CHAPTER 1

INTRODUCTION

1.1 Background, Motivation and Objectives

Flexible and agile manufacturing is of increasing importance in advancing factory automation that keeps a manufacturer in a competitive edge. Flexibility signifies a manufacturing system's ability to adjust to customers' preferences and agility means the system's speed in reconfiguring itself to meet changing demands. Both together make it possible for manufacturers to respond instantly to the market. Design of such flexible and agile manufacturing systems requires tremendous team effort with information, knowledge and expertise from customers, system analysts, designers, and engineers in many disciplines such as manufacturing, industrial, mechanical, electrical, computer, software, and systems engineering. Approaches that allow all the involved persons to communicate and apply effectively at all the design stages have been sought by researchers and practitioners for the past several decades. This book aims to present a unified and consistent approach based on Petri nets (PN) to tackle numerous challenging problems that researchers and engineers encounter in the design of advanced manufacturing systems. The book focuses on the applications of Petri nets to modeling, analysis, simulation, control, and scheduling of flexible manufacturing systems (FMS).

The increasingly expensive hardware and software investment involved in building up a flexible manufacturing system forces engineers and designers to model it and experiment with the model using computers prior to its actual implementation. A model is a mathematical representation of the important features of the system. Conventional modeling tools such as continuous differential and difference equations or inequalities are not sufficient to deal with flexible manufacturing systems that can be characterized as discrete-event driven systems. These systems can be asynchronous, comprising many concurrent components or subsystem interacting in a complex way over time. Petri nets have been developed to serve this modeling purpose. By the manipulation of Petri net models, we hope that new knowledge about the modeled manufacturing systems can be obtained without danger, cost, or inconvenience of manipulating the real system itself. Such Petri net models can be conveniently analyzed with the computer tools available. They can be simulated under various operational policies such as push and pull strategies. They can also be evaluated for different schedules. Furthermore, these models can be used to derive the best schedule by applying such methods as heuristic search. They can serve as a basis for control software development for complex systems.

The main objectives of this book are as follows:

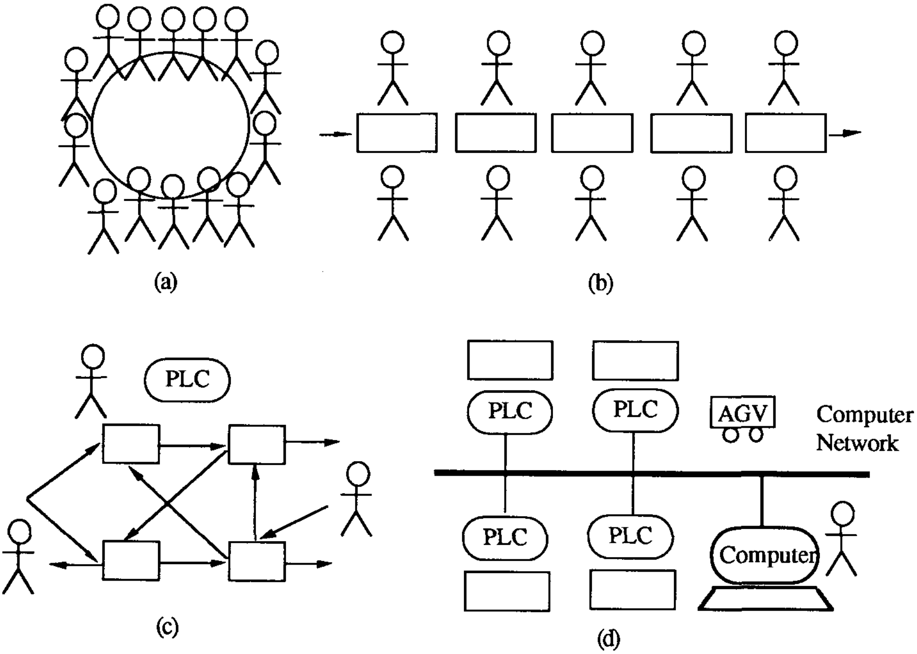
1. To overview flexible manufacturing, agile manufacturing, computer-integrated manufacturing as well as intelligent automation, and review applications of Petri nets as an integrated tool and methodology in FMS design;
2. To present the fundamentals of Petri nets and their properties and introduce special classes of Petri nets, and the modeling methods via several manufacturing systems;
3. To present timed Petri nets and the analytical approaches to their solution for performance evaluation of manufacturing systems;
4. To present Petri net simulation tools and their applications to the performance comparison of a flexible factory automated system with push and pull paradigms;
5. To present and apply augmented timed Petri nets for the simulation of robotic assembly systems with breakdowns;
6. To propose real-time Petri nets for discrete-event control of automated systems and compare with the conventional approaches such as relay ladder logic diagrams;
7. To present a methodology to use Petri nets in the object-oriented design of reusable control software; and
8. To present scheduling approaches using Petri nets and heuristic search algorithms and illustrate their advantages over the previous methods such as mathematical programming and rule-based approaches.

1.2 Historical Perspective on Manufacturing Systems

Production has evolved from labor-intensive production systems, mass-production lines, automated mass-production lines, jobshop, group/cellular manufacturing cells, to flexible manufacturing systems, agile manufacturing systems, and computer-integrated manufacturing systems. They can be shown roughly into four stages in Fig. 1.1.

Labor-intensive production systems remained as a primary way before the 19th century. Highly-skilled craftsmen made a complex product, e.g., watch, by using crude tools and materials. Human energy is a main energy source with limited usage of wind and water power. The invention and development of steam engines in the eighteenth century empowered human being in producing more complex products such as locomotives.

Mass production arose from the need to supply a large volume of uniform products, such as automobiles, to satisfy the market at the end of the last century. The early production and assembly systems were rigid and expensive, thus impossible to change to handle many variations in product types. Development of electrical devices such as electrical switches and motors led to better control of machines and resulted in machines with certain flexibility. Computer technologies allowed researchers and engineers to develop computer numerically controlled machines and programmable logic controllers to automate manufacturing operations. Recent developments in information technology including computer hardware/software and network techniques make it feasible to achieve the purposes of rapid product prototyping, concurrent engineering, flexible and agile automation and computer-integrated manufacturing.



(a) (b)

(c) (d)

Figure 1.1 Production evolvement (a) Labor-intensive systems, (b) Mass production lines, (c) Job-shops, and (c) Flexible manufacturing systems/Computer integrated manufacturing systems.

The invention and applications of steam engines and other water-driven power generators in the eighteenth century triggered industrial revolution. It freed people from heavy labor work. Meanwhile, the concept of the division of labor was developed and used to increase the productivity. Continuous flow manufacturing systems were introduced and shifted toward mass-production-oriented factories. Each individual was responsible for a small part of a product. Entering the nineteenth century, the communication and transportation infrastructure started to be significantly and increasingly improved for further industrial growth. Electricity was introduced to develop electrically powered machine tools. The internal combustion engines developed around 1900 helped mass-production of products such as automobiles. Large-scale assembly and mass production became very popular over the world. Another revolution lied ahead.

In the late 1940s, a first transistor was invented. Then integrated circuits as well as digital computers were developed in the 1950s. In the manufacturing field, a first numerically controlled milling machines was introduced. In 1967, computer numerically controlled (CNC) machines were developed and put into use. Integration of computer and machines greatly improved the flexibility associated with the machines and manufacturing systems. Fixed automation started to evolve into flexible automation. The fast development of computer and information technologies led to information revolution in many fields including manufacturing. In the early 1970s, computer-integrated manufacturing (CIM) was coined to provide a new concept and direction to grow manufacturing enterprises [Harrington, 1973]. The CIM concept has changed over time from computerized workcell, flexible manufacturing systems, large-scale automation, computer aided design and manufacturing (CAD/CAM), interfacing and communications concepts to the current state: an information system that controls data flow among all the function units in a manufacturing enterprise [Groover, 1980; Rembold *et al.*, 1993; Ranky, 1997]. CIM as a system provides computer assistance to all business functions within an enterprise - from customer needs, product design, order entry to product manufacturing and shipment. The goal of CIM is to reduce the product cycle time, improve the product quality and reliability, increase the productivity, and lower product cost, thus maintain and improve the manufacturing competitiveness in the world.

Implementation of successful CIM systems has taken place in many enterprises. It requires tremendous effort made by both management and engineering personnel in a company. A central technical challenge is the modeling, planning, scheduling, control and monitoring of automated factory systems, in particular, flexible manufacturing systems that are often most precious resources. This book is dedicated to the presentation of Petri net-based methods and tools to resolve this challenge facing to managers and engineers who are responsible for development of FMS and CIM systems. It also offers researchers with fundamentals of Petri nets and related approaches to study design issues of flexible manufacturing systems as well as computer integrated manufacturing systems.

1.3 Historical Perspective on Petri Nets

Petri nets have been originated from Carl Adam Petri's doctoral dissertation work on communication with automata in 1962. He described using a net the casual relationships between events in a computer system. His work came to the attention of A. W. Holt and others of the information System Theory Project of Applied Research, Inc. in the United State. Their work illustrated how Petri nets could be used to model and analyze systems of many concurrent components. Petri's work also came to the attention of The Computation Structure Group at Massachussetts Institute of Technology, led by Professor J. B. Dennis. Several doctoral theses and technical reports were published during the early 1970s. Most of the publications on Petri nets before 1980 were listed in the annotated bibliography of the first book on Petri nets [Peterson, 1981]. The work done in Europe on Petri nets and the published papers are annotated in the second Petri net book [Resig, 1985].

Starting in the late 70's, Petri nets became a very active area, especially in Europe. Annual conferences on Applications and Theory of Petri Nets have been held since 1979 and the proceedings published in the series of Lecture Notes of Computer Science by Springer Verlag. Most of the studies focused on information processing systems in the computer science community. An excellent tutorial paper was given by Professor T. Murata in 1989, which comprehensively presented properties, analysis, and applications of Petri nets and a list of references of significance [Murata, 1989]. Most of the earlier applications and theory of Petri nets aimed to information processing systems. The books [Peterson, 1981; Resig, 1985] and most of the papers were primarily targeted at computer scientists and graduate students.

Researchers with engineering background started their probe into the application of Petri nets in engineering systems particularly automated manufacturing systems in the early 1980's. They found that Petri nets were a powerful tool in describing event-driven systems. These systems may be asynchronous, contain sequential and concurrent operations, and involve conflicts, mutual exclusion and non-determinism. Such systems are termed as discrete event systems or discrete event dynamic systems (DEDS). The book [David and Alla, 1992], which is the first in its kind, presents Petri nets as a modeling tool of discrete event systems. The book [Zhou and DiCesare, 1993] focused on modeling and synthesis methods and discrete-event control of manufacturing systems. Another book [Desrochers and Al-Jaar, 1995] presented a comprehensive review of Petri nets in manufacturing automation and their applications to analysis, performance evaluation, and control of manufacturing systems. Other significant books and tutorials addressing Petri nets and their applications to automation include [Silva, 1985; DiCesare *et al.*, 1993; Proth and Xie, 1996; Silva and Vallette, 1990; DiCesare and Desrochers, 1991; Zhou and Robbi, 1994]. A tutorial on Petri nets and their industrial applications is presented in [Zurawski and Zhou, 1994]. A volume addressing the issues of flexible and agile automation using Petri nets is edited [Zhou, 1995]. Special issues on Petri nets and their applications were published, e.g., 1991 February issue of IEEE Transactions on Software Engineering on "Petri Net Performance Models," 1994 December issue of IEEE Transactions on Industrial Electronics on "Petri Nets in Manufacturing" [Zurawski and Zhou, 1994], and several to be published for IEEE Transactions on Semiconductor Manufacturing and Journal of Advanced Manufacturing Technology. These publications indicate the depth and breadth of the development and applications of Petri nets in the area of manufacturing automation.

While many activities are currently engaged in Petri net applications to the manufacturing automation area, the following foci could be observed since the late 70's [Zhou and Robbi, 1994]:

1. Early interest in Petri nets aroused from the need to specify and model discrete manufacturing systems. The activities in this area started with the Petri net representation of simple production lines with buffers, machine shops, and automotive production systems, and proceeded with modeling of flexible manufacturing systems, automated assembly lines, resource-sharing systems, and recently just-in-time and Kanban manufacturing systems. The work has continued to specify and model plant-wide production systems under different operational policies such as push and pull paradigms.
2. Early research focused on qualitative analysis of PN models of manufacturing systems. Reachability analysis shows whether a system can reach a certain state. Desired sequences of events are validated according to the system requirements. Other PN properties are used to derive the DEDS stability, cyclic behavior, and absence of deadlocks. Deadlock avoidance policies and their evaluation in a PN framework are still a hot research topic.
3. As temporal or quantitative properties become an important consideration, timed PNs are used to derive the cycle time of repetitive and concurrent manufacturing systems. To deal with the stochastic nature of many production operations, stochastic PNs are used to derive the system production rates or throughputs, critical resource utilization, and reliability measures. Their underlying models are Markov or semi-Markov processes. The direct construction of Markov chains is avoided thanks to the conversion algorithms for stochastic PNs.
4. When state explosion problems arise or the underlying stochastic models are not amenable for tractable mathematical analysis, simulation must be conducted for analysis of both qualitative and quantitative properties. Fortunately, PN models can be easily utilized to drive complex discrete event simulation. Several packages based on PNs exist for the simulation purpose. The use of simulation techniques to

evaluate flexible manufacturing systems and a fault-prone robotic assembly system will be explored later in this book.

5. Programmable logic controllers (PLCs) are commonly used in industrial sequence control of automated systems. They are designed through ladder logic diagrams that are known to be very difficult to debug and modify. It is observed that a PLC can be converted into a PN and vise versa for a subclass of PNs. Early work includes the conversion of a PN into a ladder logic diagram for PLC implementation. Direct PN controllers without help of PLCs can also be implemented through either a Petri net interpreter or their corresponding control code. For most cases, additional information to represent the real-time signals and status needs to be incorporated into such PN models. The advantages of PNs include their relative ease to represent and modify the control logic, and their potential for mathematical analysis and graphical simulation to validate a design. It can be proved that the graphical complexity of PNs grows with system complexity less than that of ladder logic diagrams. The comparison results between Petri nets and ladder logic diagrams will be presented later in this book.
6. With a mathematical representation available, designers are able to use PNs for rapid prototyping of a process control system or discrete event control. Virtual factories can be realized through computer graphics using PNs. Stepwise testing and implementation can be achieved by connecting the actual equipment into a PN-based design system, thereby reducing design and development time.
7. Petri nets have been combined with other approaches to achieve various purposes in process planning and scheduling, intelligent control, expert system construction, knowledge representation, and uncertainty reasoning. For example, the correspondence between a PN and an expert system can be established. This can greatly aid in consistency checking of an expert system. Optimal or suboptimal schedulers can be derived with heuristic search algorithms. The research over the past several years demonstrated the efficiency of the PN-based approaches.

1.4 A Robotic Cell and Petri Net Model

We conclude this chapter with an example that illustrates the issues in FMS design and the applications of Petri nets in resolving them. Several Petri net concepts will be introduced and demonstrated. If you do not understand all the concepts and methods discussed here, do not be concerned. They will be more fully developed and explained in more detail later.

Consider a Fanuc machining center located on the factory floor at the Center for Manufacturing Systems at New Jersey Institute of Technology. We start with a simplified version of the cell and finish with an FMS version. The actual cell is shown in Fig. 1.2. It comprises two workstations (WS), a milling machine WS1 and a drilling machine WS2, and a robot R. Consider a product type that needs only milling. Then the operational specifications for this cell are as follows:

1. To start a cycle, a raw part and the robot must be available.
2. The robot moves a raw part from the incoming conveyor, and loads it at WS1.
3. The milling operation is performed at WS1 while the robot backs off (returns).
4. The robot unloads the part from WS1 and deposits it on the outgoing conveyor, and returns.

The above steps repeat.

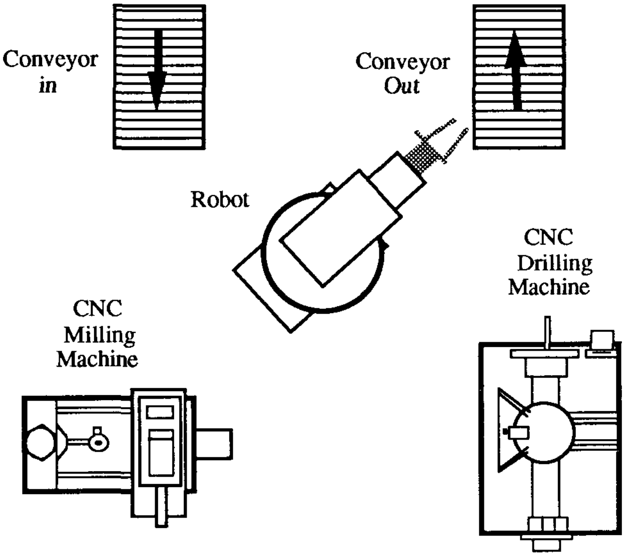


Figure 1.2 Fanuc System Layout.

In Petri net modeling, there are two nodes, places and transitions, represented by circles and bars, respectively. The former are used to represent the status of a resource, e.g., its availability, a process, e.g., its undergoing, or condition, e.g., its satisfaction. The latter is used to model events, e.g., start and end of an operation. A token's presence in a place indicates whether a resource is available, a process is undergoing, or a condition is true. A token is represented by a dot located in the circle representing a place. Multiple tokens often imply availability of multiple resources or the undergoing of operations of several parts. When the conditions for an event become all true, the corresponding transition is enabled and thus can fire. Firing enables the flow of tokens from places to places, implying the change of system status.

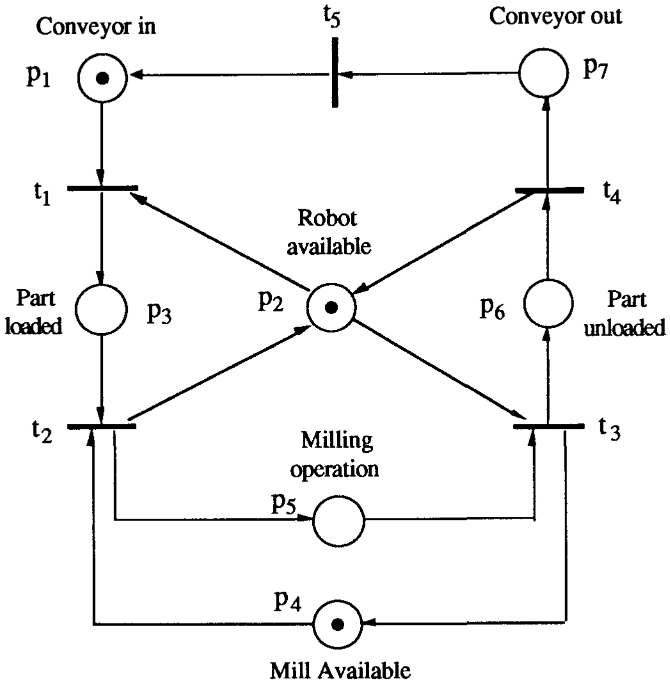


Figure 1.3 Petri Net Modeling.

For the Fanuc system, we determine the initial system conditions, events that can take place, several intermediate conditions and events, and finally the completion of the finished parts and restart the operations. In particular, we model the availability of a raw material in the input conveyor as a place p_1 and robot as p_2. We model the start of loading a part with Robot as transition t_1. Then two directed arcs are introduced linking from p_1 and p_2 to t_1 respectively as shown in Fig. 1.3. They mean that two conditions in p_1 and p_2 have to be met before the event in t_1 can happen. Next, firing t_1, i.e., occurrence of the event "start of loading" allows the robot to enter the status of "loading a part" modeled by place p_3. The arc from t_1 to p_3 is used to reflect this fact. Then the loaded part is available for Mill to initiate a milling operation. Place p_4 models "availability of Mill", and transition t_2 is used to model both "completion of robot's loading" and "start of milling operation". Two arcs leading from p_3 and p_4 to t_2 represent that t_2's being enabled requires two conditions met, i.e., a part being loaded and Mill being available. Two arcs from t_2 to p_2 (Robot available) and p_4 (Milling operation) signify that firing t_2 releases the robot and initiates milling. Applying the similar method, we modeled the remaining portions of the system. Transition t_5 is introduced to model the fact that the system is able to obtain a raw material once there is one product generated. Initially there is one robot and one miller idle. Thus p_2 and p_4 are marked with one token. Place p_1, however, can have one token to represent initially one raw piece available; two or more tokens represent two or more raw pieces available. In practice, the input raw piece may arrive faster than their processing. Then two or more pieces become available.

The dynamic behavior of the system can be observed through the model. When there is only one raw piece in place p_1, the net is executed as follows:

1. Initially only transition t_1 is enabled since only t_1's enabled conditions are met. Firing t_1, i.e., start of loading, removes two tokens in total from p_1 and p_2 and deposits a token to p_3. Places p_1 and p_2 hold no token and transition t_1 is disabled.
2. Only t_2 is enabled and firing t_2, i.e., completion of loading and start of milling, removes two tokens in total from p_3 and p_4 and deposits a token to p_2 and p_5, respectively. Since only t_1 related input and output places, i.e., $p_{1\text{-}3}$ are affected, only $p_{1\text{-}3}$ related transitions, i.e., t_1 and t_2, are checked.
3. Only t_3 is enabled and firing t_3, i.e., completion of milling and start of unloading, removes two tokens in total from p_2 and p_5 and deposits a token to p_6.
4. Only t_4 is enabled and firing t_4, i.e., completion of unloading, removes a token from p_6 and deposits a token to p_2 and p_7, respectively.
5. Only t_5 is enabled and firing t_5, i.e., releasing another raw material piece, removes a token from p_7 and deposits a token to p_1. The system returns to the initial condition and repeats the above process.

Therefore, the above model accurately describes the system's discrete-event behavior. A strict sequence of the events is followed when only one raw piece is available in place p_1.

Assume that two tokens are assigned to p_1, signifying the availability of two raw pieces initially. Then

1. Initially only transition t_1 is enabled. Firing t_1, i.e., start of loading, removes two tokens in total from p_1 and p_2 and deposits a token to p_3. Place p_2 holds no token and transition t_1 is disabled. However, place p_1 still has a token.
2. Only t_2 is enabled and firing t_2, i.e., completion of loading and start of milling, removes two tokens in total from p_3 and p_4 and deposits a token to p_2 and p_5, respectively.
3. Both t_1 and t_3 are enabled and firing t_3 will continue the system operations smoothly. Firing t_1, however, leads to an undesirable status, a deadlock. Firing t_1 removes two tokens in total from p_1 and p_2 and deposits a token to p_3. At this state, t_1 is not enabled due to the absence of tokens in p_1 and p_2, t_2 is not due to the absence of a token in p_4, and t_3 is not because of no token presence

in p_2, and t_4 and t_5 are neither because of no token in places p_6 and p_7. In other words, Robot is waiting for Mill to start milling operation of the loaded part, while Mill is waiting for Robot to unload its milled part. Such a circular waiting in an automated system is a catastrophe. This is what a designer has to avoid in the design of an automated manufacturing system.

Performance issues arise when operation times are considered. When there is only one token in the model, the cycle time for a product is simply the sum of all the times spent on loading, milling, and unloading plus the conveyor's transportation time in reality. The performance evaluation can be much more complicated when we consider randomness as well as failures of machines and robots. The evaluation of cycle time and productivity is necessary in order to optimize the system performance when several alternatives exist for system configurations, operational policies, etc.

When robot and machine failures are involved, different modeling paradigms have to be used for different faults. Since breakdowns may come any time, their real-time handling has to be effectively modeled. Petri nets can be augmented for this purpose. Furthermore, the optimal operational and/or design parameters of a system may have to be adjusted when breakdowns are taken into account.

Scheduling issues arise whenever there are shared resources or alternative routes in an FMS. Resource sharing happens at two levels in a flexible manufacturing system. At the physical level, robots, automatic guided vehicles, conveyors, and related transportation systems are shared by machines. At the job level, all the flexible machines are shared by different types of jobs. The advantages of Petri net approaches to be fully demonstrated later are that since the model captures all the precedence and concurrent relations among operations, the resulting schedule is automatically deadlock-free, and two level resource-allocation problems are resolved in one step. Many existing approaches such as integer programming cannot handle these without significantly increasing the solution complexity [Xiong, 1996]. Suppose that the FMS in Fig 1.2 produces alternatively Type A parts using only Mill and Type B parts using first Drill and then Mill by using the raw material. Then Fig. 1.4 is the model for the system to produce Type B products only. It models the sequence of loading, drilling, milling, and unloading processes for Type B products. The places such as p_1, p_2, p_4, and p_7, and transition t_5 remain unchanged while others are introduced to model the Type B related events and operations.

In order to obtain a global net model, common places, transitions, and arcs are shared and integrated into the model shown in Fig. 1.5. To guarantee the alternative production of types A and B, a pair of control places, i.e., p_{14} and p_{15} are introduced. Thus initially both t_1 and t_6 are enabled. Firing either one will disable itself without firing the other one. For batch processing, two output places need to be introduced to count the number of Type A and B products generated, respectively as shown in Fig.

1.5. Locating the optimum schedule can be then converted into the search of the optimum path from the initial marking to the desired final marking.

The complexity of a Petri net model becomes a problem from a graphical point of view. Fortunately, its model has its underlying mathematical representations through matrices. In computer implementation, a special structure, e.g., link structure, can be used to reduce the memory requirements for a huge matrix format. The enabling and firing rules can be easily implemented to perform scheduling and discrete event control. The graphical form can be made available in a modular way for a quick visual analysis and help designer debug and maintain the system.

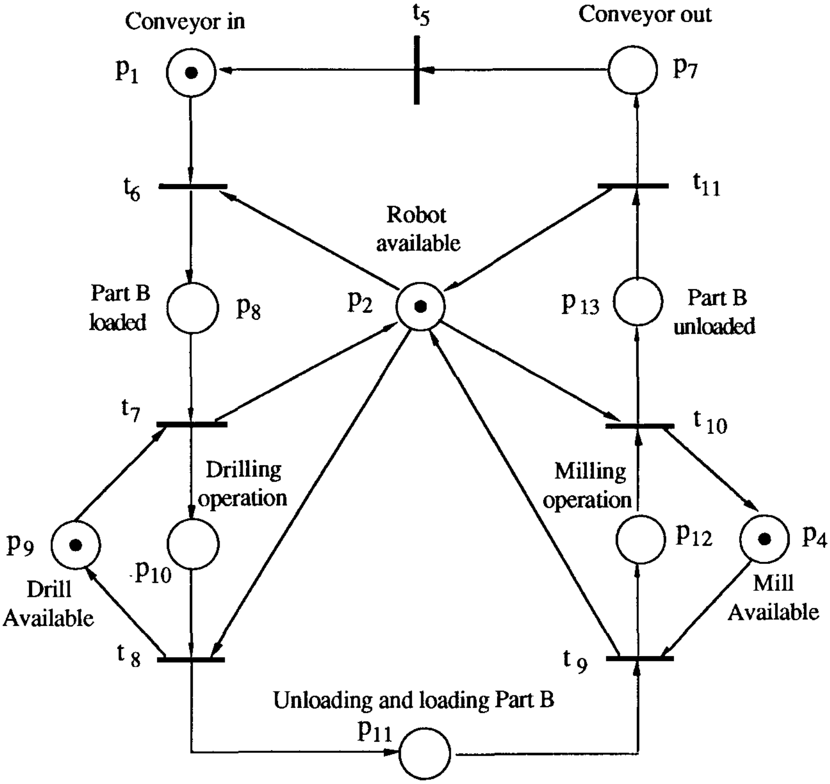


Figure 1.4 Petri net model for Part Type B with the Fanuc cell.

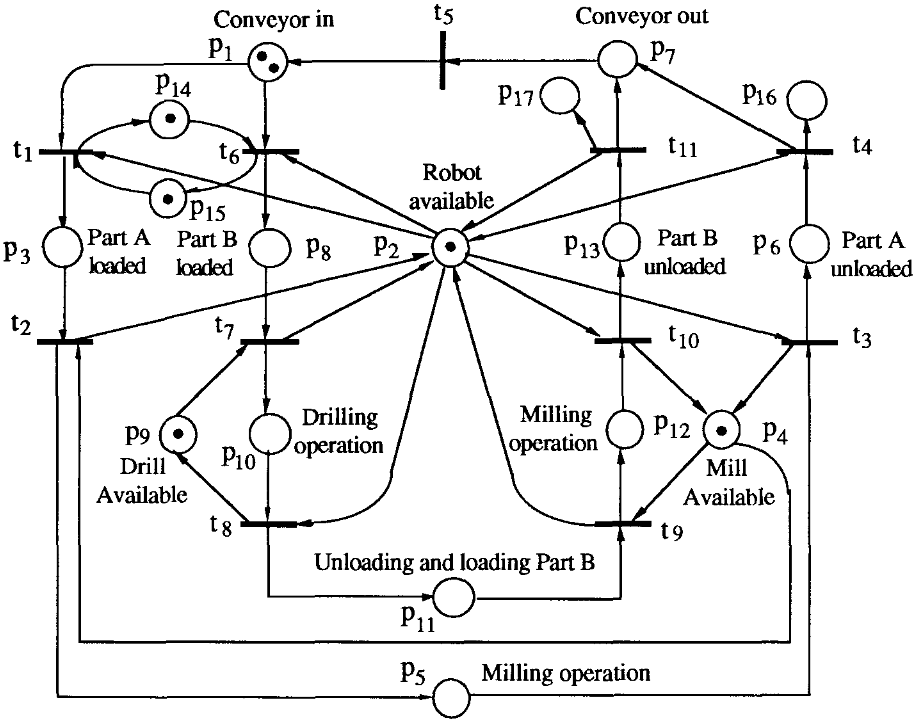


Figure 1.5 A Petri net model of two job types for the Fanuc cell.

1.5 Summary

The flexible manufacturing system (FMS), a job shop of sorts, presents formidable design problems. Formal discrete event system models are useful to predict operational characteristics such as throughput rates and utilization of machines and to perform scheduling and discrete event control of the FMS. The modeling using Petri nets for the Fanuc cell is intuitively conducted and the related issues to system design are discussed. These issues include deadlock avoidance, system behavior modeling, performance evaluation, scheduling, fault modeling and handling, as well as discrete event control.

An FMS features diversity in material movement and job types, requiring flexible automation. Flow shop production can be efficiently implemented by hard automation, with material movement handled by conveyors and fixed-movement robots. In an FMS material flow is mechanized by programmed robots, Automatic Guided Vehicles (AGVs), and people. Since these devices are more costly than their hard automation counterparts, the system design must permit them multiple roles, allowing competition for their services. System designs that allow for concurrent activities and competition for resources are difficult to realize. Discrete event system technology can help the designer predict behavior and tune system performance consistent with economic goals.