Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds..

SIMULATION SYSTEM MODELING FOR MASS CUSTOMIZATION MANUFACTURING

Guixiu Qiao
Charles McLean
Frank Riddick

Manufacturing Simulation and Modeling Group
National Institute of Standards and Technology
Gaithersburg, MD 20899, U.S.A.

ABSTRACT

Emerging rapidly as a new paradigm of the 21st century, Mass Customization Manufacturing (MCM) systems possess some special characteristics that make the modeling of such systems extremely difficult. These characteristics include concurrency, synchronization, and cooperation among subsystems. Moreover, MCM emphasizes shortened product life-cycles, which means production lines have to be changed or reconfigured frequently. Highly flexible and re-configurable factories must be designed, simulated, and analyzed. To support the development and analysis of these systems, new approaches to modeling and simulation must be developed.

In this paper, a methodology for representing manufacturing systems using valid, colored Petri Net is presented. This method for modeling and simulating is flexible enough to support the dynamic nature of the operation of MCM systems. It is able to represent solutions to problems such as dynamic rescheduling, shop reconfiguration, part rework processing, and mechanisms for recovery from machine failure. MCM systems must often support product design modifications at late stages of production, and must respond and adjust to these changes quickly, without postponing delivery time.

1 INTRODUCTION

In today's economy, continuous competition and the dynamic global market have pushed manufacturers to transition from mass manufacturing techniques toward flexible and rapid response methods, to enable them to deliver products rapidly while keeping costs down. This can mean embarking on an approach called "Mass Customization Manufacturing" (MCM). The goal of MCM is to build customized products, even if the lot size is one, and to achieve a customization/costs balance (Pine 1993).

The splendid future of MCM has been realized by manufacturers from apparel to automobiles (Bourke and Arts 1999, Alford and Sackett 2000). The reason is simple. With their markets becoming fragmented into lower volume and customer-driven products, the product life-cycle has been significantly shortened. Traditional mass manufacturing cannot keep up with this pace. The future success of manufacturing companies will depend on rapid innovation, rapid production performance, and the ability to react quickly to changes. To remain competitive, highly flexible and re-configurable factories must be designed, simulated, and analysed. To guarantee profits they must support the efficient production of small-batch, highly varied, and highly customized products, while keeping costs to the level of mass-produced items. The earlier manufacturers can achieve this capability, the stronger their advantage in the marketplace.

This is an extreme challenge for manufacturers. If Mass Customization Manufacturing is the goal of manufacturing, can companies be agile enough to compete? In MCM, as product varieties increase and batch sizes drop, effective approaches must be developed to solve critical problems of production costs and time.

To meet such challenges, new methodologies must be developed. As Joseph Pine has pointed out, "MCM is not the continuous improvement plus"(Pine, Victor, and Boynton 1993). That is, the improvement of traditional techniques such as Lean Production (LP) will not solve the problem totally. For example, in the domestic Japanese automobile market, manufacturers found that, even for their efficient LP systems, it was hard to adapt to the frequent change of manufacturing batch quantities and the short delivery times required for MCM. Manufacturers were forced to explore ways to modify or moderate their LP approaches. There are still many problems and key technologies to be solved (Margaret 1996).

One problem is the development of simulation models that describe MCM systems. With shortened product life-cycles, production lines have to be changed or reconfigured frequently to support new product design. Today's relatively rigid production lines and their design methods have become the bottleneck of this process. The methods for modeling and simulating MCM systems must be flexible enough to support the dynamic nature of the operation of these systems. The modeling method must be able to represent solutions to problems such as dynamic rescheduling, shop reconfiguration, part rework processing, and mechanisms for recovery from machine failure. Such manufacturing systems can require design changes even at late stage of production. Responding and adjusting to these changes quickly, without postponing delivery time must be supported by the modelling methodology if it is to be useful.

If such a modelling methodology could be developed, highly flexible, rapidly re-configurable production lines could be designed in detail and analyzed using discrete-event simulation tools. Manufacturing processes could be planned and optimized. Factory layout could be simulated. Moreover, if demand changed, the simulation model could be quickly modified to perform analysis according to the new demand. Based on this analysis, manufacturing capability and production process can be adjusted, layout can be reconfigured, and resources can be reassigned. MCM emphasizes dynamically and seamlessly adjusting current production to customer demand without interrupting current production activities.

2 WHY PETRI NET?

Petri Net (PN) is a methodology that can be used to design discrete-event-system models graphically and mathematically where concurrency, synchronization, and cooperation exist among subsystems. It is also good at describing static and dynamic system characteristic, and system uncertainty. These are the characteristics necessary for modeling MCM systems. Moreover the structure of Petri Net models can be exploited to develop efficient algorithms for system control.

A simple PN is defined as a bipartite graph consisting of places, transitions, and tokens. Places, or P elements, are defined as resources, which are classified by functions. Transitions, or T elements, represent the consumption of resources, and the corresponding changes of tokens. Tokens represent factors that affect system state, including raw materials, labor, equipment, data, and information. In addition, a PN may have an associated set of enabling and firing rules to determine under which conditions (particular marking) a transition is enabled and may fire.

Some basic definitions of Petri Net are given below (Yuan 1998):

Definition 1: Petri Net (PN) is a 5-tuple, PN=(P, T, I, O, M_0), where:

- $P=\{ p_1, p_2,\dots, p_n\}$ is a finite set of places represented by circles, $n\geq 0$
- $T=\{ t_1, t_2, \dots, t_m\}$ is a finite set of transitions represented by bars or rectangles, in such a way that $P\cap T=\Phi$ and $P\cup T\neq\Phi$, $m\geq 0$
- I: $P\times T\text{->}N$ is the input function that defines directive arcs from places to transitions ($\{N=0,1,2,\dots\}$)
- O: $T\times P\text{->}N$ is the output function that defines directive arcs from transitions to places. Where ($\{N=0,1,2,\dots\}$)
- M_0 is the initial mark, $M_0(p)$ indicates the number of token at the initial state.

Definition 2: If the marking $M(p_i)= M_i$, then the number of tokens contained in place p_i is M_i .

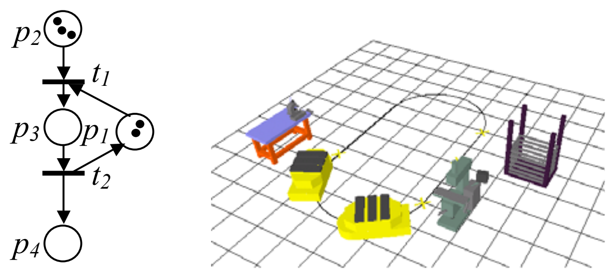
Definition 3: $I(p_i, t_j)$ indicates the directive arc connection from p_i to t_j. If $I (p_i, t_j)=K$, K is the priority value. The definition of $O (p_i, t_j)$ is similar to $I(p_i, t_j)$.

Definition 4: $PN=(P, T, I, O, M_0)$, If $p_i\in P$ and $M(p_i) \geq \#(p_i, I(t_j))$, then transition t_j is enabled. Among them, $\#(p_i, I(t_j))$ stands for the priority factor of the transition from p_i to t_j.

Definition 5: When transition t_j is enabled, it is said to be "fired," and a new mark $M'(P)$ is generated, $M'(P)= M_0(P)+O(p_j, t_j)-I(p_j, t_j)$.

In real-world systems, it is often found that even though many parts or operations are similar, they must be represented by disjoint and identical sub-nets in Petri Net. This means that the net becomes large and it becomes difficult to see the similarities between the individual subnets. Colored Petri Nets provide a more compact representation where individual sub-nets are replaced by one subnet with different kinds of tokens, each token having a color and representing a different sub-net in the equivalent Petri Net. (Kasturia, Dicesare, and Desrochers 1988, Elkoutbi and Keller 1998).

An example of Petri Net is presented below in Figure 1. The system is composed of computer-controlled machining centers (CNC), auto-guided vehicles (AGV), and buffers. The AGV takes a case from storage, delivers it to the CNC machine, takes the empty case back to storage, then



p_1: AGV idle, waiting for task assignment

p_2: Part cases in warehouse waiting to be transport

p_3: AGV taking out parts and transporting them to CNC

p_4: machining process in CNC

t_1: indicating the state before process p_3

t_2: indicating the state after process p_3 and before process p_4.

Figure 1: Example of Petri Net

returns to the starting place and waits for new commands. The Petri Net model is described as:

PN=(P, T, I, O, M_0) , where:
$P=\{p_1, p_2, p_3, p_4\}$
$T=\{t_1, t_2\}$
$\Sigma I (p_i, t_i) = \{ I(p_1, t_1), I(p_2, t_1), I(p_3, t_2)\}$
$\Sigma O(p_i, t_i) = \{ O(p_3, t_1), O(p_3, t_2), O(p_4, t_2)\}$
$M_0=(2, 3, 0, 0)^T$.

M_0 indicates the initial marking. It means that at the initial state, there are two AGVs and three part cases available in the system. This initial marking state satisfies the fire rule for transition T_1. After transition T_1 is fired, place P_1 is enabled. Then the marking state changes to:

$$M_1=(1, 2, 1, 0)^T$$

One issue with using Petri Net to model a manufacturing system is that the difficulty of building and analyzing a PN increases greatly with the complexity of the system being modeled. (Lee, Favrel, and Baptiste 1987). If a system model is very complex, containing thousands of nodes and transitions, the analysis of this model will be very difficult and time consuming. An approach must be developed where correct PN models of a complex system can be developed and extended from simpler models that are easy to prove valid.

3 VALID PETRI NET MODELS

It is critical to guarantee that Petri Net model being developed is valid. The validity of Petri Net model is defined by the three properties of being bounded, live, and reversible.

To be bounded indicates the absence of overflow in the system model. This characteristic allows the specification of a limit on the number of tokens that may be in a place at any time.

To be live implies that there is no possibility of deadlock. This characteristic is significantly important in manufacturing systems with concurrent processes and shared resources, where deadlock conditions can easily occur (Viswanadham, Narahari, and Johnson 1990).

To be reversible indicates that the system can return to its initial state from any current state. This characteristic is very important for error recovery.

Once a system model can be verified as being valid, it is assured of being reliable, without overflow, deadlock or conflicts.

The analysis of validity is difficult and time-consuming especially for large models with thousands of nodes and transitions. To avoid the expensive traditional verification approaches, a methodology for validly modeling MCM systems using color Petri Net is presented.

A manufacturing system model can be considered as a list of resources (places), a list of operations (transitions), and the precedence relationships (paths). We can construct some basic Petri Net models of systems whose validity is easy to prove. To establish the simulation model of a manufacturing system, decompose the system into cells according to manufacturing function and the type of processing the cell will do. Represent each of these cells with a valid PN model. These models will be the sub-nets of the complete system model. Utilizing the theorem of PN valid extension (see below), connect the sub-nets by using standard linkages or valid buffer models. Finally, the simulation model of the manufacturing system is established and guaranteed to be valid. In the following section, a discussion of the theorem of Petri Net valid extension is presented.

3.1 Theorem of Petri Net Valid Extension

In figure 2, assume *that* $Z=\{P, T ,I, O, M_0\}$ and $S=\{P_S, T_S, I_S, O_S, M'_0\}$ are two subsets of $Z'=\{P', T', I', O', M'\}$, where $P'=P\cup P_S$, $T'=T\cup T_S\cup\{T_S\}\cup\{Te\}$ and $M'=\{M_0, M'_0\}$, Then

- Z' is bounded if and only if Z and S are bounded
- Z' is live if and only if Z and S are live
- Z' is reversible if and only if Z and S are reversible.

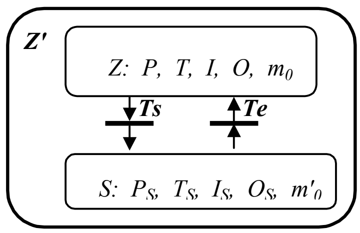


Figure 2: Valid Extension of Petri Net

Places connected by *T*s and *T*e could be a same place in *Z* or different places of a serial event. When *Ts* and *Te* are reversed, it is called backward firing, and the theorem remains true.

This theorem states that if Petri Net *Z'* is composed of *Z* and *S*, then *Z'* is valid if and only if *Z* and *S* are valid.

A proof of this theorem can be found in reference (Zhou 1989).

3.2 Establishment of Basic Valid Petri Net Model

Below some basic valid Petri Net models are defined. These basic models possess the properties of being bounded, live, and reversible. A complex manufacturing system can be achieved by extending these basic models according to Petri Net valid extension theorem.

A simple serial system is depicted in Figure 3a. Once the initial fire condition m_0=(1, 0)^T, is satisfied, the sys-

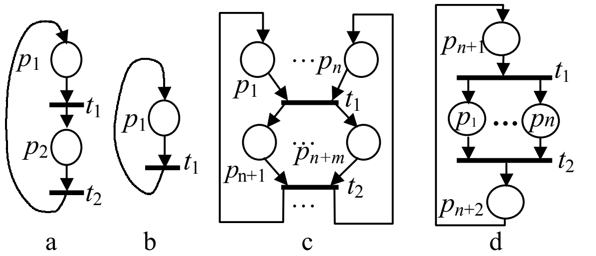


Figure 3: Basic Valid Petri Net Models

tem can run recurrently. Figure 3b is the generalized model of serial system.

For parallel system, as depicted in Figure 3c, the initial fire condition is m_0=(1, 1,…,1, 0, 0,…,0)^T, where place p_1 through p_n initially contain tokens and places p_{n+1} through p_{n+m} do not. Figure 3d is a valid Petri Net model for series-parallel system.

The validity of these models are easy to prove (Yuan 1998).

3.3 Handling of Shared Resources

The models developed above do not solve the problem of how to validly add shared resources to Petri Net model. Shared resources are the main reason for deadlock.

How can shared resources be safely added while keeping the whole system model valid? Many people have contributed great effort to this problem. Fortunately, Zhou presented a theoretical basis for Petri Net synthesis methods that provides for modeling systems with shared resources. Parallel mutual exclusion (PME) and sequential mutual exclusion (SME) resource-sharing concepts can be formulated in the context of Petri Net theory (Zhou and DiCesare 1991).

If we add shared resources according to this methodology, the model developed with such structures remains bounded, live, and reversible. The definition and proof of PME and SME can be seen in reference (Zhou and DiCesare 1991).

3.4 Valid Buffer Petri Net Models

Figure 4 shows some valid Petri Net models of buffers. A manufacturing system can be separated into individual workcells that can be described as basic valid Petri Net models. Different workcells are connected through various kinds of buffers. According to the valid Petri Net extension theorem, valid buffer models have to be established.

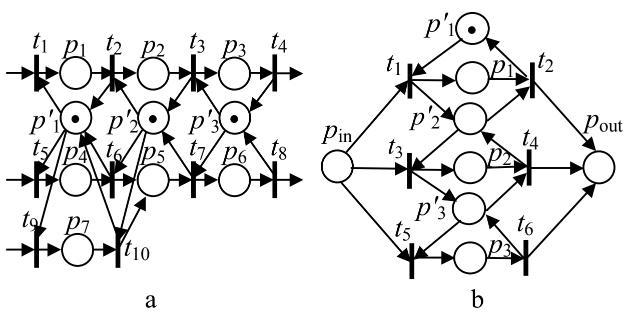


Figure 4: Valid FIFO (a) and LIFO (b) buffer Petri Net

In figure 4a, a valid PN for a FIFO buffer is shown. It can stand for any buffer characterized by First-In-First-Out ordering with indistinguishable parts. For example, this type of buffer is useful as the buffer between two CNCs. It stores parts dispatched from the first CNC that wait for the processing by the next CNC.

The PN in figure 4b presents a buffer with last-in-first-out (LIFO) ordering and indistinguishable parts.

4 HANDLING OF UNPREDICTABLE CASES

The unpredictable cases discussed here include robot breakdowns, labor unavailability, and recoverable system errors.

To understand the robot breakdown case, suppose a robot named Ri suddenly stops working on an assembly line. A recovery procedure can be started that makes other surrounding robots available to perform Ri's work. The running pace of the line can be slowed down and workspaces recalculated to decide which robots should be used to substitute for Ri. If two or more robots are used, calculations to detect interference can be performed and kinematics parameters can be modified to avoid collision. The system model can support dynamic operation insertion and adjustment.

For recoverable system errors, such as a tool breaking, an error handling procedure can be started (e.g., replacing the broken tool and re-executing the same operation again). Petri Net that describe four types of error handling procedures are depicted in Figure 5: a) Restart process after error handling procedure; b) Alternate path method; c) Backward error recovery method; d) Forward error recovery method (Zhou 1989, Feicht, DiCesare, and Goldbogen 1987).

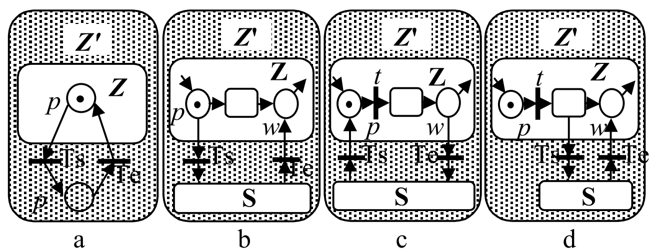


Figure 5: Four Types of Error Recovery Procedures

From the theorem of Petri Net valid extension (figure 2), an error handling procedure can be treated as a sub-net S, T_s as the firing transition and T_e as the ending transition. Then if S is valid, the whole system's Petri Net model denoted by Z' is still valid.

5 EXAMPLE OF A MULTI-ROBOT FLEXIBLE ASSEMBLY LINE AND ITS ASSOCIATED VALID PN MODEL

A multi-robot flexible assembly line and its Petri Net model are presented in figures 6, 7 and 8. It is composed of 3 CNC machines (M1, M2, M3), 2 MOTOMAN type robots(R1, R2), one PUMA type robot(R3), two SCAR type robots(R4, R5) and two buffers(B1, B2), 3 conveyors and an assembly station (AS). Parts to be processed are Part A, Part B and Part C. The system is divided into 3 cells.

- Processing cell A: R1, M1, R2 and M2 process Part A. When parts are finished processing, they are stored in Buffer 1
- Processing cell B: R3 and M3 process Part B. When Parts are finished the machining process, they will be stored in Buffer 2
- Assembly cell C: Robot 4 and Robot 5 work co-operatively. They pick up parts from Buffer A and Buffer B, and assemble them together with a new part C. If one of these robots is broken, an emergency step is adopted to use the other robot to perform the broken one's work.

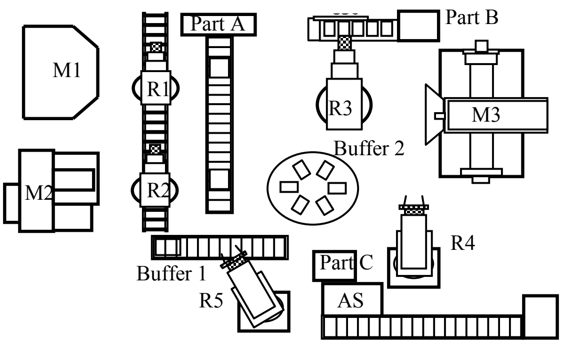


Figure 6: System Layout

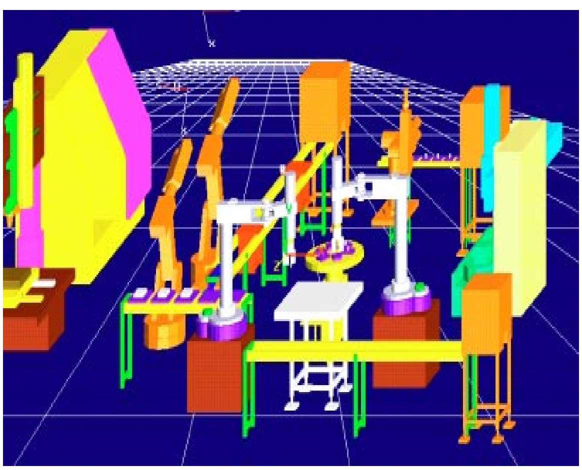


Figure 7: Multi-Robot Flexible Assembly

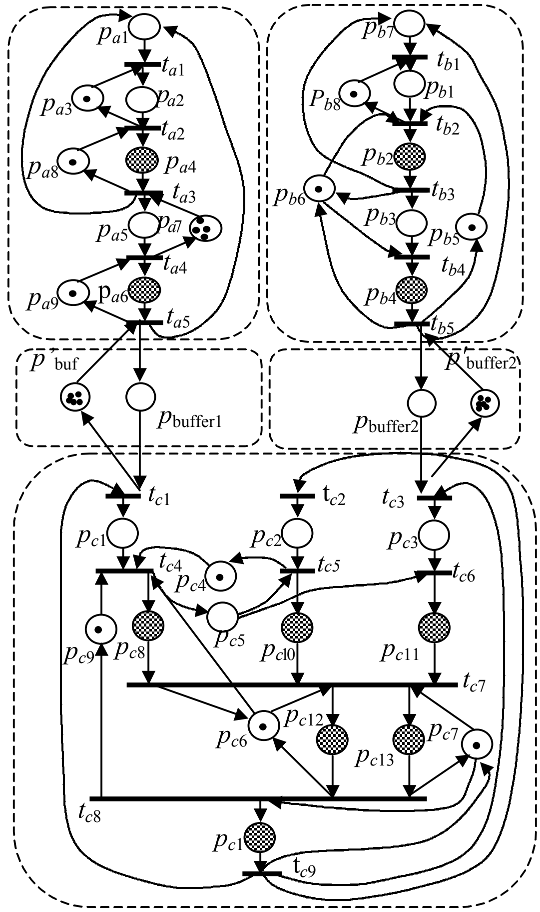


Figure 8: System Valid Petri Net Model

Basically this system is a parallel system. First each individual cell's valid model is developed. Each of them basically is a serial type model. According to the valid modeling methodology, they can be combined into the whole system model through valid buffer models. The Petri Net model of the whole system is depicted in figure 8.

The model developed can be directly applied to system controller programming, realizing the integration of factory planning and process control. A simulation was developed with C and C++ on SGI workstation based on GL. It is a hierarchical platform composed of a modeling module (including geometry modeling, kinematics modeling and calculation, collision detection), a database (including a mechanism and device database), a robotic task design language and platform upper level control module. Detail information about this platform can be found in reference (Qiao 2001).

6 CONCLUSION

In this paper, based on the Petri Net theory, an efficient valid Petri Net manufacturing system modeling methodology is developed. The model developed is flexible enough to handle problems of dynamically inserted schedules, system changes, multi-robot co-operation control and the handling of unpredictable cases. It offers not only a means to model discrete-event systems graphically and mathematically where concurrency, synchronization and co-operation exist among subsystems, but also can be easily converted into computer control code and interfaced to practical manufacturing processes.

ACKNOWLEDGMENTS

This document was prepared based on the prior work of many other researchers. In particular, Wang Shuguo deserves special mention for his help and assistance.

REFERENCES

Alford D., Sackett P., Nelder G. 2000. Mass customization-an automotive perspective. *Int. J. Production Economics* 65:99-110.

Bourke R., Arts J., Roest M. V. 1999. Achieving success with mass customization: the vital contributions of engineering. *Computer-Aided Engineering* (10):42-52.

Elkoutbi Mohammed, Keller Rudolf K.. 1998. Modeling interactive systems with hierarchical colored petri nets. Available online via http://citeseer.nj.nec.com/elkoutbi98modeling.html [accessed Feb. 16, 2002].

Feicht D., DiCesare F., Goldbogen G. 1987. Intelligent error recovery for manufacturing workstations. *Proc. Tenth IASTED Symp. of Robotics and Automation*, Lugano, Switzerland, 271-279.

Kasturia E., Dicesare F., Desrochers A. A. 1988. Real-time control of multi-level of manufacturing systems using colored petri nets. *IEEE Int. Conf. Robotics and Automation,* Raleigh, NC,: 1114-1119.

Lee K. H., Favrel J., Baptiste P.. 1987. Generalized petri net reduction method. *IEEE Trans. Syst. Man Cybern* 17(2): 297-303.

Margaret A E. 1996. Implementing mass customization. *Computers in Industry* (3) :171-174.

Pine B.J. II. 1993. Mass customization-the new frontier in business competition. McGraw-Hill.

Pine B.J. II. Victor B., Boynton A. C. 1993. Making mass customization work. *Harvard Business Review* (9):108-119.

Qiao Guixiu et al. 2001. CAID platform for mass customization manufacturing. *Journal of Software* 12 (Suppl.): 393-398.

Viswanadham N., Narahari Y., Johnson L. 1990. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petri net models. *IEEE Trans. on Robotics and Automation* 16(6): 713-722.

Yuan Chongyi. 1998. Petri nets theory. Machinery Press.

Zhou M C. 1989. Adaptive design of petri net controllers for error recovery in automated manufacturing systems. *IEEE Trans. On Systems, Man and Cybernetics* 19: 963-973.

Zhou M C, DiCesare F. 1991. Parallel and sequential mutual exclusions for petri net modeling of manufacturing systems with shared resources. *IEEE Trans. on Robotics and Automation* 7 (4): 515-527.

AUTHOR BIOGRAPHIES

GUIXIU QIAO is a researcher in the Manufacturing Simulation and Modeling Group in the U.S. National Institute of Standard and Technology (NIST) Manufacturing System Integration Division. Before NIST, She worked as a postdoctoral researcher in Automatic Department of Tsinghua University, Beijing China. She received her Ph.D. from Robot Research Institute of Harbin Institute of Technology in 1999. Her research interests are mainly on virtual manufacturing, virtual reality, computer graphics, simulation integration, system control and optimization. Her email addresses are `<qiao@cme.nist.gov>` and `<qiaogx@hotmail.com>`

CHARLES MCLEAN is Leader of the Manufacturing Simulation and Modeling Group in the U.S. National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has managed research programs in manufacturing simulation, engineering tool integration, product data standards, and manufacturing automation at NIST since 1982. He has authored more than 50 papers on topics in these areas. He holds a Master's Degree in Information Engineering from University of Illinois at Chicago and Bachelor's Degree from Cornell University. His e-mail and web addresses are `<mclean@cme. nist.gov>` and `<www.mel.nist. gov/msid>`.

FRANK RIDDICK is a staff member in Manufacturing Simulation and Modeling Group in the U.S. National Institute of Standards and Technology (NIST) Manufacturing Systems Integration Division. He has participated in research and authored several papers relating to manufacturing simulation integration and product data modeling. He holds a Master's Degree in Mathematics from Purdue University. His e-mail address is `<riddick@cme.nist. gov>`