Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

MODELING AND CONTROL OF DEADLOCKS IN A FLEXIBLE MACHINING CELL

KELWYN A. D'SOUZA
School of Business
Hampton University
Hampton, VA 23668
U.S.A.

ZBIGNIEW BANASZAK
Faculty of Science
Kuwait University
Safat 13060
KUWAIT.

ROBERT WOJCIK
Technical University
of Wroclaw
Wroclaw
POLAND.

ABSTRACT

In recent years, there has been a growing need for Flexible Machining Cells to cope up with the demand for small and medium-batch products. Controlling processes in a Flexible Machining Cell is more difficult compared to conventional manufacturing systems. This is primarily due to the concurrent and asynchronous interaction of events which cause system deadlocks. An approach to model the controls of a cell for a given process specification is presented. Petri Nets was used as the modeling formalism. An algorithm is used to test the model for deadlock occurrences. The results obtained from this research indicates the scope for designing adaptive control procedures capable of on-line applications.

1 INTRODUCTION

Flexible Machining Cells (FMCs) are a relatively new class of automated manufacturing systems, consisting of a group of computer controlled machining centers serviced by automated handling devices. The cell is designed to manufacture a wide variety of parts simultaneously. Parts enter the cell at a loading station, and are routed through a sequence of operations at the machining centers. The completed part is finally unloaded at an unloading station. The machining operations and part movements are controlled by a host computer through a Programmable Logic Controller (PLC).

A key advantage of a FMC over conventional manufacturing systems is the flexibility feature, which allows a variety of parts to be manufactured simultaneously. This feature results in a number of planning, scheduling, and control problems during the manufacturing cycle. Sophisticated planning, scheduling, and control models and procedures have been suggested to optimize the throughput from the system. An important control problem, which affects throughput of the FMC are detection and avoidance of system's deadlocks. Deadlock occurs when a group of concurrent processes become interlocked in such a way that they cannot be completed. Deadlock occurs primarily due to the concurrent nature of activities involved in the simultaneous processing of multiple parts (Viswanadham and Narahari 1992). Production engineers are expected to solve these problems while developing PLC programs for a manufacturing plan, to know ahead of time how to eliminate these production bottlenecks.

Thus, in the planning of manufacturing processes of such complex systems it would be useful to develop a model that could predict deadlock occurrences. Petri Nets (PNs) have been proven to be convenient for modeling concurrent and asynchronous activities as well as for the verification of deadlocks (Murata 1989; Peterson 1981). A PN model of admissible control flows of competing processes in a FMC was developed by the authors (Banaszak, Wojcik, and D'Souza 1992). The Reachability Tree was utilized to search all paths in order to set a resource allocation policy that would prevent deadlock states from being reached. Since the tree represents all possible sequences of transition firings, it could become an infinite tree even for a finite reachability set (Peterson 1981).

In this paper, our previous research is extended by the development of a PN Control Model to detect deadlock in the FMC. The model incorporates an algorithm for efficient deadlock detection. The results shows the Control Model to be deadlock-free. Due to space limitations, we have excluded detailed PN concepts. The interested reader can find a comprehensive coverage of the theory and concepts in (Peterson 1981).

2 CONTROL PROBLEMS IN THE FLEXIBLE MACHINING CELL

The FMC (Refer to Figure 1) consists of machine tools M_1 and M_2, industrial robots R_1 and R_2, a gripper

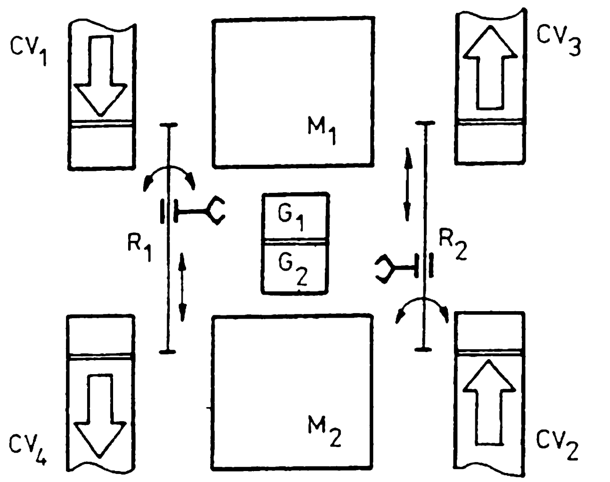


Figure 1: A Flexible Machining Cell. (Banaszak et al 1992).

changing station equipped with grippers G_1 and G_2, and conveyors CV_1 ... CV_4. A machining cycle is initiated when a raw part is available at the input conveyor CV_1 (CV_2). The robot R_1 (R_2) equipped with gripper G_1 (G_2) is activated and loads the part on the machine tool M_1 (M_2) for carrying out machining operation. The machined part is unloaded from the machine M_1 (M_2) by the robot R_2 (R_1) and deposited in the output conveyor CV_3 (CV_4) using gripper G_2 (G_1). The machining cycle of the cell is controlled by the PLC.

Two processes PR_1 and PR_2 are scheduled on the FMC. Let $O_{i,u}$ = order of processing of the i^{th} process to be performed at the u^{th} stage. The process sequence can be stated as:

$$PR_1 = (O_{1,1},\{R_1,G_1\},(O_{1,2},\{M_1\},(O_{1,3},\{R_2,G_2\}) \quad (1)$$
$$PR_2 = (O_{2,1},\{R_2,G_2\},(O_{2,2},\{M_2\},(O_{2,3},\{R_1,G_1\}) \quad (2)$$

At the control design stage, production engineers are required to develop PLC programs according to the operation sequence specified by the process sequence. Ladder logic, the language of PLCs, analyzes process input conditions, makes yes/no logic decisions, and dispatches outputs back to the process. Verification of sequential operations from ladder logic is difficult to comprehend and maintain even for trained engineers (Chocron and Cerny 1980; Murata et al 1986).

During the machining cycle, sharing of the machine tools, robots, grippers and the conveyors by the parts can often result in deadlocks, causing the cell to come to a stop due to a "circular wait" (Havender 1968). The machined part holds onto the machine tool, waiting for the robot to unload it. The robot in turn is being held by a raw part. Both the machined part and the raw part will wait forever until some external action is taken, thus affecting the overall performance of the cell. There are two methods to prevent the deadlock state. Firstly, a storage buffer can be provided to accommodate raw parts while the machine tool is busy. Thus, the robot can be made free to unload finished parts. This requires additional investments and will result in deadlocks when the storage buffer gets filled. Alternatively, the cell's controller can be programmed to prevent the robot from picking a raw part while the machine tool is busy. The latter approach seems practical in the industrial environment where several FMCs are interconnected and deadlock occurrence may not be obviously known in advance.

The conventional method to ascertain if an FMC will run without deadlocks and at the estimated performance level during production cycles is to make pilot runs using test data. This approach unnecessarily extends the period of production startup, and is ineffective in view of the nondeterministic characteristic of the system. Production engineers have to test run controller programs for each manufacturing plan. Still, there is no guarantee that the program will function correctly through the entire production run. A Control Model of a manufacturing plan must be developed and analyzed prior to the start of production to detect deadlocks, and evaluate performance of the cell.

3 BACKGROUND OF CONTROL MODELING

PNs are convenient for modeling controls of an automated manufacturing system due to its ability to graphically represent concurrent and asynchronous activities, and its well developed mathematical foundation for analysis. Developing a PN model for real-time control of large systems has been observed to be tedious, and structure and detail varies depending on the individual analyst. Several researchers have contributed towards development of PN Control Models for automated manufacturing systems (D'Souza and Khator 1993a). Some of their work which is related to this research has been briefly summarized here.

A PN-based controller was implemented by Crockett et al (1987) on a machining workstation consisting of a machining center and parts handling operation. The machining center processes different parts according to the NC programs automatically downloaded from the machine controller. The PN controller uses extensions to model various control procedures in a concise manner. The controller prevents a program from being downloaded and initialized on the machining center while the operator is handling a part. A PN Controller Model for the real-time control of an AMC consisting of a CNC lathe serviced by a robot was developed by Boucher, Jafari and Meredith (1990). PLC events taking place in the AMC were transformed into the PN Controller Model. This model was shown to have an advantage over the conventional PLC diagram, since it provides information to the programmer about the state of the whole system.

Teng and Black (1990) designed a PN model to represent a Cellular Manufacturing System, and proposed to use this model to generate control algorithms. The Reachability Tree approach used for the model analysis becomes complex and time consuming for constructing and analyzing large systems. An attempt to describe an FMS control mathematically was made by Čapkovič (1988). He has presented an algorithm for support of decision making in Flexible Manufacturing System control giving the user a choice of selecting the most suitable control strategy. The model was applied to a NC machine-robot cell and the resulting control strategies were evaluated.

To analyze large systems, it is often convenient to convert the PN model in a matrix form and utilize algorithms for model synthesis. Such an approach was applied to design an adaptive concurrent control software for a Flexible Machining Cell (Banaszak et al 1992). The Control Model automatically generated from the process specifications guaranteed the deadlock-free operations of the cell. A Control Model that combines the functions of PN generation, deadlock detection and performance evaluation was developed for a Computer-Integrated Assembly Cell (D'Souza and Khator 1993b). This model was simulated to corroborate correct functioning under varying operating conditions (D'Souza and Khator 1992).

4 DEADLOCK DETECTION

The FMC processes a manufacturing plan consisting of a set of parts. Each part has an individual process plan to convert product design specifications into logical manufacturing sequence of machine operations. For a given manufacturing plan, understanding the complex interactions of parts and resources requires a Control Model. The Control Model represents a mathematical formulation of the manufacturing plan, reflecting system behavior under varying operating conditions. By analyzing the Control Model off-line, properties leading to deadlocks are detected and the throughput rate is estimated. Such feedback information is useful to the production engineer designing PLC programs since it reduces the overall set-up time.

Consider the FMC described in Figure 1. Input signals from the FMC are received by the host computer indicating availability of part and the readiness of the machine tool and robot. The host computer issues output signals to the machine tool and robot to start the machine cycle. The concurrent activities in the cell are to be controlled to avoid deadlocks.

From equations (1) and (2), a PN model of the control flow developed for the FMC. This model is a modified version of our previous model (Banaszak et al 1992), and is shown in Figure 2. States of the machine tool, robot, gripper, and conveyor are represented by places (P). The start and finish of an event are represented by transitions (T), and directional sequences by arcs ($\rightarrow$). Tokens ($\bullet$) have been added to the model in Figure 2 to represent the current state of the cell. When a tokens resides in a particular place it indicates a condition is satisfied. For example, if a place represents the robot, a token in it indicates that conditions are satisfied for the robot to begin loading or unloading operations.

Having modeled the controls of the FMC as a PN, analysis for deadlocks is required before implementing the PLC program. During operation of the FMC, state **M** is reached as indicated by tokens in the places (P) of Figure 2. Banaszak et al (1992) developed a satisfactory condition for a state (**S**) to be deadlock-free. This was further analyzed by means of the Reachability Tree. The Reachability Tree approach for the model analysis becomes complex and time consuming for constructing and analyzing large systems. A Control Model must be developed for automatic generation of the PN and deadlock detection.

4.1 Control Model for Deadlock Detection

The PN Controller Model developed for the FMC is transformed into a PN in Matrix Definitional Form (MDF). This PN is analyzed for deadlocks resulting from a concurrent demand for resources using the Deadlock Detection Algorithm shown in the Appendix. The algorithm applies some of the basic concepts and terminologies of PN. The MDF of the PN is the graphical form of PN recast in vector and matrix terms as:

$$MDF=(C^+,C^-,M_0).$$

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

is the output matrix, where

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

0 otherwise.

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

is the input matrix, where

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

0 otherwise.

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

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

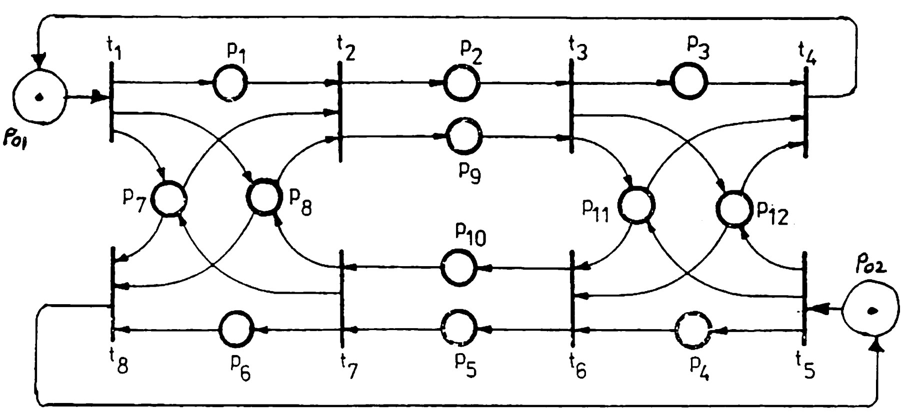
is the transition-to-place incidence matrix of MDF.

$\mathbf{M}_0$ = Initial state of the system.

Deadlock occurs at a stage of a process when a part cannot advance to the next state due to resource restriction. In a PN, deadlock is related to the reachability problem. A reachable marking $\mathbf{M} \in R(\mathbf{M}_0)$ is called a deadlocked marking if no transition is enabled in **M**. The transition causing deadlock is not enabled since,

$$(\forall\ p \in {}^\cdot t)(\mathbf{M}(p) = 0).$$

The Reachability set is the set of all markings reachable by the firing sequence from an initial marking $\mathbf{M}_0$ of a PN, and is denoted by $R(\mathbf{M}_0)$. The Control Model determines the set of input places at each state having paths to transitions. For a transition to be enabled, all places in this set must have a token. If no set has tokens in all places, a deadlock state occurs. A path from these sets to transitions is determined from the input matrix (C^-). When an enabled transition fires, the output matrix (C^+) provides a path for token movement to the output places. The state equation computes the next resulting state when the enabled transition is fired.



Places
- p_{01} - part available at CV_1
- p_{02} - part available at CV_2
- p_1 - part being transported by R_1 to M_1
- p_2 - part being machined by M_1
- p_3 - part being transported by R_2 using G_2 from M_1 to CV_3
- p_4 - part being transported by R_2 using G_2 to machine M_2
- p_5 - part being machined by M_2
- p_6 - part being transported by R_1 using G_1 from M_2 to CV_4
- p_7 - R_1 available
- p_8 - G_1 available
- p_9 - M_1 available
- p_{10} - M_2 available
- p_{11} - R_2 available
- p_{12} - G_2 available

Transitions
- t_1 - R_1 picks part from CV_1
- t_2 - start machining part on M_1
- t_3 - finish machining part on M_1
- t_4 - unload part to CV_3
- t_5 - R_2 picks part from CV_2
- t_6 - start machining part on M_2
- t_7 - finish machining part on M_2
- t_8 - unload part to CV_4

Figure 2: PN Model of the Control Flow for the FMC.

The Control Model uses the PN in MDF to detect deadlocks in a manufacturing plan processed by the FMC. At each state, paths for the token from P → T are determined from the C^- matrix. This determines a set of places that must enable transitions. A transition selected for firing must have tokens in the input places, and a path from T → P for transporting tokens to output places. These are determined from state $\mathbf{M}_k$, and the C^+ matrix. The algorithm continuously executes the steps until a live PN or a deadlock state is detected.

Deadlock is detected when the initial, or goal state, cannot be reached by firing any transitions. In the case of the FMC, the goal state is set at a point where the machine tools and robots are ready, the grippers are free, and a raw part is in the input conveyor.

5 CONTROL MODEL RESULTS

Raw parts are supplied randomly to the input conveyors of the cell. When the machine tools are ready, the robots load raw parts. When the machine cycles are complete, the robots unload the parts, and the cell is available for the next part.

5.1 Generation of the Control Model

The PN model in graphical form (Figure 2) is converted into a PN in MDF as:

MDF = $(C^+, C^-, \mathbf{M}_0)$, where C^+ and C^- matrices are shown in Figure 3 and $\mathbf{M}_0$ = (00000000000011).

5.2 Checking the Control Model for Deadlocks

The PN generated in MDF is analyzed for deadlocks using Deadlock Detection Algorithm (Refer to Appendix) The PN in MDF is used as an input to the Algorithm (Step 1). In Step 2, the state counter is set at zero, and the goal state is represented as:

$$\mathbf{M}_g = \mathbf{M}_0.$$

In Step 3, the current state $\mathbf{M}_0$ of the system is examined. Places p_{01} and p_{02} each having a token is listed. From the C^- matrix, paths exist from $p_{01} \rightarrow \{t_1\}$; $p_{02} \rightarrow \{t_5\}$. The selected enabled transition $(t_s) = t_1, t_5$.

In Step 4, paths from t_1 and t_5 are checked from the C^+ matrix. Paths exist from $t_1 \rightarrow P_1$ and $t_5 \rightarrow P_4$. The control vector is:

$\mathbf{u}_0$ = (10001000).

The next state is computed in Step 5 using the state equation as:

$\mathbf{M}_1 = \mathbf{M}_0 + \mathbf{u}_0.C$ = (10010011001100).

C⁺ Matrix

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p01	p02
t1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
t2	0	1	0	0	0	0	0	0	1	0	0	0	0	0
t3	0	0	1	0	0	0	0	0	0	0	1	1	0	0
t4	0	0	0	0	0	0	0	0	0	0	0	0	1	0
t5	0	0	0	1	0	0	0	0	0	0	1	1	0	0
t6	0	0	0	0	1	0	0	0	0	1	0	0	0	0
t7	0	0	0	0	0	1	1	1	0	0	0	0	0	0
t8	0	0	0	0	0	0	0	0	0	0	0	0	0	1

C⁻ Matrix

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p01	p02
t1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
t2	1	0	0	0	0	0	1	1	0	0	0	0	0	0
t3	0	1	0	0	0	0	0	0	1	0	0	0	0	0
t4	0	0	1	0	0	0	0	0	0	0	1	1	0	0
t5	0	0	0	0	0	0	0	0	0	0	0	0	0	1
t6	0	0	0	1	0	0	0	0	0	0	1	1	0	0
t7	0	0	0	0	1	0	0	0	0	1	0	0	0	0
t8	0	0	0	0	0	1	1	1	0	0	0	0	0	0

Figure 3: Output and Input Matrices for the FMC.

$\mathbf{M}_1$ is compared with goal state $\mathbf{M}_g$ in Step 6. Since the goal state is not reached, Step 7 increments the counter. The Algorithm branches back to Step 3 and the procedure is repeated. The Deadlock Detection Algorithm found the model to be live, since $\mathbf{M}_8 = \mathbf{M}_g$.

6 CONCLUSIONS AND FUTURE RESEARCH

Conflicts and deadlocks pose a serious problem in concurrent operating systems such as the FMC, and can result in complete stoppage of the cell. This problem can be avoided by off-line analysis of the PLC program using a Control Model. In this paper, the PN Controller Model developed for an FMC was analyzed for deadlocks, and was found to be deadlock-free.

The model must be validated under varying operating conditions and input buffer capacities. The throughput rate to the input buffer is to be varied for different machining rates, to study the effects on the overall performance of the FMC. At higher input rates, the machine tool's utilization is expected to drop while the robot utilization will increase, thus making the robot the controlling component in the subsystem. Robot utilization can be increased by servicing several machine tools located in a robot cell layout. Deadlock possibility is increased since the robot is simultaneously required to attend to multiple machines. A Control Model is to be designed for the robot cell layout, and determining the optimum number of machines a robot can service needs to be further investigated.

ACKNOWLEDGEMENTS

1. Research reported in this paper was supported in part by the Hampton University Faculty Research Grant.
2. The authors wish to thank Higher College of Engineering Publishing House, Zielona Gora, Poland for extending the right to use portion of the material from their paper referenced in this article.

APPENDIX

Deadlock Detection Algorithm.

Step 1: Given PN in MDF as,
MDF = $(C^+, C^-, \mathbf{M}_0)$.

Step 2: Set k = 0.
Set goal state $\mathbf{M}_g$.

Step 3: List places p_q having $m(p_q) \geq 1$ in $\mathbf{M}_k$.
Check columns p_q of C^- matrix.
Determine paths from $p_q \rightarrow t_r$.
Check rows of C^- matrix for each transition t_r.
Determine sets of input places to each t_r.
Form a set of enabled transition $\{t_{re}\}$, such that all input places to this set have tokens in $\mathbf{M}_k$.
If $\{t_{re}\} = 1$,
then set $t_s = t_{re}$, goto step 4.
If $\{t_{re}\} > 1$,
then t_s = earliest transition,
goto step 4.
If $\{t_{re}\} = 0$,
then indicate deadlock state,
goto step 8.

Step 4: Check row t_s of C^+ matrix.
If path exist from $t_s \rightarrow$ {Set of places in P},
then determine control vector $\mathbf{u}_k$, having entry 1 for transition t_s, and 0 elsewhere.

Step 5: Compute next state using state equation,
$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{u}_k.C$, k = 0,1,2, ...

Step 6: If $\mathbf{M}_{k+1} = \mathbf{M}_g$, then PN has reached final state. Indicate PN is live and deadlock-free,
goto step 8.
Else, goto Step 7.

Step 7: Set k = k + 1,
goto step 3.

Step 8: End ALGOL 2.

REFERENCES

Banaszak, Z. A., R. Wojcik, and K. A. D'Souza. 1992. Petri Net Approach to the Synthesis of Self-Reprogramming Control Software. *Applied Mathematics and Computer Science*. **2**: 65-86.

Boucher, T. O., M. A. Jafari, and G. A. Meredith. 1990. Petri Net Control of an Automated Manufacturing Cell. *Advanced Manufacturing Engineering*. **2**: 151-157.

Čapkovič. F. 1988. A Decision Support Algorithm for Flexible Manufacturing Systems Control. *Computers in Industry*. **10**: 165-170.

Chocron, D, and E. Cerny. 1980. A Petri-Net Based Industrial Sequencer. *Proceedings of the Conference on Applications of Mini and Micro Computers (Industrial Electronics and Control Instrumentation Society), Philadelphia, PA*. 18-22.

Crockett, D., A. Desrochers, F. DiCesare, and T. Ward. 1987. Implementation of a Petri Net Controller for a Machining Workstation. *Proceedings of IEEE International Conference on Robotics and Automation, Raleigh, NC*. 1861-1867.

D'Souza, K. A., and S. K. Khator. 1993a. A Survey of Petri Nets in Automated Manufacturing Systems Control. *Working Paper, School of Business, Hampton University, Hampton, VA*.

D'Souza, K. A., and S. K. Khator. 1993b. A Petri Net Approach for Modeling Controls of a Computer-Integrated Assembly cell. To appear in *International Journal of Computer Integrated Manufacturing* August Issue.

D'Souza, K. A., and S. K. Khator. 1992. Petri Net Based Simulation of Controls for a Computer-Integrated Asssembly Cell. *Proceedings of 1992 Winter Simulation Conference, Arlington, VA*. 955-962.

Havender, J. W. 1968. Avoiding Deadlock in Multitasking Systems. *IBM Systems Journal*. **7**: 74-84.

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

Murata, T., N. Komoda, K. Matsumoto, 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*. Prentice Hall Inc., Englewood Cliffs, NJ .

Teng Sheng-Hsien, and J. T. Black. 1990. Cellular Manufacturing Systems Modeling: The Petri Net Approach. *Journal of Manufacturing Systems*. **9**: 45-54.

Viswanadham, N., and Y. Narahari. 1992. *Performance Modeling of Automated Manufacturing Systems*, Prentice Hall, Englewood Cliffs, NJ.

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 modeling Automated Manufacturing Systems, and theory of Petri Nets.

ZBIGNIEW A. BANASZAK is a Professor in the Department of Mathematics at Kuwait University. He received his Ph.D. in Control Engineering from the Technical University of Wroclaw, Poland in 1977. His current research interests are in theory of Petri Nets, Concurrency Control problems, and Modeling and Control of FMS.

ROBERT WOJCIK is a researcher at the Technical University of Wroclaw, Poland. He received the M.S. degrees in Electronic Engineering and Computer Science from the Technical University of Wroclaw, Poland in 1989 and 1991, and is currently pursuing a Ph.D. degree. His research interests are focused on real-time concurrency problems, modeling discrete event systems, and Manufacturing Systems Design and Analysis.