Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

PETRI NET BASED SIMULATION OF CONTROLS FOR A COMPUTER-INTEGRATED ASSEMBLY CELL

KELWYN A. D'SOUZA

Department of Management
School of Business
Hampton University
Hampton, Virginia 23668, U.S.A.

SURESH K. KHATOR

Department of Industrial and Management
Systems Engineering
University of South Florida
Tampa, Florida 33620, U.S.A.

ABSTRACT

In this paper, Petri Nets are used to model controls for a manufacturing plan processed in a Computer-Integrated Assembly Cell. The Control Model is automatically generated from the manufacturing plan and used to detect deadlocks at the workstations. Deadlocks are avoided by re-allocating buffer capacities at these critical workstations. The Control Model is simulated for an application to demonstrate deadlock detection and avoidance functions. Results show that re-allocating buffer capacities at the critical workstations avoid deadlocks and improve the overall performance of the cell. Control Model implementation could possibly reduce costs involved in writing, debugging, and maintenance of programs for real-time industrial controllers.

1 INTRODUCTION

A Computer-Integrated Assembly Cell (CIAC) processes a manufacturing plan consisting of a set of jobs. Each job is concurrently assembled at workstations according to the process plan. Deadlocks occur in a CIAC when a set of concurrent processes become interlocked in such a way that they cannot be completed. Deadlock detection is difficult unless the cell's programmable electronic controller (PEC) is programmed for a manufacturing plan and tested. This is time consuming, unsafe, and decreases overall cell utilization. In view of the nondeterministic characteristic of concurrent systems there is no guarantee that the program will function correctly through the entire production run.

A Control Model of a manufacturing plan must be analyzed prior to the start of production to detect deadlocks. The Control Model is developed using the theory of Petri Nets (PNs). Deadlocks detected in a manufacturing plan are avoided by re-allocating buffer capacities at critical workstations resulting in an improved performance. Further organization of the paper follows:

In Section 2 we present the architecture of a CIAC and the deadlock problem. Section 3 discusses the need for a Control Model and the common modeling tools. In Section 4 we develop the Control Model for a manufacturing plan scheduled on the CIAC. The simulation results obtained using a Petri Net software package is presented in Section 5. Finally in Section 6 we present our conclusions and further research plans.

2 ARCHITECTURE OF A COMPUTER-INTEGRATED ASSEMBLY CELL

The layout of a Computer-Integrated Assembly Cell (CIAC) is shown in Figure 1. Assembly workstations (WS_1, WS_2, .., WS_5), having input buffers (I) and output buffers (O), are arranged around a common transporter (TR). A manufacturing plan is prepared consisting of a list of jobs to be executed concurrently in the CIAC under the control of a programmable electronic controller. Each job consists of a batch of parts to be assembled according to a process plan. At the transfer workstation (WS_1 in Figure 1), pallets holding base assemblies are loaded on transport carts. The carts then circulate between the other workstations to fulfil assembly requirements. Once in process, the total number of pallets circulating continuously in the system are constant.

On arrival at a workstation, a free transfer mechanism allows assemblies to be removed from the cart and stored at the input buffer (I). Upon completion of the workstation operation, the same mechanism allows temporary storage of the assembly at the output buffer (O) before it is transferred to the cart to move to its next required location. Finally, the completed assembly is unloaded at the transfer workstation and leaves the system.

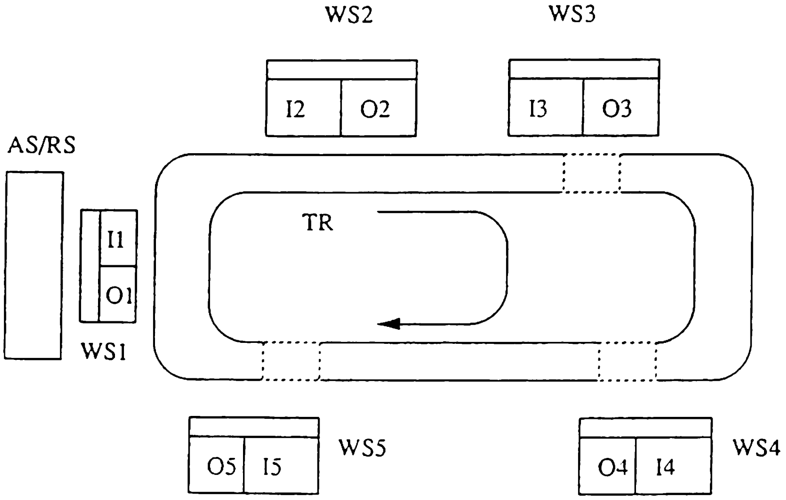


Figure 1: A Computer-Integrated Assembly Cell (CIAC)

2.1 The Deadlock Problem

To maximize cell utilization, multiple assembly jobs are scheduled concurrently without consideration for the complex interaction between workstations. At the control design stage, production engineers are required to develop programmable electronic controller programs according to the operation sequence specified by the process plan. Programming frequently uses ladder logic, making it difficult to comprehend a program and verify the controller operation at that stage (Murata et al, 1986).

A manufacturing plan processed in a CIAC may consist of a single job or multiple jobs sequenced to a workstation more than once, with priority given to new assemblies entering the cell. Such a manufacturing plan causes deadlocks as batch size is increased. In this paper, we consider manufacturing plan consisting of a single job whose process plan requires routing through two workstations (Refer to Figure 1) in the following sequence:

WS_1, WS_2, WS_1.

Since there is a single transporter always available, and a single workstation in each group, the sequence can be represented by input/output buffers (Banaszak and Krogh, 1990) as

$\{I_1, O_1, I_2, O_2, O_1, I_1\}$

(Note that the input and output buffers of WS_1 are interchangeable).

Deadlock condition occurs when the next buffer required by an assembly is currently unavailable, since it is being held by another assembly in the production sequence. For example, if buffers of WS_1 are busy with assemblies, WS_2 will not be able to release any of its assemblies to WS_1. According to Coffman et al (1971) the "circular wait" condition for deadlock occurrence is satisfied in this case.

Deadlock causes the system to come to a complete stop, until some action is taken by the production engineer. One way to break the deadlock is to remove newly arrived assemblies from output buffer O_1 of WS_1, and allow assemblies from WS_2 to proceed to WS_1. When the number of workstations is large, it becomes difficult to locate the workstation causing deadlocks. If such a condition can be detected in advance, a reallocation of buffer capacity at workstations could be planned for avoiding deadlocks.

3 NEED FOR A CONTROL MODEL

Deadlock conditions for a manufacturing plan processed in a CIAC can be detected off-line by a Control Model. The Control Model represents a mathematical formalism of the manufacturing plan, reflecting specified system behavior under varying operating conditions. Mathematical models for deadlock detection and avoidance in computer operating systems have been studied extensively by Coffman et al (1971), and Habermann (1969). However, such models are not directly applicable to manufacturing systems, where resources are allocated to a process in a sequence specified by the process plan. There is clearly a need to develop similar models for a CIAC, using an appropriate tool.

3.1 Modeling Tools for Automated Manufacturing Systems

Modeling tools discussed below have been developed for analysis of Automated Manufacturing Systems (Suri, 1985). It is essential to select an appropriate tool for modeling the controls of a CIAC which captures the concurrent and asynchronous characteristic of operations.

3.1.1 Simulation

Simulation is a numerical technique for analyzing the performance of a system evolving in time. Simulation is frequently employed when the complexities in the system makes it hard to derive detailed mathematical modeling relationships. It is used extensively for evaluating the performance of manufacturing systems. It provides reasonably accurate and detailed information about the system's performance, but requires a large amount of programming time, input data, and computer time to run the simulation model (Suri, 1985; Stecke and Solberg, 1981).

3.1.2 Queuing Networks

Queuing Networks (QNs) are widely used for modeling Automated Manufacturing Systems as a closed network of queues. The network consists of individual servers, corresponding to workstations, through which there is a flow of jobs demanding service. QNs provide reasonably accurate information about the average system behavior under steady-state conditions. This method becomes complex when conditions are not satisfied leading to approximate results.

3.1.3 Petri Nets

Petri Net (PN) modeling is a powerful tool for modeling and analysis of asynchronous, concurrent systems which are difficult to model using simulation and QNs. It is convenient to model non-product form characteristics, such as multiple workstation holding, blocking, synchronization, and prioritization, which are common in a CIAC (Viswanadham and Narahari, 1988).

Researchers have modeled Automated Manufacturing Systems using PNs (Al-Jaar and Desrochers, 1988). PN models in the common graphical form offer a good visual description, and are convenient for modeling controls of a CIAC. Such models become unmanageably complex to develop and analyze for large systems (Viswanadham et al, 1990). Researchers have developed a simplified approach of recasting a PN in matrix form (Hack; 1974; Peterson, 1981; Saha and Bandyopadhyay, 1988). Deadlock detection is efficiently carried out on this form of PN using matrix algebra. Control Models developed using this approach result in a standardized PN generation and analysis procedure.

4 CONTROL MODEL DEVELOPMENT AND APPLICATION

Control Models presented in this paper (Refer to Figure 2) are automatically generated from the process specifications using an algorithm, and have the following functions:

They represent the manufacturing plan in PN formalism, encompassing flow of jobs according to the process plans.

They analyze the PN for properties leading to deadlocks.

They evaluate the performance of a deadlock-free PN.

The Control Model utilizes some of the basic concepts and terminologies of PNs. It is assumed

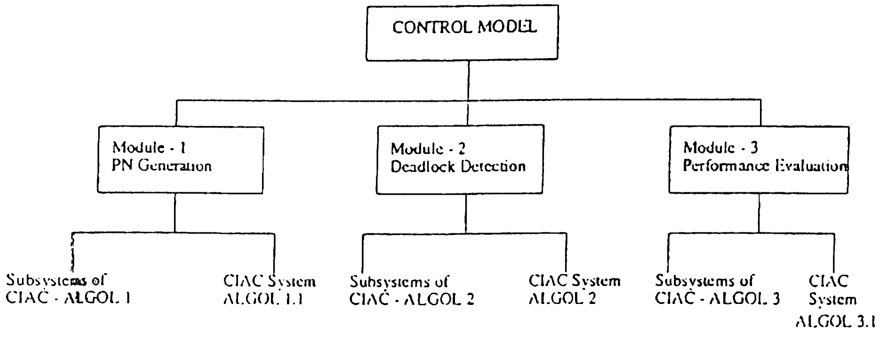


Figure 2: Overview of Control Model

that the reader has an understanding of the concepts which are well covered in Peterson (1981) and Murata (1989). Due to space limitation, detailed definitions of the concepts have been omitted.

4.1 PN Generation and Deadlock Detection

A manufacturing plan scheduled on the CIAC is automatically converted into a PN in MDF using ALGOL 1.1 (Figure 2) as follows:

$$MDF=(C^{+},C^{-},M_{0}).$$

$$C^{+}=[c^{+}_{ij}]_{n\times m}$$

is the output matrix, where

$$c^{+}_{ij}=1 \ \ if \ p_{j}\in t^{\bullet},$$

0 otherwise.

$$C^{-}=[c^{-}_{ij}]_{n\times m}$$

is the input matrix, where

$$c^{-}_{ij}=1 \ \ if \ p_{j}\in {}^{\bullet}t,$$

0 otherwise.

$$1\leq i\leq n;\ 1\leq j\leq m.$$

$$C=[C^{+}-C^{-}]_{n\times m}$$

is the transition-to-place incidence matrix of MDF.

M_0= Initial state of the system.

The PN generated for the manufacturing plan is analyzed for deadlocks using ALGOL 2 (Figure 2). At each state (M_k),the $C^+$and C^-matrices are checked for enabled transitions. Firing the enabled transition generates a new state (M_{k+1}) which is determined by the state equation (Murata, 1989). Deadlocks in a PN occur due to the absence of tokens in the sets of places at a given state. The steps in ALGOL 2 checks places at each state, and determines the sets of places that cannot enable transitions (D'Souza, 1991). For the PN of a manufacturing plan processed in a CIAC, absence of tokens in the process places or buffer places results in deadlocks. In the physical system, this occurs due to restriction of the buffer capacity at workstations. When workstation buffer capacity becomes full with assemblies, the "circular wait" condition is satisfied, and no further movement of assemblies takes place.

4.2 Control Model Application

The Control Model is applied to a range of manufacturing plans processed in the CIAC (D'Souza, 1991). In this Section, only application to a manufacturing plan consisting of a single job is presented. The job contains a batch to be assembled at a set of workstations, according to the process plan. For a job containing a batch size of three, the PN is generated using ALGOL 1.1 as

$$MDF=(C^{+},C^{-},M_{0})$$

where $C^+$and C^-matrices are shown in Figure 3. The initial state is

M_0= (3000000010101010).

C^+ Matrix

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15
t0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
t1	0	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0
t2	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0
t3	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1
t4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
t5	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0
t6	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

C^- Matrix

	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15
t0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
t1	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0
t2	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
t3	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
t4	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
t5	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0
t6	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0

Figure 3: Output and Input Matrices for a CIAC (Two Workstations)

The PN is analyzed for deadlocks using ALGOL 2. Goal state is set for all completed assemblies to be in the final process place $p_7$of the production sequence as

$$M_g = (0000000310101010).$$

Deadlock is detected by ALGOL 2 at state $\mathbf{M_9}$. Sets of places identified at state $\mathbf{M_9}$by ALGOL 2 are as follows:

$$\{p_0,p_8\}\rightarrow t_0, \{p_2,p_{11},p_{12}\}\rightarrow t_2,$$

$$\{p_3,p_{13},p_{14}\}\rightarrow t_3, \{p_4,p_{10},p_{15}\}\rightarrow t_4,$$

$$\{p_5,p_8,p_{11}\}\rightarrow t_5.$$

None of the above sets have tokens in all places. Therefore, no transition is enabled, and deadlock occurs.

Deadlocks are avoided by increasing tokens at the sets of places identified by ALGOL 2. In the physical system, this involves re-allocating buffer capacities at workstations. There are several buffers that could be re-allocated. The production engineer needs to identify the correct buffers to be selected. Selecting a wrong buffer for capacity increase could result in future deadlocks, and lower overall performance.

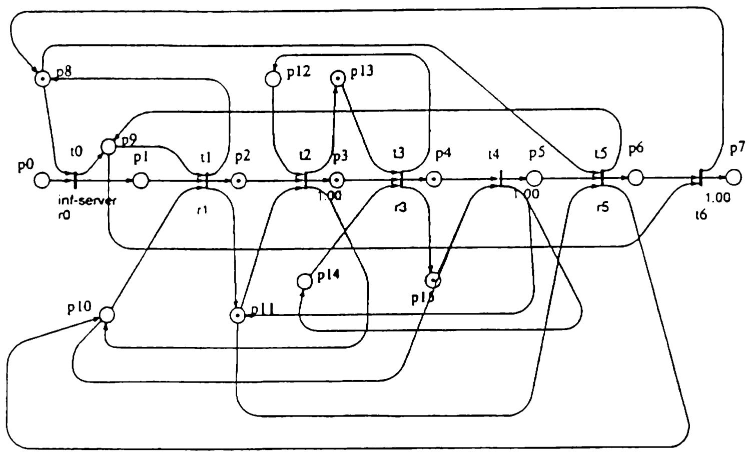
A Buffer Re-allocating Algorithm (D'Souza, 1991) selects the buffer place for increasing tokens. The Plan examines all sets of places at the deadlock state, and selects the set to which additional tokens are to be provided. Simulation of the Control model is required to corroborate increased performance due to buffer re-allocation.

5 SIMULATION OF THE CONTROL MODEL

The simulation of the Control Model is in the GreatSPN 1.5 software package (Chiola, 1990). The PN model is first constructed on the screen of a graphical workstation. Structural properties, such as the place and transition invariants computed by the package, are used to check its correctness. The model is checked for deadlocks by playing the "token game." Starting from the initial state, enabled transitions are made to fire until the final state is reached. Then, the performance measures are generated using assigned transition times for analyzing the operating characteristic (OC) under varying conditions.

5.1 Deadlock Detection and Avoidance

The PN in graphical form, representing the simulated deadlock state, is shown in Figure 4. Place $p_0$initially has three tokens (assemblies)



Places
p_0 - assembly available for processing
p_1 to p_6 - production flow sequence
p_7 - completed assemblies
p_8 - input buffer of WS_1 free
p_9 - input buffer of WS_1 full
p_{10} - output buffer of WS_1 free
p_{11} - output buffer of WS_1 full
p_{12} - input buffer of WS_2 free
p_{13} - input buffer of WS_2 full
p_{14} - output buffer of WS_2 free
p_{15} - output buffer of WS_2 full

Transitions
t_0 - assembly enters system
t_1 - process 1
t_2, t_4, t_6 - transport assembly
t_3 - process 2
t_5 - process 3

Figure 4: Deadlock State for a Single Job Assembled on Two Workstations

available for processing. The initial state of the system is

$\mathbf{M}_0$ = (3000000010101010).

Transitions fired in a sequence

$\sigma = (t_0t_1t_0t_2t_1t_0t_3t_2t_1)$

result in a "dead marking," indicating a deadlock, and denoted by state

$\mathbf{M}_9$ = (0011100010010101).

No tokens are enabled at this state. From Figure 4, it is noted that tokens (assemblies) have occupied input buffer place (p_{13})and output buffer place (p_{15})of WS_2, and output buffer place (p_{11}) of WS_1respectively. Buffer status represented by places p_{11},p_{13},and $p_{15}$are now busy. The real cause of deadlock is transition t_4, which is unable to fire since there is no token in p_{10}. For $p_{10}$to get a token, $t_2$must fire. But $t_2$cannot fire, since there is no token in p_{12}. The "circular wait" condition for deadlock is satisfied, and is represented by the transition dependency loop formed when:

$t_4$depends on t_2,
$t_2$depends on t_3,
$t_3$depends on t_4.

Following the steps of Buffer Re-allocating Algorithm (D'Souza, 1991), the output buffer capacity of workstation 1 is increased to avoid deadlocks for the single job containing multiple batch sizes. It needs to be ascertained if this reallocating policy avoids deadlocks and improves performance for a range of jobs to be assembled by the CIAC system.

5.2 Results of Simulation

OC Curves shown in Figures 5 indicate the throughput rate of the system at varying input rates (r_0). Deadlock is detected by ALGOL 2 in Sec tion 4.2 for a batch size of three assemblies. Increasing batch size beyond three assemblies has no effect on throughput rate, since deadlock continues to occur unless buffer capacities at critical workstations are re-allocated by the Deadlock Avoidance Plan. The Deadlock Avoidance Plan re-allocates buffer capacities at selected workstations called critical workstations. Re-allocating buffer capacities at critical workstations avoids future deadlocks, and increases the overall performance of the system. It must be noted that, although several sets of places are identified by ALGOL 2 at the deadlock state, the Buffer Re-allocating Algorithm only selects a subset of these places.

Increasing buffer capacities at critical workstations is expected to avoid deadlocks, and improve the overall performance of the CIAC system. Throughput rates computed for the samples are statistically analyzed to show that the selected buffer capacity results in improved overall performance (D'Souza 1991).

The Buffer Re-allocating Algorithm enables transition t_4, by increasing tokens at place p_{10}, i.e., increasing buffer capacity at O_1. The OC Curves obtained by independently increasing buffer capacity of O_1,I_2,and $O_2$to two are shown in Figure 6. Increasing buffer capacity of I_2 results in a lower throughput rate, as compared to increasing buffer capacity of $O_1$and O_2. Increasing O_2produces an increased throughput rate at higher input rates.

6 CONCLUSIONS

Deadlocks pose a serious problem in concurrent operating systems such as the CIAC resulting in the potential for complete stoppage of the cell. This problem can be avoided by analyzing manufacturing plans off-line using a Control Model. In this paper, the Control Model was generated using PN formalism, and applied to a manufacturing plan processed by the CIAC system. Deadlocks detected by the Control Model were avoided by re-allocating buffer capacities at the critical workstations.

The approach followed for Control Model generation, deadlock detection, and avoidance was heuristic. Using PNs, it was shown that the Buffer Re-allocating Algorithm avoids deadlock states. Improved performance due to buffer re-allocation at the critical workstation was corroborated by simulation.

The Control Models developed in this research are to be extended to multiple jobs in future research, and validated under varying operating conditions and buffer capacities. Buffer re-allocation plans are to be analyzed under these conditions, to determine the effective buffer capacities at critical workstations.

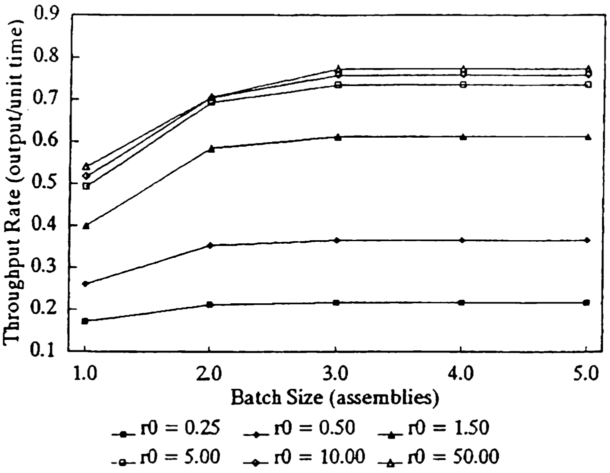


Figure 5: OC Curves for Performance

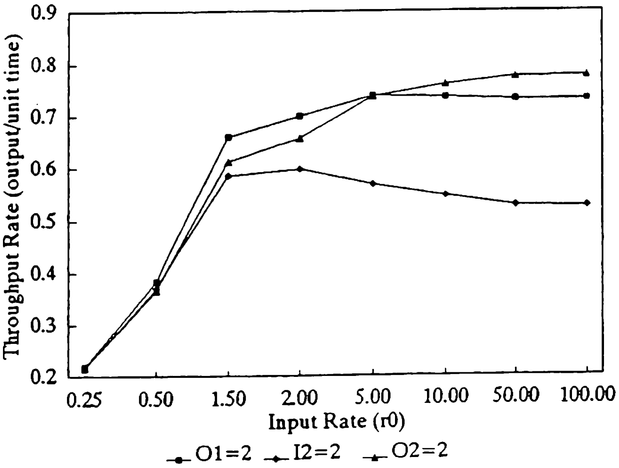


Figure 6: OC Curves for Buffer Re-allocation

REFERENCES

Al-Jaar, R. Y., and A. A Desrochers. 1988. A Survey of Petri Nets in Automated Manufactur ing systems. In Proceedings of the 12th Inte-rnational Association for Mathematics and Computers in Simulation World Conference, ed. R. Wichnevetsky, P. Borne, and J. Vigness, 503-510.

Banaszak, Z. A., and B. H Krogh. 1990. Deadlock Avoidance in Flexible Manufacturing Systems with Concurrently Competing Process Fl ows. IEEE Transactions on Robotics and Automation 6:724-734.

Chiola, G. 1990. The GreatSPN 1.5 Software Architecture. Torino, Italy: Dipartimento di Informatica Università di Torino corso svizzera 185, 10149.

Coffman Jr., E. G., Elphick, M. J., and A. Shoshani. 1971. System Deadlocks. Computing Surveys 3:67-78.

D'Souza, K. A. 1991. Modeling Controls for a Computer-Integrated Assembly Cell using Petri Nets. Unpublished Ph.D. Dissertation, Industrial and Management Systems Department, University of South Florida, Tampa, Florida.

Habermann, A. N. 1969. Prevention of System Deadlocks. Communications of the ACM 12: 373-385.

Hack, M. 1974. The Recursive Equivalence of the Reachability Problem and the Liveness Problem for Petri Nets and Vector Addition Systems. In Proceedings of the 15th Annual IEEE Symposium on Switching and Automata Theory, 156-164.

Murata, T. 1989. Petri Nets: Properties, Analysis and Applications. In Proceedings of the IEEE 77:541-580.

Murata, T., Komoda, N., Matsumoto, K., and K. Haruna. 1986. A Petri Net-Based Controller for Flexible and Maintainable Sequence Control and its Applications in Factory Automation. IEEE Transactions on Industrial Electronics IE-33:1-8.

Peterson, J. L. 1981. Petri Net Theory and the Modeling of Systems. Englewood Cliffs, New Jersey: Prentice Hall Inc.

Saha, B., and S. Bandyopadhyay. 1988. Representation and Analysis of Petrinets via the Matrix State Equation Approach. International Journal of Electronics 65:1-7.

Stecke, K. E., and J. J. Solberg. 1981. Loading and Control Policies for a Flexible Manufacturing System. International Journal of Production Research 19:481-490.

Suri, R. 1985. An Overview of Evaluative Models for Flexible Manufacturing Systems. Annals of Operations Research 3:61-69.

Viswanadham, N., and Y. Narahari. 1988. Stochastic Petri Net Models for Performance Evaluation of Automated Manufacturing Systems. Information and Decision Technologies 14:125-142.

Viswanadham, N., Narahari, Y., and T. L. Johnson. 1990. Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems Using Petri Net Models. IEEE Transactions on Robotics and Automation 6:713-723.

AUTHOR BIOGRAPHIES

KELWYN A. D'SOUZA is an Assistant Professor in the Department of Management at Hampton University. He received his Ph.D. in Industrial Engineering from the University of South Florida in 1991. His research interests are focused on performance modeling and analysis of Automated Manufacturing systems using Petri Nets. He is an associate member of the Society of Manufacturing Engineering, and Institute of Industrial Engineers.

SURESH K. KHATOR is an Associate Professor in the Department of Industrial and Management Systems Engineering at the University of South Florida. He received his Ph.D. in Industrial Engineering from Purdue University in 1975. His research interests include modeling and analysis of manufacturing systems, plant facilities design, and simulation. He is a senior member of the Institute of Industrial Engineers and a registered professional engineer in the State of Florida.