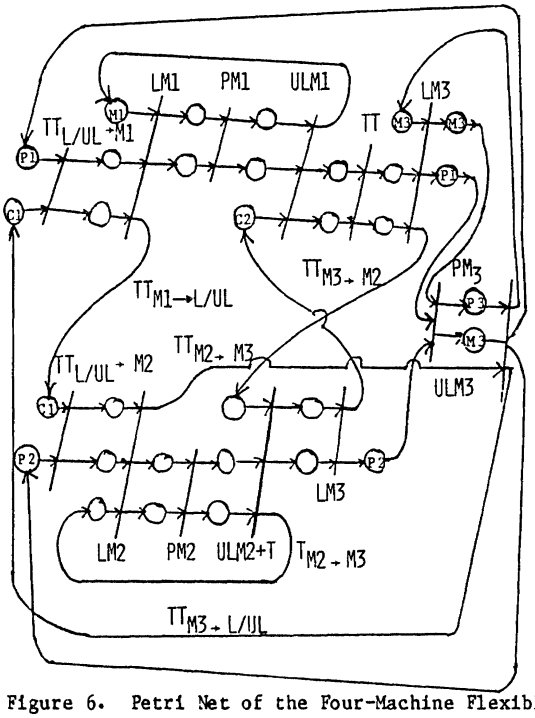

Figure 6. Petri Net of the Four-Machine Flexible Assembly System.

and even loading and unloading times (set-up times, if the set-up at a machine tool is not automated) are included. In this model, the tokens are continuously disappearing (into the assembled unit) and then reappearing (as the raw castings for a case and a cover). Hence this Petri net is not strictly conservative. Non-conservative Petri nets seem to be specific to assembly processes. However, for the obvious distributions of 6 or 7 tokens in the initial marking, the Petri net is safe and live. We have constructed this model to be decision-free, also. Finally, notice that we did not explicitly require places to model the L/UL station.

3.4 FCFS Control Rule

All of the previous Petri nets were decision-free: all tasks were explicitly ordered a priori. In many situations, however, the ordering of parts on machines is specified by means of priority rules. The FCFS rule is often assumed (for example, in product form queueing networks having exponential servers).

Consider a service station with a buffer consisting of n-1 consecutive places. To model the FCFS queue discipline requires n transitions, where transition i means "enter place i in the buffer", for i=1,...,n. The last (n-th) transition is the service. Time T_k is required to reach place i, where i=1,...,n and k=1,..., number of customers. The Petri net is given in Figure 7. There are n-1 tokens required for the buffer places, one token for the server, and tokens for the customers. Customers are generated according to some arrival process. Customer tokens can be input when-

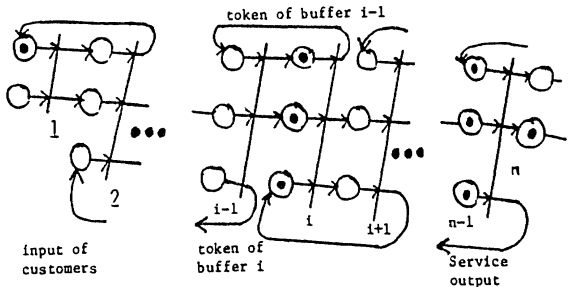

Figure 7. Petri Net of the FCFS Queue Discipline for n-1 Buffers.

ever the first customer place is free. Transition i ($\varepsilon[2,\ldots,n]$) fires only when:

1) a customer is waiting in the (i-1)st buffer (hence the tokens are in the input places of transition i); and
2) the i-th buffer is free (hence its token is also in the input place of transition i).

Again, the Petri net is safe, conservative, live, and decision-free.

4. ANALYSIS OF PETRI NET MODELS

The fixed-route FMS presented in §3.1 has been modeled equivalently by a set of "linear" state equations in the sense of a (max,+) algebra in [1]. The main appeal of the algebraic representation lies in the existence of very efficient algorithms to evaluate system performance. This is a significant advantage over time-consuming and expensive simulation, which is usually required to obtain the same information.

Petri nets have been used to plan a well-designed simulation. The simulation run consists of letting tokens flow through the Petri net structure and collecting statistics on the results. For example, a simulation of the FMS described in §3.1 that is starting with an empty system, produces the Gantt chart of Figure 8.

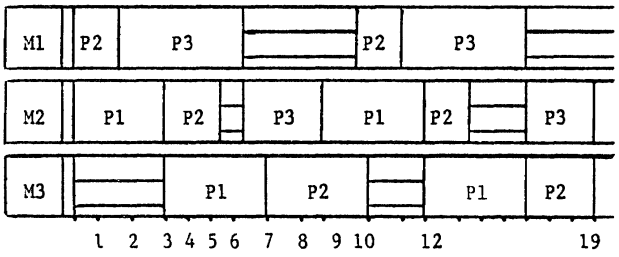

Figure 8. Schedule for the Fixed-Route FMS.

Notice that the system has a periodical behavior, with a period of 19 time units. The same result is obtained analytically in Cohen et al. [1]. A set of critical tasks and resources is also provided as determined by the critical circuit. The results indicate that:

1) The critical resources are the second machine and pallet type 2;
2) The bottleneck machine is M2 and is underutilized;
3) The utilization of the bottleneck machine can be improved only by adding a pallet of type 2;
4) The period (minimum cycle time) is 9.5, which provides the maximum production rate as the reciprocal.

In addition, all machine utilizations are provided.

As demonstrated by the simple example in Figure 8, the simulation gave the periodicity result rapidly, in fact immediately. However, in more realistic and more complex situations, this will not be the case. The duration of the transient period that precedes the periodical behavior depends both on the initial state and the lengths of the sub-critical circuits of tasks. It may be very long.

As an alternative to simulation, the algebraic approach presented in Cohen et al. [1] is independent of the length of the transient period and, in addition, directly calculates the periodical steady state minimum cycle time.

The translation of a particular Petri net model into algebraic equations is straight-forward when the Petri net is safe and decision-free. Each transition is then characterized by the meeting of a specified set of tokens, one per input place. For example, in the FMS example of Figure 2, the transition requiring one time unit occurs when the machine 1 token meets the pallet 2 token. All other transitions are similarly enabled.

The description variables of the algebraic representation can be the dates x_j when transition t_j fires. Denoting $\Gamma^-(j)$ as the set of transitions t_k such that there is a place p_i which is an output of t_k and an input of t_j, we have

$$x_j \geq \max_{k\varepsilon\Gamma^-(j)} x_k + T_k, \qquad (4)$$

where T_k is the duration of transition t_k. The variable x_j can also be conditioned by the availability date of one of its input tokens, as prescribed by the initial marking. Equations similar to (4) can be specified for all transitions. These can then be analyzed via efficient algebraic techniques [1]. If the Petri net is strongly connected, we can specify a linear, (max,+), recurrence equation, $Y(n)=Y(n-1)A$. The dimension of matrix A is the number of tokens and $y_r(n)$ is the date when token r reaches a prescribed place in its circuit for the nth time. The real complexity of a safe, decision-free, strongly connected, Petri net is the number of its tokens (which remains constant throughout--see §1).

Further research is required to investigate performance evaluation algorithms for other classes of Petri nets, such as the non-conservative ones that describe assembly processes. In addition, the safeness property is not guaranteed for a strongly connected, decision-free Petri net. For such decision-free Petri nets, Ramamoorthy and Ho [7] suggest a rudimentory procedure that finds the minimum cycle time by totally enumerating all circuits. However, no additional useful information is provided, such as the order of periodicty or the determination of critical resources (see [1]).

Another analysis technique for timed Petri nets is that suggested by Sifakis [10], which captures the evolution of successive markings. In addition to depending on the structure of the Petri net and its firing rules, it appears that the times when firing occur must be known. This assumption is not made with the (max,+) algebraic technique: the firing times are found. The framework of [10] is more similar to that of vector addition systems [4].

5. POTENTIALS AND LIMITATIONS OF PETRI NET MODELS TO ANALYZE CONTROL PROBLEMS OF PRODUCTION SYSTEMS

There appears to be not much of a limit to the modeling capabilities of Petri nets. However, at present, the decision-free requirement appears to be un-

avoidable to use analytical techniques to analyze the performance of Petri nets. In theory, all sequencing on machines must be known. In addition, the problems of optimizing control rules remains unsolved. On the other hand, it is now possible to efficiently check the performance of a particular control strategy.

Petri nets appear to be a very general modeling tool that is useful for representing discrete-event systems by means of several primitive symbols (places, transitions, and tokens), which are simply interpreted. Systematic system investigation is allowed by varying parameters. Petri nets model a set of events and the conditions which cause the occurrence of events. This provides at least a useful design tool to ensure a well-defined simulation.

Because of the simplicity of the primitives, one might conjecture the existence of efficient means to analyze discrete event systems. This expectation is founded as reported in §4 for the subclass of safe, decision-free Petri nets.

Petri net modeling and analysis can provide insight into the real-time behavior of FMSs. Some issues that can be addressed include the determination of:

i) optimal buffer size via the knowledge of the maximum buffer occupation;
ii) optimum pallet distribution;
iii) appropriate priority rules;
iv) material handling system capacity and performance;
v) critical tasks and resources;
vi) machine utilizations (dynamical, not just the steady-state mean values);
vii) appropriate input sequence of parts;
viii) operation sequence at each machine tool.

In contrast with the type of information obtained by queueing networks, Petri nets provide dynamical information, over a finite time horizon, concerning the evolution of the states of a system.

6. SUMMARY AND FUTURE RESEARCH

A new model has been proposed that can be much more efficient than simulation, to help the investigation of dynamical, real-time control problems of FMSs. The state-of-the-art in Petri net modeling capabilities has been extended. Because of the new application of Petri nets, new modeling conventions needed to be developed. In addition, with the equivalence of a subclass of Petri nets to a linear (max,+) algebraic representation, the existence of efficient analysis techniques was noted.

Future research includes further extending the modeling capabilities, for example, to efficiently model groups of pooled machines [13], or queue disciplines in addition to FCFS. Other plans include the determination of systematic rules for translating Petri nets into the (max,+) equations. Larger classes of Petri nets that can be expressed in the algebraic setting should be investigated. Perhaps non-safe, conservative, decision-free Petri nets can be so represented. Other techniques to analyze Petri nets will be investigated. The complementary nature of Petri nets and queueing networks shall be explored. Procedures that derive appropriate (perhaps "optimal") periodic sequences of parts on machine tools, given FMS priority rules, will be investigated. The corresponding Petri net model would be equivalent to a multiclass, deterministic, closed queueing network with finite buffers. The real-time planning and control issues that were summarized in §5.2 will be investigated. Finally, Petri nets will be used to model the hierarchical computer control system for FMSs. The same model can represent both the physical and control systems.

REFERENCES

1. Cohen, G., Dubois, Didier, Quadrat, J. P. and Viot, M., "A Linear-System-Theoretic View of Discrete-Event Systems", Proceedings of the 22nd IEEE Conference on Decision and Control, San Antonio TX (December 14-16, 1983).

2. Cuninghame-Green, Raymond, Minimax Algebra, Springer-Verlag, Berlin (1979).

3. Diaz, Michel, "Modeling and Analysis of Communication and Cooperation Protocols Using Petri Net Based Models", Computer Networks, Vol. 6, No. 6, pp. 419-441 (December 1982).

4. Gelenbe, Erol, "Stationary Properties of Timed Vector Addition Systems", in Deterministic and Stochastic Scheduling, Dempster et al., editors, D. Reidel Publishing Company, Dordrecht, Holland, pp. 223-232 (1982).

5. Ho, Yu Chi and Cao, Xiren, "Peturbation Analysis and Optimization of Queueing Networks", Journal of Optimization Theory and Applications (1983).

6. Peterson, James L., Petri Net Theory and the Modeling of Systems, Prentice-Hall, Inc., Englewood Cliffs NJ.

7. Ramamoorthy, C. V. and Ho, Gary S., "Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets", IEEE Transactions on Software Engineering, Vol. 6, No. 5, pp. 440-449 (September 1980).

8. Ramchandani, Chander, "Analysis of Asynchronous Concurrent Systems by Timed Petri Nets", Ph.D. dissertation, Department of Electrical Engineering, MIT, Cambridge MA (1974).

9. Shapiro, R. M. and Saint, H., "A New Approach to Optimization of Sequencing Decisions", Annual Review of Automatic Programming, Vol. 6, No. 5, pp. 257-288 (1970).

10. Sifakis, J., "Use of Petri Nets for Performance Evaluation", Acta Cybernetica, Vol. 4, No. 2, pp. 185-202 (1979).

11. Solberg, James J., "Analytical Performance Evaluation of Flexible Manufacturing Systems", Proceedings of the 18th IEEE Conference on Decision and Control, San Diego CA, pp. 640-644 (December 1979).

12. Stecke, Kathryn E. and Solberg, James J., "Loading and Control Policies for a Flexible Manufacturing System", International Journal of Production Research, Vol. 19, No. 5, pp. 481-490 (September-October, 1981).

13. Stecke, Kathryn E. and Solberg, James J., "The Optimality of Unbalancing Both Workloads and Machine Group Sizes in Closed Queueing Networks of Multi-Server Queues", Working Paper No. 322, Division of Research, Graduate School of Business Administration, The University of Michigan, Ann Arbor MI (November 1982).