01 Jan 1991

Petri Net Modeling of a Flexible Assembly Station for Printed Circuit Boards

M. Zhou

Ming-Chuan Leu
Missouri University of Science and Technology, mleu@mst.edu

Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

PETRI NET MODELING OF A FLEXIBLE ASSEMBLY STATION FOR PRINTED CIRCUIT BOARDS

MengChu Zhou

Electrical and Computer Engineering Department
Center for Manufacturing Systems
New Jersey Institute of Technology
Newark, NJ 07102, USA

Ming C. Leu

Mechanical and Industrial Engineering Department
Center for Manufacturing Systems
New Jersey Institute of Technology
Newark, NJ 07102, USA

ABSTRACT

This paper presents Petri net modeling approaches for a flexible workstation for automatic assembly of printed circuit boards. In order to improve the productivity of such a system, the building of mathematical models is a crucial step. Concentrating on the operational aspects, we construct ordinary and temporal Petri net models for an existing physical flexible workstation (AT&T FWS-200). Three outcomes follow from such models. First, designers can have better understanding of concurrency, synchronization, mutual exclusion, and sequential relations involved in the system control from the graphical representations of Petri nets. Second, the performance analysis of system operations under different settings can be conducted. Thus, the results can be used to help designers choose the best operational setting based on the system parameters. Third, the models can be used as an aid for automatic generation of real-time control programs and construction of Petri net based simulation if needed. The approaches suggested in this paper can be generalized to many other applications of multi-robot assembly systems.

Keywords: *Petri nets, flexible assembly systems, printed circuit boards (PCB), system performance*

I. INTRODUCTION

There is growing importance in automating electronic assembly lines in order to compete effectively in today's market. The goal of this research is to offer a solution to achieve a higher speed and more flexibility in electronic assembly and thus to increase manufacturing productivity by searching for the optimal setting of operational environment, locations of feeders, sequence of component insertions, and planning of assembly task. Petri nets can be used to study the optimal operational setting problems by modeling concurrent assembly systems and evaluating the production rates of various possible settings. Mathematical programming approaches may be used to solve other problems [2, 11, 18].

A critical step toward such a research goal is to use Petri nets as a modeling tool to describe the concurrency, synchronization, and mutual exclusion of assembly systems. We have used the petri net modeling method to model an assembly system for printed circuit boards (PCB). The method has been demonstrated through the modeling of an AT&T FWS-200 flexible assembly workstation.

Generally speaking, Petri net modeling includes two parts: ordinary Petri nets for system behavior analysis and temporal Petri nets for system performance evaluation. The ordinary Petri nets (or non-timed Petri nets) [13] are used to analyze such system properties as deadlock-freeness, buffer boundedness, reversibility (or re-initialization property), and conflict-freeness. After ordinary Petri nets are built, time variables can be used to associate with their places and transitions, thus resulting in temporal Petri net models. These models serve to derive system performance indices including, for example, productivity and machine utilization. When there are several operational settings possible, different Petri net models can be obtained and used to study the optimal setting by comparing the results of Petri net based performance analysis.

Based on the designed Petri net models, Petri net controllers (in particular cases the programmable logic controllers) can be constructed for real-time applications [5, 17, 23]. These models can help designers figure out the correct control code when a controller is implemented in a specific environment. They may be used as an aid to design controller software written in general purpose languages such as C and C^{++} or manufacturing systems oriented languages such as Modular Manufacturing Language, i.e., M^2L [1].

Also, Petri net models can help formulate Petri net based simulators to simulate complicated systems for which analytical solutions are impossible. It is easier to write simulation code with the help of Petri net models. Using existing Petri net tools, such as SPNP [7] and Great SPN [4], designers can derive the effect of increasing the speed of operation or decreasing failure time upon system performance. Thus key operational parameters can be identified. Often the results can be used to guide the directions along which the system performance may be improved greatly with small effort.

These possible advantages resulting from Petri net models of a system motivated the research reported in this article. The paper is organized as follows. Section 2 discusses modeling issues for assembly systems and reviews briefly the previous work which used Petri nets to study assembly systems. The features of PCB assembly systems are discussed. Section 3 describes a printed circuit board assembly system, based on the AT&T FWS-200 flexible work station which consists of two robot arms. The fourth section discusses the ordinary and temporal Petri net models of this system. Finally, future research directions are presented and conclusions are drawn.

II. MODELING ISSUES IN ASSEMBLY SYSTEMS

2.1 Robotic Assembly Systems for Printed Circuit Boards

A modern robotic assembly system for printed circuit boards may include the following components:

1) Robots to perform assembly operations
2) Tools provided to robots
3) Feeders to supply components
4) Conveyors to supply boards.

Such a robotic system aiming at assembling printed circuit boards is considered in our study. Circuit boards with different specifications often have different layouts. Therefore, the optimal sequence of insertions/placements varies from one board to another. Also, some insertions may be more restricted than others. Different insertions may require different robots with perhaps different tools. In a factory, there may be many assembly tasks to be completed. To achieve the global optimum, all these tasks should be best planned and their operations coordinated. The complexity involved makes the formulation of an optimization model for the entire system extremely difficult.

After certain layouts of the system such as sequence of assembly tasks, sequence of insertions, and locations of feeders have been determined, the problem of setting robot operations, including pick-up, transport, insertion etc., and decision making in case of robot failure arises when two or more robots exist in the same system. A certain order of operations needs to be followed by each robot in the system. For example, the sequence {*pick-up-a-component*, *move-to-PCB*, *insert-a-component*, *move-to-a-feeder*} is true for each robot in performing component insertion on a PCB. Therefore, the first important issue is the modeling of sequential operations for each robot in the system.

The second modeling issue is synchronization. For example, a robot may pick up a component only when it is present. It will never finish the *pick-up* operation if the component is missing. Similarly, the PCB to be assembled has to be present before any insertions on the PCB can be fulfilled. When other resources are necessary to complete an operation, they must also be ready.

The third issue is modeling of concurrency. By concurrency we mean that there are no sequential relationships among the concerned events [22]. For example, two physical events $pick\text{-}up\_a\_part\_by\_Robot_1$ and $pick\text{-}up\_a\_part\_by\_Robot_2$ are concurrent if either event may occur before the other. Two robots can operate concurrently if both can process tasks at the same time. High concurrency among system resources often implies high productivity.

The other modeling issues include conflicts when the sharing of resources or spaces is encountered. Mutual exclusion concepts proposed in [21, 22] are especially useful to model the characteristics of resources shared by either independent processes or sequentially related ones. Before a practical system is implemented, one may have to assure that there exists no deadlock situation. The Petri net models can be used to analyze such a case. A more rigorous approach for Petri net modeling [20, 90, 26] is to synthesize the Petri net of a system so that it has desirable properties such as boundedness and deadlock freeness.

2.2 Literature Review

Using Petri nets for repetitive operations has been investigated by several researchers. Beck and Krogh [3] modeled a two robot assembly system by using modified Petri nets and then simulated the system based on the Petri net model. Freedman and Malowany [8] used timed Petri nets to model assembly activities where a strict order of operations is required for a single robot. The resulting net model is a marked graph and thus simple formula can be used to evaluate cycle time, for example. The time constraints were considered to be deterministic.

To evaluate the performance of job-shop systems under a deterministic and repetitive function of production process, Hillion and Proth [10] applied a special class of timed Petri nets, called timed event-graphs, for a flexible manufacturing system having three machines and three job types. The number of in-process jobs can be nearly minimized using integer programming, while the system still works at its maximal productivity.

Recently, Gasmi [9] reported the build-up of a flexible multi-robot assembly system. Petri nets were used to model assembly tasks with failures of resources considered. The conceptual system under investigation consisted of planner, scheduler, execution monitor, and failure analyzer.

Some effort has been made to use Petri nets for modeling and analysis of assembly stations for printed circuit boards. For example, Sciomachen et al. [15] successfully modeled a pick-and-place machine with high concurrency using Petri nets, and they constructed a Petri net based emulator for study of system performance. In their executable net, most of timed transitions have fixed times, although actual transitions have variable times which are dependent on parameters such as distance.

Multi-robot systems for printed circuit board assembly have three features which are distinct from conventional systems for automated assembly of mechanical parts.

1) Numerous insertions are required for a moderately complex printed circuit board. The component count ranges from 100 to 200 per board [14]. The activities for each board assembly are highly repetitive.
2) There is often no strict sequence which has to be followed. This means that the insertion order of components can be random. While in many mechanical assembly tasks, different assembly possibilities are limited due to heavy precedence constraints.
3) Each insertion is completed by a robot manipulator

even in a multi-robot system. While in mechanical assembly, two or more robots may be involved in fulfilling a single operation [3].

The increasing importance of using a flexible robotic assembly system for high-volume/low-cost production of printed circuit boards motivated our research. In the following we will discuss Petri net approaches for these systems through the development of an assembly model for the AT&T FWS-200 flexible workstation.

III. SYSTEM DESCRIPTION

The front view of the Flexible Work Station (FWS-220) developed by AT&T is shown in Fig. 3.1. It consists of the following major components:

1. The *frame* structure supports control cabinet and platen for PCB.
2. The *Control Cabinet* houses hardware including an AT&T PC computer, Industrial I/O modules & STD bus, control electronics, display, pneumatic & ventilation systems, and power supplies.
3. The *Robot Arms* are able to accomplish pick, place, and X, Y, Z, and θ motions with high accuracy (± 0.001 in. along X, Y, and Z axes and ±0.15° about Z axis).

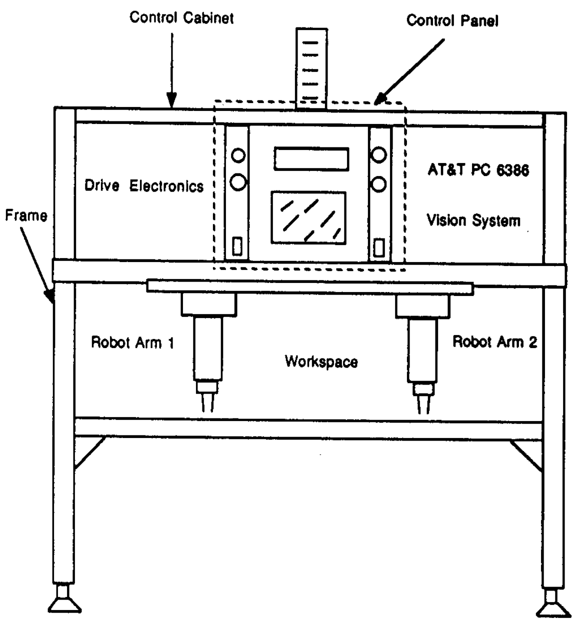

Fig. 3.1. The front view of FWS-200

4. The *AT&T PC 6386* computer serves as the supervisory controller of the system.
5. The *Drive Electronics Drawers* contain the electronics for controlling the various axes; each drawer has two servo amplifiers for Z and θ motions and a stepper amplifier for both X and Y motions.
6. The *Control Panel* contains system power buttons, touch display, pressure and vacuum gauges, and emergency stop buttons.
7. The *Vision System* integrates IRI SV512 and ICOS M10000.
8. The *Peripheral Equipment* includes parts feeders for components to be inserted.

The system is operated by programs written in M^2L [1]. These programs are application-oriented and they can be used to accomplish tasks such as surface mount of light-weight parts, assembly of through-hole components, wire harnessing, etc. The common work area is of size 24" × 32" (60.96 cm × 81.28 cm); refer to [1, 6] for details.

IV. PETRI NET MODELING

Ordinary Petri net models are first constructed which aim at the analysis of system behavior. Then temporal Petri net models are derived by careful consideration of time variables which characterize the dynamic behavior of the system. Also, for certain assembly of boards, insertion failures are explicitly represented in the Petri net models. The optimal setting of productivity varies with different system parameters such as pick-up time and distance of movement.

We assume that the reader is familiar with Petri nets in the following discussion. Refer to [12-13, 19-20, 22] fo basic knowledge of Petri nets and related properties.

4.1 Ordinary Petri Net Models

In a dedicated production line for assembly of PCB, a robot does repeatedly the following jobs: picking-moving-inserting-moving. Like such a robot, each arm of FWS-200 can do these jobs. However, since the two arms work on the same board and obtain components from the same feeder area, avoidance of arm collision has to be considered. Collision avoidance is achieved in the system by using mutual exclusion structures implemented by semaphores [1]. Thus, when one robot is doing a job above the PCB area, the other is prohibited to move into the same area. In the system of our study, possible collision may also occur in the feeder area. Based on similar reasoning, the use of mutual exclusion structures is also needed for this area.

With focus on the activities of the two robot arms in the system, we design a Petri net model as shown in Fig. 4.1. The meanings of all places are given in the net, where R_1 and R_2 represent Robot 1 and Robot 2, respectively. A transition means the start or completion of an activity. In this model, only normal operations and no insertion errors are considered. Also, we do not consider possible change of tools for robots. These considerations will be included later.

In this model, information like location data is hidden. For example, a robot arm may move from point (3.00, 2.45) to (0.00, 4.55) one time while it may move from (6.90, 5.55) to (0.00, 3.95) another time, etc. The activity "moving" is used to represent all of these motions in this model.

Two 2-parallel mutual exclusions $(p_3, \{t_{11}, t_{12}\} \cup \{t_{21}, t_{22}\})$ and $(p_4, \{t_{12}, t_{15}\} \cup \{t_{22}, t_{25}\})$ are used in this model, which can be easily verified. Using the theory developed by Zhou et al. [19-20, 25-26], we can conclude that the net is live, safe, and reversible.

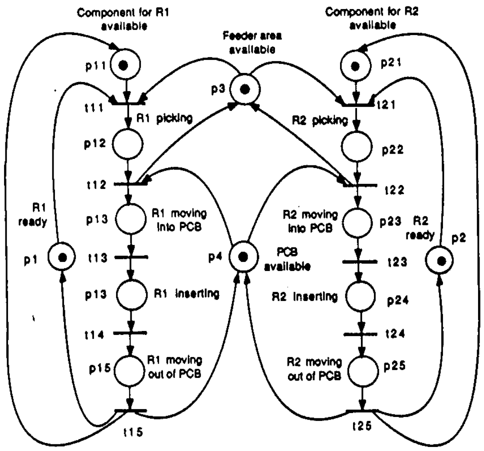

Fig. 4.1. Model with two parallel mutual exclusions

An alternative model is shown in Fig. 4.2 which specifies the exact order the two robot arms have to follow. Two control pairs, namely (p_3, p_4) and (p_5, p_6), play a key role in such a net model [22-23]. Depending on the initial marking, either robot can start the first insertion for a given printed circuit board. Suppose p_3 and p_5 are both initially marked with one token, and m_0=(1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0)^T where the values in the vector are associated with the sequence of places p_1, p_2, p_3, ..., p_6, p_{11}, ..., p_{15}, p_{21}, ..., p_{25}. The value is 1 if the place is initially marked with a token. Otherwise it is zero. Then transition t_{11} will fire before t_{21} and transition t_{12} will fire before t_{22}. After t_{11} fires once, it cannot be enabled until t_{22} fires. This is true for transitions t_{21} and t_{12}.

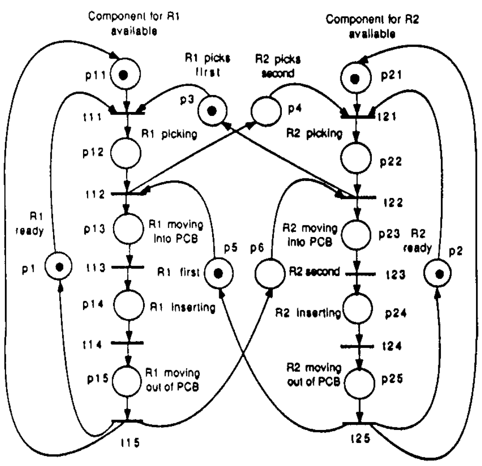

Fig. 4.2. Marked graph model

It is interesting to note that if p_3 and p_6 are initially marked with one token, neither p_4 nor p_5 is marked, and m_0'=(1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0)^T, then the Petri net models a system deadlock situation. Specifically, after t_{11} fires, no transition can be enabled in the net, thus a deadlock results. This net is a marked graph since each place in this net has exactly one input and one output transitions. Using the theory of marked graph [12], we conclude that m_0 is live and safe since each circuit contains at least one token, while m_0' is not live since the circuit t_{12}-p_4-t_{21}-p_{22}-t_{22}-p_{23}-t_{23}-p_{24}-t_{24}-p_{25}-t_{25}-p_5-t_{12} contains no token.

Although conflict is resolved in the Petri net model, the system efficiency may be degraded when such a model is coded for the system control in the real-time environment since when one robot fails, the other robot needs to wait until it is repaired [27].

When tool changes, robot errors, and insertion failures are considered, we can get the more complicated Petri net model as shown in Fig. 4.3. The newly added places are:

p_{i6}: Robot i moves to the reject container, for i =1 and 2
p_{i7}: Robot i discards the component into the reject container, for i =1 and 2
p_{28}: Robot 2 moves to the tool kit
p_{29}: Robot 2 changes its tool

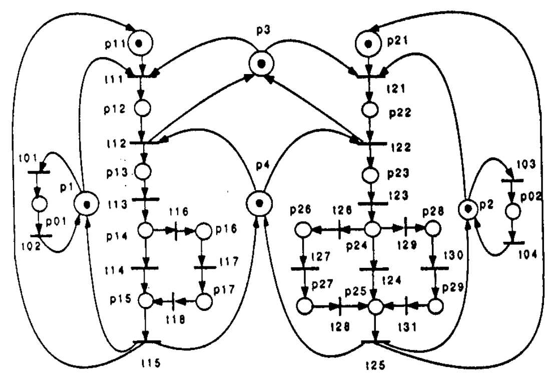

Fig. 4.3. Petri net model including tool change and errors

Here we assume that robot errors can occur only when it has completed a job, although practically it may break down at any state. Also, when there is an insertion error, the component will be simply discarded and represented in the model by three transitions and two places, i.e., t_{16}-p_{16}-t_{17}-p_{17}-t_{18} for Robot 1 and t_{26}-p_{26}-t_{27}-p_{27}-t_{28} for Robot 2. The theory about the adaptive design of Petri net controllers for error recovery [19] guarantees that the above additions preserve such properties as safeness, liveness, and reversibility. Note that these branches are equivalent to alternate path error recovery blocks [19]. Therefore, this net is also live, safe, and reversible.

4.2 Temporal Petri Net Models

All operations involved in the above Petri net models take time to complete in a real-time environment. To obtain an accurate timed net model, time data is needed for the operations. Operations like moving to PCB may be influenced by the position of the feeder and the destination of insertion on the PCB. However, since the speed of the robot arms in FWS-200 is very fast (which can be as fast as 60 in./sec, i.e., 152 cm/sec), the variation can be represented as a kind of noise. Thus, we can model the time for a place as a fixed time plus ±0.05 second.

The pick-up and insertion operations can be modeled as stochastic variables with exponential distributions. The stochastic nature is due to two reasons. One is that each operation involves different components which may have distinct geometries and weights. The second reason is that the operation cannot always be guaranteed to be successful in the robot's first try. On the average, these operations take more time than a robot movement between a feeder and the PCB. All transitions are modeled as instantaneous. When a token enters an activity place (for example, p_{12-17} and p_{22-29} in Fig. 4.3, it shall stay there for a certain time before it becomes available for initiating the next operation. The advantages using such a scheme include the ease of converting a Petri net model to a Petri net based supervisory controller and the ease of building a hierarchical Petri net model [5, 16, 21].

4.3. Petri Nets with Strict Insertion Sequences

Occasionally, it may be necessary to insert some components before others. For example, an optimal insertion sequence requires such a strict sequence. The Petri net model becomes more complicated when a strict order of component insertions is considered. Three different insertion schemes are discussed in the following, where robot errors are assumed to exist in each case.

Scheme 1: When Robot 1 (R_1) does the insertion of component P_k, Robot 2 (R_2) waits until R_1 signals. If the insertion of R_1 succeeds, R_2 picks up component P_{k+1}; otherwise, it picks up another component P_k. Then it moves to the PCB area and does insertion. Robot R_1 acts in the same way when R_2 does insertion.
Intuitively, this scheme can be effective when the pick-up operation is faster than the insertion and the robots are not very reliable.

Scheme 2: When R_1 does the insertion of component P_k, R_2 picks up component P_{k+1}. When R_1 succeeds, R_2 proceeds with the insertion of component P_{k+1}. When R_1 fails, R_2 waits until R_1 succeeds. R_1 acts in the same way when R_2 does insertion.
This scheme may be used when the robot insertion success rate is high and the robots are very reliable.

Scheme 3: When R_1 does the insertion of component P_k, R_2 picks up component P_{k+1}. When R_1 succeeds, R_2 proceeds with the insertion of component P_{k+1}. When R_1 fails, R_2 puts P_{k+1} back; R_2 then picks up another component P_k and inserts it. R_1 acts in the same way when R_2 does insertion.
This case is preferred when the pick-up operation is faster than the insertion and robot errors are frequent.

The different schemes will result in different Petri net models [27]. Then these various schemes as above described can be compared using Petri net based performance evaluation methods [4, 7, 16, 24]. With different Petri net models for the different schemes, a designer can study which scheme is the best under the consideration of variations in system parameters such as robot failure rate, time delay, etc. Also, these models can be converted into real-time control codes by including detailed assembly data and equipment operation information.

V. FUTURE RESEARCH DIRECTIONS

As discussed above, Petri net models can be designed to study various aspects of robotic systems for PCB assembly including both qualitative and temporal properties. In the future, we will aim at the following research directions:

1) Derivation of Temporal Properties

Based on the above temporal Petri net models, performance indices such as production rate and robot utilization rate can be derived either using existing software based on solutions to Markov processes or using Petri net based simulators. The best setting can be obtained by comparing the performance values of different models. Also, by studying the sensitivity of system throughput rate to different parameters, we can identify which aspects of the system should be improved for significant increase in the system throughput.

2) Generation of M^2L control codes

A practical problem of study is the implementation of a control method by coding it in M^2L and other manufacturing languages for real-time system operations. The generation of control codes can be achieved efficiently by using the derived Petri net models with the inclusion of necessary data and information in the net. Moreover, a "compiler" can be developed to directly produce a real-time control program, provided that the net model and necessary data about a particular PCB are given [22].

3) Integration of CAD/CAM Software to Assembly Systems

Information about printed circuit boards and operational parameters are usually stored in a CAD/CAM software system. The information about assembly tasks and insertion sequences can be saved in the same system. The information about the real-time status of the robotic assembly system is stored in the computer which functions as a supervisory controller of the assembly system. How to integrate the information and data in these various components effectively is another important subject of investigation, which is a key to increasing production for advanced flexible PCB assembly systems.

VI. CONCLUSIONS

This paper presents Petri net modeling approaches for design of flexible robotic systems for PCB assembly. Ordinary and temporal Petri net models are constructed for an existing physical flexible workstation. As a first step towards the design of a PCB assembly system with high speed and flexibility, Petri net modeling helps the designer understand graphically the system characteristics, such as sequential relations among system activities, concurrency, mutual exclusion, and conflicts. The resulting temporal models can be used to study the system performance by determining system productivity and utilization of resources. Furthermore, Petri net models can serve as an aid to the generation of real-time control codes with certain properties such as collision freeness and deadlock freeness. It may also be used as a tool to integrate various system hardware and software components. Furthermore, the Petri net modeling method discussed can be used to provide mathematical and graphical tools for the modeling of other complicated flexible multi-robot assembly systems. The resulting Petri net models can be used for system behavior analysis, performance evaluation, simulation, and real-time control.

VII. ACKNOWLEDGEMENTS

This research is financially supported by AT&T's Manufacturing Affiliates Program, NJIT's SBR Program, and NJIT's Sponsored Chair in Manufacturing/Productivity Program. The assistance of P. R. Mitta and S. R. Kothakota in the preparation of this paper is appreciated.

VIII. REFERENCES

[1] AT&T, *Flexible Work Station: User Guide and Manual*, Version 1.01, 1989.

[2] M. O. Ball and M. J. Magazine "Sequencing of Insertions in Printed Circuit Board Assembly," *Operation Research*, 36(2), pp. 192-201, 1988.

[3] C. L.Beck and B. H. Krogh (1986). "Models for Simulation and Discrete Control of Manufacturing Systems," *IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, pp. 305-310, 1986.

[4] G. Chiola, "A Graphical Petri Net Tool for Performance Analysis," in *Modelling Techniques and Performance Evaluation*, S. Fdida and G. Pujolle (eds.), pp.323-333, North-Holland, NY, 1987.

[5] D. Crockett, A. Desrochers, F. DiCesare, and T. Ward, "Implementation of a Petri Net Controller for a Machining Workstation," *Proc. of IEEE Int. Conf. Robot. and Auto.*, Raleigh, NC. pp. 1861-1867, 1987.

[6] L. Decelle, "Design of a Robotic Workstation for Component Insertion," *AT&T Technical Journal*, Vol. 67, No. 2, pp.15-23, 1988.

[7] J. B. Dugan, A. Bobbio, A. Ciardo, and K. S. Trivedi , "The Design of a Unified Package for the Solution of Stochastic Petri Net Models," *1985 Int. Workshop on Timed Petri nets*, Totino, Italy, pp. 6-13, 1985.

[8] P. Freedman and A. Malowany, "Petri Nets and Modeling of Repetitive Events," *TR-CIM-88-5*, McGill University, Quebec, Canada, 1988

[9] B. Gasmi, C. Castel, C. Reboulet, and R. Houdebert, "SAFIR: Flexible Multi-Robots Assembly System," *Rensselaer's Int. Conf. on Computer Integrated Manufacturing*, pp. 582-587, Troy, NY, 1990.

[10] H. P. Hillion and J. Proth, "Performance Evaluation of Jop-shop Systems Using Timed Event-graph, " *IEEE Trans. on Automatic Control*, 34(1), pp. 3-9, 1989.

[11] M. C. Leu, H. M. Wong, and Z. Ji "A Genetic Algorithm for Solving Insertion Sequence and Feeders Placement Problems in PCB Assembly," Robotics and Automation Research Program Technical Report, New Jersey Institute of Technology, 1991.

[12] T. Murata, "Petri Nets: Properties, Analysis and Application," *Proc. of the IEEE*, 77(4), pp. 541-579, 1989.

[13] Narahari, Y. and N. Viswanadham, "A Petri Net Approach to the Modeling and Analysis of Flexible Manufacturing Systems," *Annals of Operations Research*, 3, 449-472, 1985.

[14] P. J. W. Nobel, *Printed Circuit Board Assembly*, Halsted Press, John Wiley & Sons, Inc., NY, 1989.

[15] A. Sciomachen, S. Grotzinger, and F. Archeti, "Petri Net-Based Emulation for a Highly Concurrent Pick-and-Place Machine," *IEEE Trans. on Robotics and Automation*, Vol. 6, No.2, pp. 242-247, 1990.

[16] J. Sifakis, "Use of Petri Nets for Performance Evaluation," in *Measuring, Modelling, and Evaluating Computer Systems*. North-Holland, pp. 75-93, 1977.

[17] M. Silva and R. Valette, "Petri Nets and Flexible Manufacturing," *Technical Report* GISI-RR-90.03, University of Zaragoza, 1990.

[18] Q. Xiang, *Computer-Aided Planning for Automatic Assembly of Printed Circuit Boards*, Master's thesis, Department of Mechanical and Industrial Engineering, New Jersey Institute of Technology, May 1990.

[19] M. C. Zhou, F. DiCesare (1989). "Adaptive Design of Petri Net Controllers for Error Recovery in Automated Manufacturing Systems," *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-19, 5, pp. 963-973.

[20] M. C. Zhou, F. DiCesare, and A. A. Desrochers, "A Top-down Modular Approach to Synthesis of Petri Net Models for Manufacturing Systems," *Proc. of IEEE Robotics and Automation Conference*, pp. 534-539, Scottsdale, AZ, 1989.

[21] M. C. Zhou and F. DiCesare, "A Petri Net Design Method for Automated Manufacturing Systems with Shared Resources," *Proc. of IEEE Int. Conf. Robotics and Automation*, Cincinnati, pp. 526-531, 1990.

[22] M. C. Zhou, *A Theory for the Synthesis and Augmentation of Petri nets in Automation*, Doctoral Dissertation, ECSE, Rensselaer Polytechnic Institute, Troy, NY, May 1990

[23] M. C. Zhou, F. DiCesare, and D. Rudolph (1990). "Control of a Flexible Manufacturing System Using Petri Nets," in *1990 IFAC Congress Conference*, Vol. 9, pp. 43-48, Tallin, USSR, August 1990.

[24] M. C. Zhou, F. DiCesare, and D. Guo, "Modeling and Performance Analysis of a Resource-Sharing Manufacturing System Using Stochastic Petri Nets," *Proc. IEEE Int. Symp. on Intelligent Control*, pp. 1005-1010, Phildelphia, PA, 1990.

[25] M. C. Zhou and F. DiCesare, "Parallel and Sequential Mutual Exclusions for Petri Net Modeling for Manufacturing Systems," Working Paper, ECSE, Rensselaer Polytechnic Institute, 1990.

[26] M. C. Zhou, F. DiCesare, and A. A. Desrochers, "A Formal Methodology for Synthesis of Petri Net Models for Manufacturing Systems," Working Paper, ECSE, Rensselaer Polytechnic Institute, 1990.

[27] M. C. Zhou and M. C. Leu, "Modeling and Analysis of a Flexible Assembly Station for Printed Circuited Boards Using Petri Nets", Robotics and Automation Research Program Technical Report, New Jersey Institute of Technology, 1991.