PETRI NET BASED DECISION SYSTEM MODELING IN REAL-TIME SCHEDULING AND CONTROL OF FLEXIBLE AUTOMOTIVE MANUFACTURING SYSTEMS

(GERÇEK ZAMANLI PLANLAMADA PETRI NET BAZLI KARAR VERME SİSTEMİ MODELLEMESİ VE ESNEK OTOMOBİL ÜRETİM SİSTEMLERİNDE KONTROLÜ)

by

Özkan BAŞAK, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assoc. Prof. Dr. M. Ebru ANGÜN

Jan 2023

PETRI NET BASED DECISION SYSTEM MODELING IN REAL-TIME SCHEDULING AND CONTROL OF FLEXIBLE AUTOMOTIVE MANUFACTURING SYSTEMS

(GERÇEK ZAMANLI PLANLAMADA PETRI NET BAZLI KARAR VERME SİSTEMİ MODELLEMESİ VE ESNEK OTOMOBİL ÜRETİM SİSTEMLERİNDE KONTROLÜ)

by

Özkan BAŞAK, B.S.

Thesis

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Supervisor: Assoc. Prof. Dr. M. Ebru ANGÜN

Jan 2023

This is to certify that the thesis entitled

PETRI NET BASED DECISION SYSTEM MODELING IN REAL-TIME SCHEDULING AND CONTROL OF FLEXIBLE AUTOMOTIVE MANUFACTURING SYSTEMS

prepared by **Özkan BAŞAK** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Date: -------------------------

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my previous supervisor Prof. Dr. Y. Esra ALBAYRAK for her continuous support of my M.Sc. study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my M.Sc. study. I would like to thank Assoc. Prof. Dr. S. Emre ALPTEKİN for his guidance, support and stimulating discussions. My sincere thanks also goes to my current supervisor Assoc. Prof. Dr. M. Ebru ANGÜN for her support, advices, guidance, valuable comments, suggestions and her provisions that benefited much in the completion of this study and Prof. Dr. Temel ÖNCAN for dynamism, vision, sincerity and motivation have deeply inspired me.

Besides my supervisors, I would like to thank to my thesis committee: Prof. Dr. Mehmet Mutlu YENİSEY, Assoc. Prof. Dr. Tuncay GÜRBÜZ, and once again Assoc. Prof. Dr. M. Ebru ANGÜN, for their encouragement, insightful comments and questions.

Last but not the least, I am very much thankful to my wife Selay BAŞAK and my daughter Elis Derin BAŞAK for their love, understanding, patience and continuing support to complete this research work.

I am extending my thanks to Valeo Automotive for their support to do this work. Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Jan 2023
Özkan BAŞAK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Product form queuing networks and generalized stochastic Petri nets (PNs) have emerged as the principal performance modeling tools for flexible manufacturing system (FMS). A flexible automotive manufacturing system used in this environment to react for some amount of flexibility that allows the system to react in case of changes, whether predicted or unpredicted, enables quick cell configuration, and the efficient operation, of cells. In this research, we attempt to propose a flexible automotive manufacturing approach, more over this approach should be used for modeling and analysis of shop floor scheduling problem of FMSs using high-level PNs. We assumed to implement efficient production control, this study provides an object-oriented Petri nets (OOPNs) approach to performance modeling. The graphical nature and mathematical foundation has made PNs based methods appealing in real-time scheduling and control for flexible automotive manufacturing systems (FAMSs) which gives a significant time advantage to real-time scheduling and controlling of flexible automotive manufacturing systems.

Keywords : Petri net, Decision system modeling, Time scheduling, Flexible automotive manufacturing systems, Timed marked graph (TMG), Linear programming (LP)

ÖZET

Ürün formu kuyruk ağları ve genelleştirilmiş stokastik Petri Ağları (PA), Esnek Üretim Sistemi (EÜS) için temel performans modelleme araçları olarak ortaya çıkmıştır. Öngörülen veya öngörülemeyen değişiklikler durumunda, sistemin tepki vermesine izin veren bir miktar esneklik için tepki vermek üzere, bu ortamda kullanılan esnek bir otomotiv üretim sistemi, hızlı hücre konfigürasyonunu ve hücrelerin verimli çalışmasını sağlar. Bu araştırmada, esnek bir otomotiv imalat yaklaşımı önermeye çalışıyoruz, bu yaklaşımdan daha fazlası, yüksek seviye PA'lar kullanan EÜS'lerinin atölye çizelgeleme probleminin modellenmesi ve analizi için kullanılmalıdır. Çalışmada verimli üretim kontrolü uygulayacağımızı varsaydık; bu çalışma, performans modellemeye Nesne Yönelimli Petri Ağları (NYPA) yaklaşımı sağlıyor. Grafiğin doğası ve matematiksel temeli, esnek otomotiv üretim sistemlerinin gerçek zamanlı programlanması ve kontrolünde önemli bir zaman avantajı sağlarken, esnek otomotiv üretim sistemleri (EOÜS) için gerçek zamanlı çizelgeleme ve kontrolde PA'ya dayalı yöntemleri cazip hale getirmektedir.

Anahtar sözcükler : Petri ağı, Karar sistemi modelleme, Zaman çizelgeleme, Esnek otomotiv üretim sistemleri, TMG, Doğrusal programlama (DP)

1. INTRODUCTION

Flexible manufacturing systems (FMS) represent an important recent development in manufacturing automation, on the account of improved productivity, quality and resilience to demand fluctuations. An FMS consists of numerically controlled machining centers linked together by an automated material handling system, and can be quickly configured to produce multiple type of products. Flexibility in a manufacturing system provides various advantages such as increased productivity, reduced work-in-process inventory, reduced lead times, increased machine utilization, and reduced set-up costs.

Despite these productivity improvements, with the changing face of production and the supply chain, today's manufacturers will need to be flexible and adaptable. The world is on the verge of witnessing unprecedented efficiency in the manufacturing business using modern technologies. A flexible manufacturing system allows for a degree of flexibility (Javaid, Haleem, Singh and Suman, 2022). The main characteristics of an FMS include multi-layer resource sharing and routing flexibility of the jobs.

Due to the high complexity of automotive FMSs, some mathematical programming related methods such as integer programming, linear programming, and dynamic programming often lacks to describe the practical constraints, and dynamic features of complex scheduling problems, or lack of providing analytical solutions within a reasonable time. Besides, analytical methods usually require repetition as the status of the system changes, and assume that all the parts are ready at time zero, which is rarely the case in practice. On the other hand, classical scheduling techniques such as branch-and-bound and neighborhood search techniques cause substantial increase in state enumeration and exponential increase in computation time as the problem size increases.

In this perspective, either existing techniques are improved or new scheduling approaches are developed to deal with the practical constraints of the real-world scheduling problems of automotive FMSs. These methods include simulation based approaches, expert systems, PNs based methods, and hybrid methods (Lin and Lee, 1997). Petri nets are a powerful graphical tool for modeling and analyzing concurrent, parallel, simultaneous, synchronous, distributed, and resource sharing systems. There are many advantages of using Petri nets such as enabling an easy visualization of complex systems (Kaid, El-Tamimi, Nasr and Al-Ahmari, 2015).

The analysis of FMS for design and control is complex. Leonard and Brown (2015) have explained the basic structure of the FMS control hierarchy. In the literature, to deal with design complexity of FMS, a number of analytical techniques are available. The simulation of FMS is a field of research for many people. Rybicka (2017) and Parhi and Srivastava (2018) suggested the simulation as a decision support tool for defining the configuration and the control of an FMS. Manu and Vijay Kumar (2020) presented a simulation study using Taguchi’s method analysis of physical and operating parameters of the flexible manufacturing system along with flexibility. Jain, Sandhya Maheshwari and Baghel (2008) developed queuing model for the performance prediction of FMSs. Zhanga, Li, Zhanga, and Wang (2020) used an ant colony optimization approach for scheduling of FMS for a given level of flexibility. Angraa, Chandab and Chawlaa (2018) considered the run times and routes in multi-load automatic guided vehicles (AGVs) system for simulation study of dispatching rule performance. A simulation model suggested using static dispatching rules. Chawla, Chanda, Angra and Rani (2019) controlled the efficiency and productivity of the FMS using the performance of material handling operations of the system. In this study, priority hybrid dispatching rules are used and simultaneous scheduling of work centers and AGVs became highly imperative for the sustainable material handling operations in the FMS.

Mathematical modeling plays a vital role in the design, planning and operation of flexible manufacturing systems (FMSs). In the study of Ram and Viswanadham (2017), attention is focused on stochastic modeling of FMSs using Markov chains, queuing networks, and stochastic Petri nets. The paper of Singha, Singh and Khan (2016) focuses on the mathematical modelling of loading problem in FMS with heuristics. Wang, Pang and Ng

(2019) developed a simulation model to investigate the FMS performance under uncertain machine failure disruptions. In this study, machine status defined to analyze performance measures of a FMS under different operational conditions, including disrupted machine can be replenished to continue the job processing. Han, Chang, Hong and Park (2020) present a work to create an information system able to integrate different viewpoints concerning the Internet of Things (IoT) Based Data Collection System. The proposed approach tackles the problem from the information system point of view. Gamila and Motavalli (2003) developed a mathematical model to select machines and assign operations and the required tools to machines in order to minimize the summation of maximum completion time, material handling time, and total processing time. They formulate loading and routing, two of the most important FMS planning problems, as a 0–1 mixed integer programming problem. Yadav and Jayswal (2017) outlined and illustrated the basic approaches to the development of such models will be outlined and illustrated with examples pertaining to FMS. Main purpose of this research to collect information, related to modelling of FMS in one document for serving the researchers to get a clear vision of what has been investigated and what still needs to be investigated. The objective of Aized's (2009) study is to model and maximize performance of an integrated Automated Guided Vehicle System (AGVS) with multi-line FMS. The system is modeled using colored Petri net method (CPN).

A Petri net is a mathematical tool that can best describe FMSs properties. In the study of Blaga, Pop, Indre, Hule and Stănăşel (2022) main focus is the evaluation of the performance of manufacturing cells with Petri nets, in this context layout of the manufacturing cell and modeling are proposed. Incremental computations have been studied in various types of complex discrete event systems. Cherif, Leclercq and Lefebvre (2023) suggested working on a new algorithm based on a new filtering mechanism that uses the cost function to selectively explore the state of the Petri net model of partial routing of FMS in uncertain environments to find a control sequence with a trade-off between performance and robustness. Mejia and Pereira (2020) proposed a general solution for multiobjective scheduling for flexible manufacturing systems with Petri nets, including three alternative approaches with an indirect scheme to represent the solution.

In addition to the decision support and performance evaluations evaluated in most of the studies, other important factors for the FMSs are the robustness of the control mechanism, deadlock detection and recovery. The paper of Bashir, Zhou and Muhammad (2021) highlights combining both control places and transitions to construct a new method for the supervisory structure which smoothed the operational flow in FMSs. Deadlocks may occur in an FMS and may cause the system to be blocked indefinitely. Lu, Chen, Li and Wu (2022) present partial deadlock markings to detect with their relationships on loop graphs for shared resources by different processes.

From the above-cited literature, Since FMS is a technology, well-adjusted to the environmental needs, we have to manage it successfully. Several method and content used on the research area and have significant results. The literature review also identifies a potential research gap to compare and evaluate a more precise application of PN based decision system modeling in the respect of real-time scheduling for FAMSs by using time marked graphs. Due to time constraints of automotive production processes, this approach gives us an advantage on dynamic decision capability. The main contribution of this thesis, with our proposed algorithm, we use weighted functions and markings (initial marking and final marking) that the algorithm can decide for optimal solution, even faces with deadlocks, in line with our strategy. As a conclusion, FMASs used to demonstrate the method's applications and evaluate the time advantage for application of automotive industry.

This thesis is organized as follows. In Section 2, main concepts of FMS and PNs are given. In Section 3, object oriented modeling is presented. In Section 4, we introduce an illustrative model for FMS, and specially emphasize the different ways for obtaining a closed loop model. Also, we introduce the scheduling objectives and outline the problems arising during the transformation of the initial model into an unfolding TPN. In Section 5, analysis of performance of FMS is reviewed. In Section 6, reasons to use petri nets are presented. Finally, in Section 7, conclusions are given.

2. MAIN CONCEPTS

2.1 Flexible Manufacturing System (FMS)

In the middle of the 1960s, market competition became more intense. During 1960 to 1970 *cost* was the primary concern. Later *quality* became a priority. As the market became more and more complex, *speed of delivery* became something customer also needed. A new strategy was formulated: *Customizability*. The companies have to adapt to the environment in which they operate, to be more *flexible* in their operations and to satisfy different market segments. Thus the innovation of FMS became related to the effort of gaining competitive advantage.

First of all, FMS is a manufacturing technology. Secondly, FMS is a philosophy and "system" is the key word. Philosophically, FMS incorporates a system view of manufacturing. The new word for today's manufacturer is "agility". An agile manufacturer is one who is the fastest to the market, operates with the lowest total cost and has the greatest ability to "delight" its customers. FMS is simply one way that manufacturers are able to achieve this agility. An MIT study on competitiveness pointed out that American companies spent twice as much on product innovation as they did on process innovation. We need to keep in mind what Peter Drucker said: "We must become managers of technology not merely users of technology".

Since FMS is a technology, well-adjusted to the environmental needs, we have to manage it successfully. Today flexibility means to produce reasonably priced customized products of high quality that can be quickly delivered to customers. Different approaches to flexibility and their meanings are shown Table 2.1.

Table 2.1: Approaches to flexibility and their meanings

Approach	***Flexibility meaning***
Manufacturing	·The capability of producing different parts without major retooling · A measure of how fast the company converts its process(es) from making an old line of products to produce a new product · The ability to change a production schedule, to modify a part, or to handle multiple parts
Operational	· The ability to efficiently produce highly customized and unique products
Customer	· The ability to exploit various dimension of speed of delivery
Strategic	· The ability of a company to offer a wide variety of products to its customers
Capacity	· The ability to rapidly increase or decrease production levels or to shift capacity quickly from one product or service to another

While variations abound in what specifically constitutes flexibility, there is a general consensus about the core elements. There are three levels of manufacturing flexibility:

Basic flexibilities:

- Machine flexibility: the ease with which a machine can process various operations
- Material handling flexibility: a measure of the ease with which different part types can be transported and properly positioned at the various machine tools in a system
- Operation flexibility: a measure of the ease with which alternative operation sequences can be used for processing a part type

System flexibilities:

- Volume flexibility - a measure of a system's capability to be operated profitably at different volumes of the existing part types
- Expansion flexibility - the ability to build a system and expand it incrementally
- Routing flexibility - a measure of the alternative paths that a part can effectively follow through a system for a given process plan
- Process flexibility - a measure of the volume of the set of part types that a system can produce without incurring any setup
- Product flexibility - the volume of the set of part types that can be manufactured in a system with minor setup

Aggregate flexibilities:

- Program flexibility - the ability of a system to run for reasonably long periods without external intervention
- Production flexibility - the volume of the set of part types that a system can produce without major investment in capital equipment
- Market flexibility - the ability of a system to efficiently adapt to changing market conditions
- Today's manufacturing strategy is to seek benefits from flexibility. This is only feasible when a production system is under complete control of FMS technology. Having in mind the Process-Product Matrix you may realize that for an industry it is possible to reach for high flexibility by making innovative technical and organizational efforts. See the Volvo's process structure that makes cars on movable pallets, rather than an assembly line. The process gains in flexibility. Also, the Volvo system has more flexibility because it uses multi-skill operators who are not paced by a mechanical line.
- So we may search for benefits from flexibility on moving to the job shop structures.
- Actually, the need is for flexible processes to permit rapid low cost switching from one product line to another. This is possible with flexible workers whose multiple skills would develop the ability to switch easily from one kind of task to another.
- As main resources, flexible processes and flexible workers would create flexible plants as plants which can adapt to changes in real time, using movable equipment, knockdown walls and easily accessible and re-routable utilities.

Advantages:

- Faster, lower- cost changes from one part to another which will improve capital utilization.
- Lower direct labor cost, due to the reduction in number of workers.
- Reduced inventory, due to the planning and programming precision.
- Consistent and better quality, due to the automated control.

- Lower cost/unit of output, due to the greater productivity using the same number of workers.
- Savings from the indirect labor, from reduced errors, rework, repairs and rejects.

Disadvantages:

- Limited ability to adapt to changes in product or product mix.
- Substantial pre-planning activity.
- Expensive, costing millions of dollars.
- Technological problems of exact component positioning and precise timing necessary to process a component.
- Sophisticated manufacturing systems.

2.2 Flexible Automotive Manufacturing System

In the automotive industry, a successful manufacturer must make efficient use of limited resources to meet dynamic customer demand as customer preferences change on a seasonal or monthly basis. To better meet customer demand, automotive producers organize their factories into product layouts, where each cell manufactures a production lot composed of auto of a product family (Blackburn, 1991), and each operation of the lot requires a specific machine type(s).

The advantages of a product layout are that product flow is simplified to flow lines, machines in stations are closer together to reduce travel time, and employees within a cell are more involved as a team. Use of a flexible manufacturing system creates the additional advantages of quick cell creation for a new product family by simply reprogramming the FMS to route autos, and efficient operation of each cell by automatically transferring automotive between machines in stations. The backbone of an FMS for automotive production is a circular, unidirectional, automated material handling system. A cell in this environment is a cluster of stations, where each station is a physical location that accommodates an operator and a finite buffer. The FMS is responsible for routing auto between the stations.

2.3 Petri nets (PNs)

The concept of the Petri net (PN) has its origin in Carl Adam Petri's dissertation (Petri, 1962), submitted in 1962 to the faculty of Mathematics and Physics at the Technical University of Darmstadt, West Germany. The dissertation was prepared while C.A. Petri worked as a scientist at the University of Bonn.

Petri nets are graphical and mathematical modeling tool applicable to many systems. They are used to describe nondeterministic, stochastic information systems. As a graphical tool, Petri nets can be used as a visual communication aid similar to flow charts, block diagrams, and Networks. In addition tokens are used in these nets to simulate the dynamic and concurrent activities of systems. As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of systems. Petri nets can be used by both practitioners and theoreticians. Thus, they provide a powerful medium of communication between them: practitioners how to make their models more realistic.

In this study, we given the only rule one has to learn about Petri Net theory: the rule for transition enabling and firing. Although this rule appears very simple, its implication in Petri-net theory is very deep and complex.

A Petri net is a particular kind of directed graph, together with in initial state called the initial marking. The underlying graph N (N is an integer value) of a Petri net is a directed, weighted, bipartite graph consisting of two kinds of nodes, called places and transitions, where arcs are either from a place to a transition or from a transition to a place. In graphical representation, places are drawn as circles, transitions as bars or boxes. Arcs are labeled with their weights, where a k-weighted (k is an integer value) arc can be interpreted as the set of k parallel arcs.

In modeling, using the concept of conditions and events, places represent conditions, and transitions represent events. A transition has a certain number of input and output places representing the preconditions and post-conditions of the event, respectively. The presence of a token place is interpreted as a holding the truth of the condition associated with the place to indicate that k data items or resources are available. Some typical interpretations transitions and their input places and output places are shown in Table 2.2.

Table 2.2: Some typical interpretations of transitions and places

Input Places	***Transition***	***Output Places***
Preconditions	Event	Post conditions
Input data	Computation step	Output data
Input signals	Signal processor	Output signals
Resources needed	Task or job	Resources released
Conditions	Clause in logic	Conclusions
Buffers	Processor	Buffers

PNs are proven tools enabling analysis of concurrent processes performed with the use of shared resources (machines, equipment, operators, etc.). An unmarked Petri net is formally defined as a 3-tuple $C = (P, T, A)$ or a 4-tuple $C = (P, T, I(IN), O(OUT))$ where P is a finite and non-empty set of places, T is a finite and non-empty set of transitions t, I is a mapping from transitions to a bag of places such that I_t defines the input places of transition t, and O is a mapping from transitions to a bag of places such that O_t defines the output places of t. A PN can also be represented by a bipartite directed graph with two types of nodes: circles for places and bars for transitions. Directed arcs connect places and transitions (Santarek, K., Buseif, I, M., 1998).

$P = \{p_1, p_2, ...p_i\}$ is a finite set of places,
$T = \{t_1, t_2, ...t_j\}$ is a finite set of transitions with $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.
and where IN: (P x T) → N is an input function that defines directed arcs from places to transitions and where OUT: (P x T) → N is an output function that defines directed arcs from transitions to places.

$A = \{a_1, a_2, ...a_j\}$ is a finite set of arcs,
$A \subseteq (P \times T) \cup (T \times P)$ is a finite set of arcs from places to transitions or from transitions to places.

For an ordinary PN, all arcs have unitary weight, and no multiple arcs are allowed. A PN structure without a specific initial marking is denoted by N, and a PN with a given initial marking is denoted by (N, M_o). In a basic form, a marked PN is defined as a 5-tuple, $G =$

(P, T, A, W, M_0); where W: $A \rightarrow \{1,2,...\}$ is the weight function attached to the arcs (A is an integer value) and $M_0 : P \rightarrow \{0,1,2,...\}$ is the initial marking. An exemple is shown in Fig. 2.1.

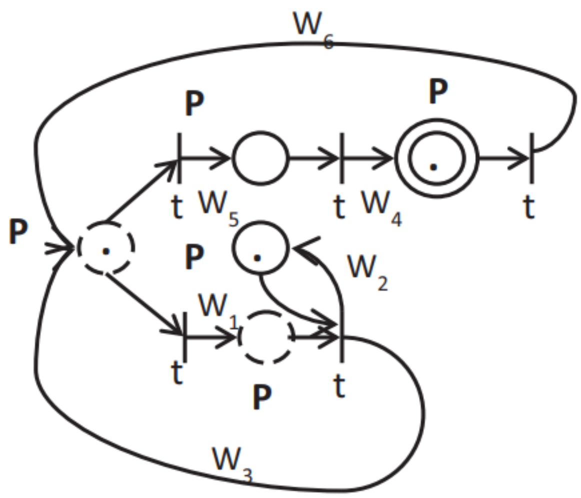


Figure 2.1: Basic form of marked PN

Places (P) are represented by dashed circle (indirect places), solid circle (direct places) and double dashed/solid circles (deadlock places), transitions (t) by segments, in particular, $M_0(1) = M_0(2) = M_0(5) = 1$, and $M_0(3) = M_0(4) = 0$, i.e., $P_0 = \{1, 2,...,5\}$.

A given marking, M_i indicates the current state of a PN, and it is denoted by M, an n x 1 vector, where n is the total number of places. The p^{th} component of M, denoted by $M(p)$, is the number of tokens in place p. A marking changes when a transition "fires", i.e., when an event or activity occurs. Transition firing is equivalent to a state change and describes the PN's dynamic behavior.

Considering the set of functions of input places (I) and of output places (O) for a transition t, as $I(t) = \{p| (p, t) \subset A\}$, and $O(t) = \{p \mid (t, p) \subset A\}$, a transition t is enabled in a marking M_i if and only if $M_i(p) \geq w(p, t)\ \forall p \in I(t)$, where $w(p, t)$ is the weight of the arc from p to t.

If IN $(p_i, t_j) = k$, where $k \geq 1$ is an integer, a directed arc from place p_i to transition t_j is drawn with label k. If IN$(p_i, t_j) = 0$, no arc is drawn from p_i to t_j. If OUT$(p_i, t_j) = k$, a directed arc is included from transition t_j to place $p_{i,}$ with label k if $k > 1$ and without label if k = 1. If k = 0, no arc is included from t_j to p_i (Wiswanadham, N., Narahari, Y., Johnson, T.L., 1990)

A transition without any input place is called a source transition, and one without any output place is called a sink transition. A source transition is unconditionally enabled. A sink transition consumes tokens, but does not produce any. A source transition is often used, in a manufacturing system model, to represent the arrival of raw material or semi-finished parts in the system, while sink transitions are often used to represent the departure of finished or semi-finished parts.

A pair of a place p and a transition t is called a self-loop if p is both an input and output place of t. A PN is pure if it has no self-loops. A PN with self-loops can be converted to a pure PN by adding a place and transition to each self-loop in the original PN.

An infinite capacity net is a PN where each place can accommodate an unlimited number of tokens. A finite capacity net is a PN where each place, p, has an associated capacity, $K(p)$, meaning that p can hold a maximum of $K(p)$ tokens. A finite capacity net can be converted to an infinite capacity net using the complementary-place transformation rule (Murata, 1989).

The zero testing, i.e. the ability to test whether a place has no token, can be done with PNs by using an inhibitor arc in the model. In the presence of the inhibitor arc, a transition is regarded as enabled if each input place, connected to the transition by a normal arc, contains at least the number of tokens equal to the weight of the arc, and no tokens are present on each input place connected to the transition by the inhibitor arc. The transition firing rules are the same as for normally connected places. The firing, however, does not change the marking in the inhibitor arc connected places.

A PN as a graphical and mathematical tool possesses a number of properties, and these properties allow one to perform a formal check of the properties related to the underlying system's behavior. Some of the important properties are as follow (Zhou and Xiong, 2001):

- A PN is said to be *K*-bounded or simply bounded if the number of tokens in each place does not exceed a finite number *K* for any marking reachable from M_o. PN is said to be safe if it is 1-bounded. For a bounded PN, there are a limited number of markings reachable from the initial marking M_o through firing various sequences of transitions.
- A PN is said to be live if, no matter what marking has been reached from M_o, it is possible to ultimately fire any transition of the net by progressing through some further firing sequence. This means that a live PN guarantees deadlock-free operation, no matter what firing sequence is chosen.
- A PN is said to be reversible if for each marking *M* in $\in R\ (Z, M_o)$, M_o is reachable from *M*. Therefore in a reversible net one can always get back to the initial marking.

The boundedness, liveness, and reversibility of PNs have their significance in manufacturing systems. Boundedness or safeness implies the absence of capacity overflows. Liveness implies the absence of deadlocks. This property guarantees that a system can successfully produce without being deadlocked. Reversibility implies cyclic behavior of a system and repetitive production in manufacturing. It means that the system can be initialized from any reachable state.

3. MODEL AND METHODOLOGY

3.1 Object Modeling of the FMS

In developing an object-oriented control model, it is needed to define all the object classes required for the FMS, then the statistical relationships among these object classes are described. The object modeling technique (OMT) method, which is the most widely used methodology, is employed to describe and analyze the object classes. It divides the system considered into object classes and it is used to model the static relations among FMS objects by the class structure. The behavior of the objects is described by operations associated with object classes.

Aggregation can be described the “part-whole” or “a-part-of” relationship in which objects representing the components of a structure are associated with an object representing the entire assembly (Booth, 1998). Generalization or inheritance describes the relationship between a class and its refined version, and is used to share the similarities among classes while preserving their diversities (Kuo et al.1998). Aggregation is drawn with a small diamond indicates the assembly end of the relationship, and the notation for inheritance is a triangle connecting a superclass to its subclasses.

3.2 Object-Oriented Modeling (OOM) of the FMS

The OMT diagram of the system incorporates the distributed decision making environment. After the OMT diagram is developed, dynamic behavior and control logic of the FMS is formulated by constructing high-level PN model of the system based on the static relations of the OMT model.

Individual classes of FMS objects are defined as follows:

- Load/unload station class

- Workstation class
- Part transport Class
- Operator class
- Scheduler Class

Since the standard PNs can be insufficient to model the complex and large size systems, they have been extended to the High-level PNs which allow arbirary complex data types for tokens and to let expressions for places and transitions be constructed by using some special programming languages.

Operations planning and scheduling (OPS) problems in automotive FMSs, are composed of a set of interrelated problems, such as part-type batching, machine grouping, part routing, tool loading, part input sequencing and on-line scheduling. Prior studies in literature point out that the performance of an automotive FMS is highly dependent on the efficient allocation of limited resources to tasks, and it is affected by the choice of scheduling rules. In this section, a conceptual framework of a high-level PN based Decision Support System using Object-oriented design approach, which integrates loading, part inputting, routing, and dispatching issues of the OPS is discussed.

In this study, Artifex PN software tool (RSoft Design Group, Inc., 2002) is employed to model the system. Artifex exploits a graphical language, which is based on extended Petri Nets, to define detailed models of a system or process. The Artifex language is integrated with standard programming languages like C and C++. This allows developers to use existing software components within their models, and to use C and C+ programming beside the Artifex language where more appropriate, as an example implement algorithms and to perform complex operations on data.

3.3 Resource-Oriented Modeling (ROM) of the FMS

For analysis and control, the concurrent processes of a manufacturing system are modeled by production-process-oriented PN's by some researchers. "Production-process-oriented model" means that the production process (or sequence of operations) is addressed and the resources are treated as conditions of operations in the model. Thus, each operation of every part type is modeled by different places or transitions and each resource in the system has multiple corresponding places in the model. Such a model cannot model the sharing of resources strongly, even it is decision free, and model is not good for deadlock control because it is the resource allocation law that controls deadlocks in the system.

To describe the characteristics of concurrent contention for resources, we use a resource-oriented model instead of a production-process-oriented one. Such a model mainly describes the dynamics of resource allocation instead of the production processes of operations in the system. In this model, each resource in a system is modeled by a single place. If a resource is shared by several operations due to the operation sequences of parts, then the corresponding place will be an output place of several transitions.

There are many kinds of resources in an FMS, such as machines, buffers, AGV's, pallets, tools, and fixtures. In general, before a part is moved to a machine for processing, the tools needed to process the part are already in the magazine of that machine. Therefore, tools do not contribute to deadlocks. Similarly, pallets and fixtures are irrelevant to deadlocks. Machines, buffers and AGV's are the main resources in the production process in FMS and impact the occurrence of deadlocks. In this study, however, we take only machines and buffers into consideration.

It is assumed that 1) each machine has one input buffer which is denoted by $p(\mathrm{I})$, and 2) no job visits the same machine more than one time consecutively. By Assumption 1, if a part requires processing by a machine, it must be sent to the input buffer of that machine first, where it waits for the machine to be available. From Assumption 2), the next place to which a part completed on a machine goes is the input buffer of another machine or load/unload station.

Because of Assumption 1), ROPN (Resource Oriented Petri Net) models each machine and its input buffer as a primitive. Two places $p(\mathrm{I})$ and $p(\mathrm{O})$ are used to model such a primitive: one for the input buffer and the other for the machine. The capacity of the places represents the number of parts that can be held by the buffer or machine. A token in a place means that there is a part in the buffer or on the machine. A transition connecting these two places represents the operation process of loading a part onto the machine (see Fig. 3.1). The multiple input transitions of place $p(\mathrm{I})$ in Fig. 3.1 represent the sharing of the machine and its buffer by multiple operations. The multiple output transitions of place p(O) represent the selective output. For the sake of simplicity, we often write $\{u_i\}$ for $\{p_i(\text{I}), t_i, p_i(\text{O})\}$, or u_i denotes the primitive of resources corresponding to machine i . Notice that because p(O) represents a machine the capacity of p(O) must be of 1.

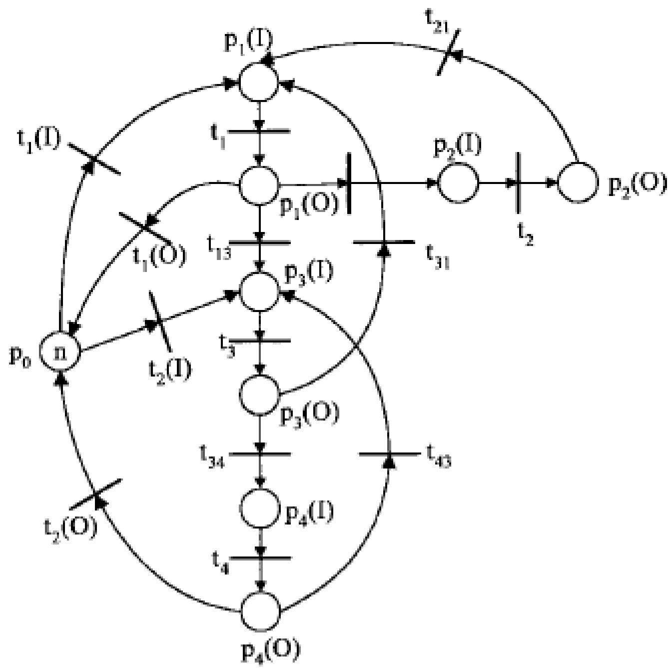


Figure 3.1: Resource-oriented petri net (ROPN)

Based on the primitive, we can model the production processes with routing flexibility as follows. Let Q denote the set of types of parts to be produced by the FMS. The operation sequence for a part type is prescribed and some operations may be processed by alterative machines because of the routing flexibility. When an operation requires machine i to process, the resources required for the operation correspond primitive i in our model. Assume that the number of operations of part type q ϵ Q is L_q , and let $A_q = \{A_q(1), A_q(2), \ldots, A_q(L_q)\}$ denote the required sequence of producing the part type, where $A_q(i) \neq \emptyset$ is a set of primitives corresponding to the alterative machines for operation i . For instance, if $A_q = \{u_1, u_2, (u_3, u_1)\}$, then operation 1 requires machine 1 to process, operation 2 requires machine 2, and operation 3 can be processed either by machine 3 or by machine 1. We call $(a_q(i), a_q(i+1))$ a primitive pair from $A_q(i)$ and $A_q(i+1)$ if $a_q(i) \in A_q(i)$ and $a_q(i+1) \in A_q(i+1)$. In the primitive pair, $a_q(i)$ is the upstream primitive of

$a_q(i+1)$. To model the production process of a part type is to add a transition between each primitive pair from $A_q(i)$ and $A_q(i+1)$ for $i=1,2,\ldots,L_q-1$, respectively, such that the transition is the output transition of an upstream primitive and the input transition of a downstream primitive. Finally, an input (output) transition is added to each primitive $a_q(1) \in A_q(1)$ $a_q(L_q) \in A_q(L_q)$). These transitions are connected through a place representing the load/unload station. Therefore, to complete the modeling of the system, we first choose a part type arbitrarily, say, q_1, and carry out the above procedure. Then, choose q_2 and do the same, but this time, if u_i and u_{i+1} are the primitive pair for q_2 and there is already a transition connecting these two primitives, then we do not add new transition to connect these two primitives. Carrying out this procedure for all other part types, we obtain the so-called ROPN because we address the resources in FMS's and each resource corresponds only one place in the model. Let n_{qi} denote the number of parts for part type q_i and $n = \Sigma\, n_{qi}$. Without loss of generality, we let $M_o\,(p_o) = n$ and $M_o\,(p_i) = 0$, $i \neq 0$ be the initial marking, where this implies that in the initial state, all the parts to be processed are in the load/unload station.

Fig. 3.1 shows that the ROPN for a system contains four machines and three types of parts, where the operation sequences of the three types of parts are $A_1=\{u_1, u_3, (u_4, u_1)\}$, $A_2=\{u_3, u_1, (\{u_3, u_4\}, \{u_2, u_1\})\}$, and $A_3=\{u_3, u_4, u_3, u_1\}$. The parts of type 1 can be released into the system from p_0 to u_1 by going through t_1(I) . The third operation can be processed by machine 1 or 4, or the parts can go from p_3(O) to p_1(I) or p_4(I) through t_{31} or t_{34} . When a part is completed by machine 1 (machine 4), it goes back to p_0 through t_1(O) (t_2(O)). The flow of the other part types is modeled similarly by the ROPN. It should be pointed out that according to the operation sequences, there are four operations for part type 2. The third operation can be processed by machine 3 or machine 2. If it is processed by machine 3, the forth operation should be processed by machine 4. If it is processed by machine 2, the forth operation should be processed by machine 1.

In a PN each place (transition) can be regarded as a node. A PN is said to be a strongly connected if and only if there is a directed path from any node to any other node. A PN is called a state machine if each transition $t \in T$ has exactly one input place and exactly one output place. Observing the ROPN, we know that structurally, the ROPN is a strongly connected state machine PN: a class of free choice PN. The ROPN is also a finite capacity

PN; all the places other than p_o (without loss of generality, we assume that p_o has infinite capacity) have finite capacity; therefore, the enabling and firing rule is given by (1) and (2). In ROPN, when a token is in a place and (1) is satisfied for one of the output transitions of that place, we say that the transition is process-enabled, or we say, without confusion, that the token enables that transition. When (2) is satisfied for a transition, we say that the transition is resource enabled. Because of the exclusive use of a resource in manufacturing systems, each of the arc weights in an ROPN is 1. It should also be noticed that there is at most one transition connecting two primitives in the ROPN.

Because of the strongly connectedness of the ROPN, the ROPN will contain a number of circuits. The production process circuit in the ROPN is a special class of circuits that plays an important role for the liveness of the ROPN.

4. APPLICATION

The design process of FMSs is a relatively complex task. In the proposed approach, we firstly construct an object modeling diagram of an automotive FMS and develop a heuristic rule-base to solve the resource contention problem, then formulate the dynamic behavior of the system by high-level PNs. The modeling methodology is illustrated by an automotive FMS, and the proposed rule-based system is compared with several dispatching rules with respect to part flow time and tardiness related performance measures. Finally, the system performance under different operational configurations is analyzed to find out the best implementing policy for the system under consideration.

Its modeling methodology using high level PNs and Object-oriented design approaches has also been presented. Firstly, object modeling diagram of the system is constructed and a heuristic rule base is proposed to solve the resource contention problem, then the dynamic behavior of the system is formulated by high-level PNs. Subsequently, the proposed rule-based system is compared with common dispatching rules with respect to part flow time and tardiness related performance measures, then the simulation experiments were carried out under different experimental conditions to analyze the impact of the different levels of independent variables on performance measures.

4.1 Petri Net Model of the Valeo Transmission Turkey FMS with Key Factors

In general, an FMS is structured as a set of workstations, where products must be processed, and a flexible transport system, the goal of which is to load and unload the workstations. The sequence of operations performed in order to manufacture a product is what we call a working process (WP). Roughly speaking, a deadlock is a system state so that some working processes can never be finished.

We have adopted Petri Nets to model the dynamic behavior of the working processes. The use of the Petri Net analysis theory will give us the techniques for checking interesting properties about the good behavior of the system and also some “hints” on how to avoid non desirable situations.

In the Petri Net Model of the Valeo FMS, a place represents one of the following; available machines in a workstation, busy machines in a workstation, blocked machines in a workstation, parts waiting in an input buffer, parts waiting in an output buffer, and fixtures of a particular type.

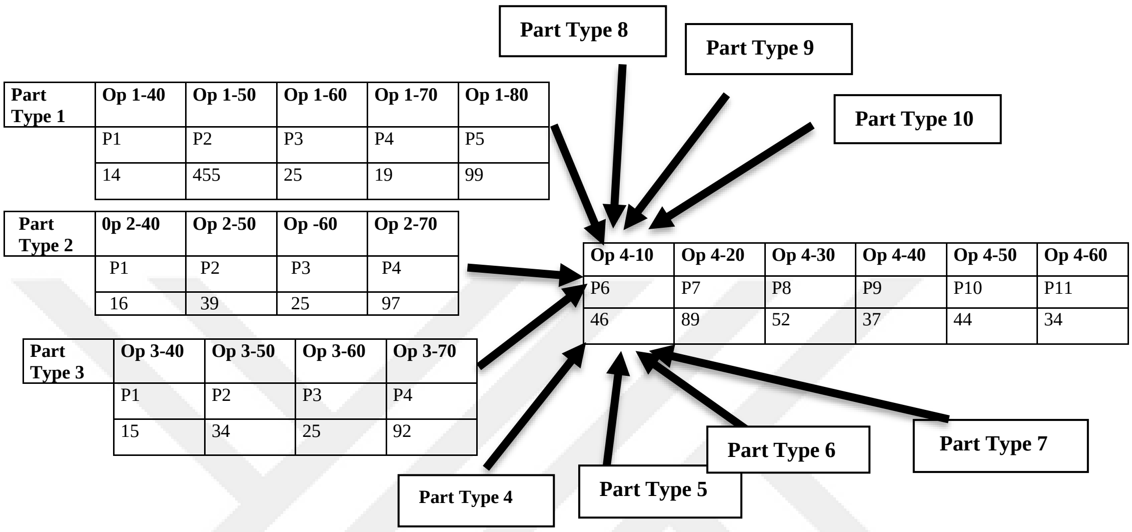
4.2 The Modeling Working Process in Valeo Turkey FMS Automotive Production Line

In Table 4.1 the model of the working process is presented. In each table, first row shows operation flow node, the second machine options for the operations, and the third row gives corresponding processing times. This figure is completed with the resource places used by the WP (places represented in Fig. 4.2 and Table 4.3). Part I, Part II and Part III are semi-finished goods for final assembly line, also Part 4 to 10 are raw materials. Operations 4-10 to 4-60 show briefly WP of final assembly of Clutch disc production line. Due to process flow, all material handling operations are managed by Robots / Automatic Guided Vehicles (AGVs) /Manipulators / Automatic Conveyors. The marking of Manipulator / Robot / AGV models availability of both machines and the robot, respectively (we assume that each resource can hold only one product at a time) (see Fig. 4.1).

Valeo strictly obsessed to eliminate all non-value added operations (MUDAs). Automatic carriage systems help to focus easily on our working processes.

To define refined modeling, we prefer to use just one final assembly line with key customer objectives. If application would be positive, flow can be distributed and applied from supplier to customer. Working process is shown for Valeo Turkey – Bursa Manual Clutch Disc Assembly Line. 3 sub-assembled groups are Metallic Disc, Retainer Plate and Drive Plate products. These lines have more than 500 parts with unique customer references and have the core spirit of FMAS in very accurate raw materials.

Table 4.1: Continuous automatic flow of Valeo FMS for one piece



Operation	Machine option
P1	M1 or M2
P2	M3 or M4 or M5 or M6
P3	M7 or M8 or M9 or M10
P4	M11
P5	M12 or M13
P6	M14 or M15
P7	M16
P8	M17
P9	M18 or M19
P10	M20 or M21
P11	M22 or M23

Table 4.2 shows weight (W) ranking (W: A → {1, 2,...}), Weight functions attached to the arcs are selected as an integer value, also it could be directly linked with process' importance according to customer specifications [Failure Mode Effect Analysis (FMEA) study gives direct weight of this value].

Table 4.2: Weight ranking for arcs

Weight Ranking	**Value**
Not at all important	1
Slightly important	2
Important	3
Fairly important	4
Very important	5

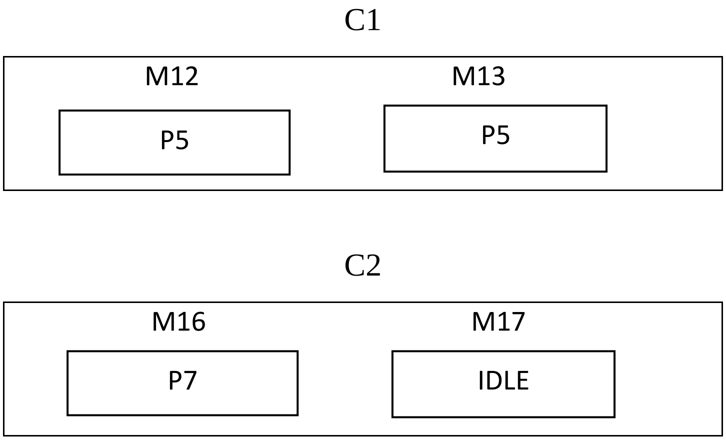


Figure 4.1: Deadlock conditions for Valeo FMS

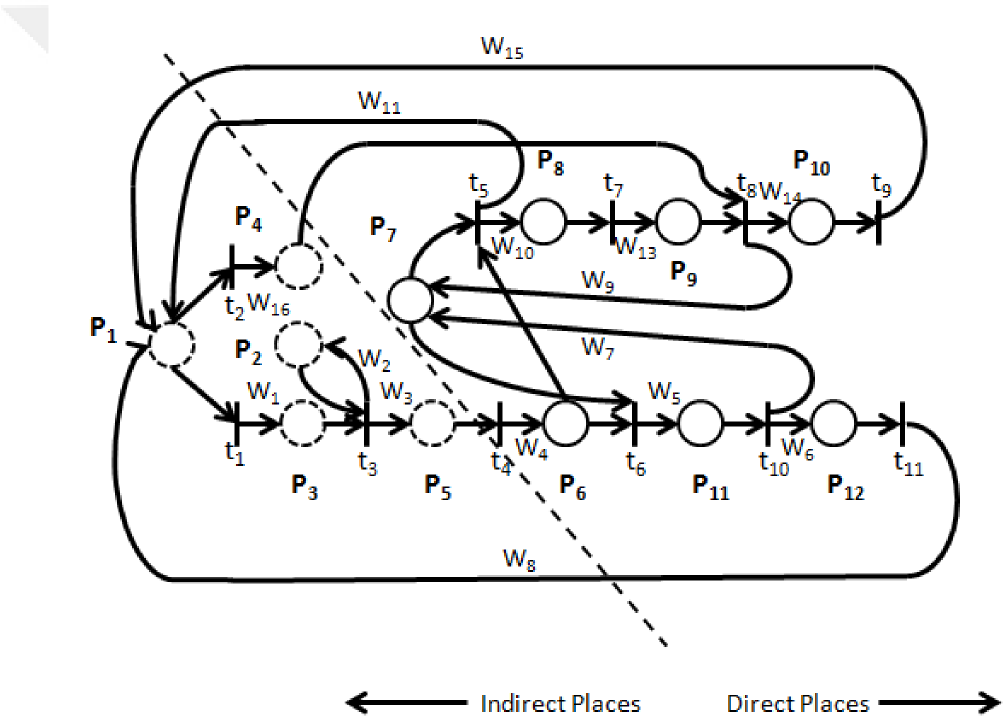


Figure 4.2: Modelling places (direct & indirect due to processing structure) and transition – transformed petri net flow

Table 4.3: Modeling places (direct & indirect due to processing structure) and transition descriptions

	Places
	Indirect Places on Processing
P1	Manipulator / Robot / AGV is available
P2	Semi-finished goods / Raw materials are available
P3	Manipulator / Robot / AGV is available to transfer Semi-finished goods / Raw materials
P4	Manipulator / Robot / AGV is available to transfer Finished goods
PS	Manipulator / Robot / AGV to transfer Semi-finished goods
	Direct Places on Processing
P6	Manipulator / Robot / AGV with Semi-finished goods / Raw materials, waiting for the **Final Assy Line**
P7	Final Assy Line is available
P8	Manipulator / Robot / AGV is released. Final Assy Line Processing
P9	Final Assy Line waiting for Semi-finished goods / Raw materials
P10	Manipulator / Robot / AGV unloading processed Finished goods
P11	Manipulator / Robot / AGV is not released. Final Assy Line Processing
P12	Manipulator / Robot / AGV unloading while Processing
	Transitions
T1	Manipulator / Robot / AGV assıgned to Semi finished goods / Raw materials
T2	Manipulator / Robot / AGV assıgned to Finished goods
T3	Manipulator / Robot / AGV starts transportıng Semi finished goods / Raw materials
T4	Manipulator / Robot / AGV carrying Semi finished goods / Raw materials to the Final **Assy Line**
T5	Manipulator / Robot / AGV released after finding Final Assy Line free
T6	Manipulator / Robot / AGV not released after finding Final Assy Line free
T7	Final Assy Line Processing a part; Manipulator / Robot / AGV released
T8	Manipulator / Robot / AGV starts unloading a Finished goods
T9	Manipulator / Robot / AGV carrying a finished good to unloading station
T10	Final Assy Line processıng a part; Manipulator / Robot / AGV not released
T11	Manipulator / Robot / AGV to transfer to unloading while processing

4.3 Marking on Petri Net Modeling

A marking is a vector $M: P \rightarrow N$ N that assigns to each place of a P/T (net system) a non-negative integer number of tokens, represented by black dots. The marking of place p is denoted as M (p).

A P/T system or net system $< N, M_0 >$ is a net N with an initial marking M_0. The dynamic behavior of the PN is defined by its marking M, where M is a state vector with M(p) being the number of tokens in place p. Transitions allow the net to change its state , i.e. the dynamics of the PN are controlled by firing its transitions. A transition t is enable when each place in its input set •t contains at least one token. A transition t fires by removing tokens from its input places and depositing tokens at its output places. If a transition is

enabled, it can fire. This implies the removal of one token from each place in •t and the introduction of one additional token in each place in the output set t•.
Each marking defines a system state. Firing a transition may result in a new state. All possible states define the state space of the PN. A live and consistent PN is cyclic, which is a typical property of manufacturing systems.

In Fig. 4.3 application of new modeling with marked Petri Net, we assumed to start with 3 (M_0) initial markings (P_1: Manipulator/Robot/AGV is available, P_2: Semi-finished goods/Raw materials are available and P_7: Final Assembly Line is available), due to this assumption we can highlight the main objective to see indirect materials of Final Assembly Line, which can be obviously seen by applications efficiency at t_0 and which weight functions should be chosen for which activity as placed with Petri-Net modelling.

To take a decision and to configure main marked model, optimal cycle times are defined with real on-time scheduling and optimal cycle time studies. We sliced net can be precisely calculated with token and firing transitions.

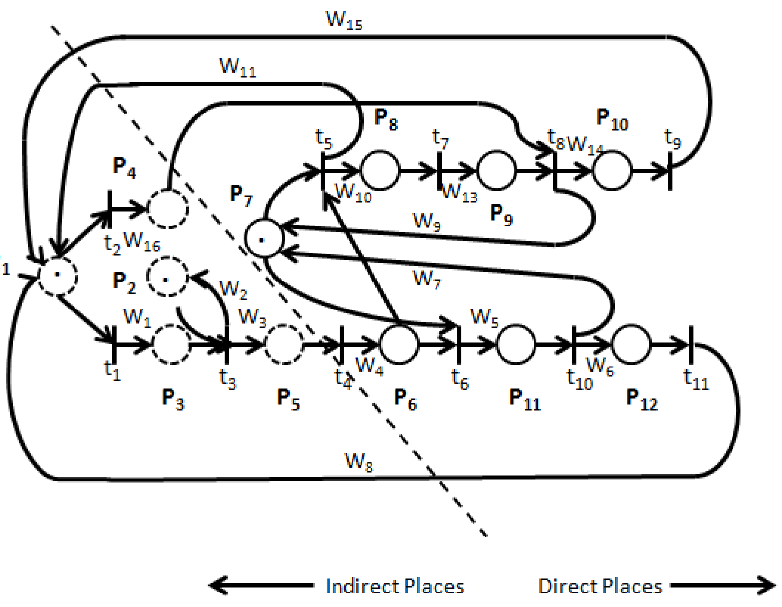


Figure 4.3: Modelling Places (Direct & Indirect due to processing structure) and Transition – Initial Marked & Transformed Petri net flow

4.4 Deadlocks in Automated / Flexible Manufacturing Systems

Automated manufacturing systems, including FMS's, belong to the class of discrete event dynamical systems (DEDS) that are gaining in prominence in the recent literature. In a typical FMS, raw parts of various types enter the system at discrete points of time and are processed concurrently, sharing a limited number of resources such as numerically controlled (NC) machines, robots, material handling system (MHS), fixtures, and buffers. In such resource-sharing systems, deadlocks constitute a major issue to be addressed at the design and operation phases. A deadlock is a highly undesirable situation in which each of a set of two or more jobs keeps waiting indefinitely for the other jobs in the set to release resources. The occurrence of a deadlock can cripple the entire system and renders automated operation impossible. In addition, a deadlock, occurring in a subsystem of the given system, can propagate to other parts of the system, eventually completely stalling all activities in the entire system. Deadlocks usually arise as the final state of a complex sequence of operations on jobs flowing concurrently through the system and are thus generally difficult to predict. In an improperly designed FMS, the only remedy for deadlock may be manual clearing of buffers or machines and restart of the system from an initial condition that is known to produce deadlock-free operation under normal production conditions. Both the lost production and the labor cost in resetting the system in this way can be avoided by proper design and careful operation.

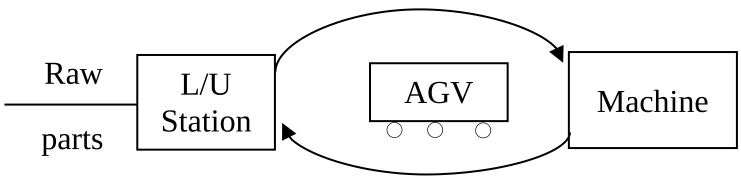


Figure 4.4: Simple manufacturing system comprising an AGV and an NC machine

To visualize a simple example of a deadlock in a manufacturing system, consider the system depicted in Fig. 4.4. There is a load/unload (L/U) station at which raw parts are always available. An AGV carries a raw part from the L/U station to an NC machine, which carries out some operations on the raw part. The finished part is carried by the AGV to the L/U station, where it is unloaded. It is assumed that the AGV can only carry one part at a time, and the NC machine can only process one part at a time. In addition, the AGV takes a certain amount of time to carry a part from L/U to machine or from machine to L/U. However, if it is not carrying a part, it can travel very quickly between the L/U and AGV. Imagine the following sequence of events, starting with an initial state in which the AGV and the machine are free, and raw parts are available: 1) The AGV carries a raw part, say part 1, and loads it onto the NC machine, which starts processing part 1; 2) the AGV returns to the L/U station and carries another raw part, say part 2, to the machine but waits for the machine, which is still processing part 1. Thus, the AGV gets blocked waiting for the machine; 3) the machine finishes the operations on part 1 and starts waiting for the AGV to carry the finished part 1 to the L/U station. At this juncture, the machine gets blocked waiting for the AGV. If the machine and the AGV can only accommodate one part at a time and there is no additional buffer space, the two resources here are then involved in a deadlock since each keeps waiting for the other indefinitely. Even if some buffer space is provided for raw parts and finished parts in the above system, a deadlock can still occur because the AGV can fill the entire buffer with raw parts during the processing of part 1 by the machine.

In the recent literature, several efforts have focused on the problem of deadlocks in automated manufacturing systems. One of the major traditional applications of PN's has been in the deadlock analysis of concurrent systems. In manufacturing systems, studies on deadlocks, using PN-based models are presented in and these studies essentially prove the existence (absence) of deadlocks using the invariants of the PN model. In this research, we address the important issues of prevention and avoidance of deadlocks in automated manufacturing systems using PN-based techniques. The terms prevention and avoidance have been used in the Computer Science literature on deadlocks to mean static and dynamic policies, respectively, for eliminating deadlocks. It is known that deadlock prevention policies that are usually implemented in the design stage lead to inefficient resource utilization. Deadlock avoidance policies that can be enforced during the operation of a system lead to better resource utilization and throughput.

4.5 On-line Controller for Deadlock Avoidance

We now present an on-line controller for deadlock avoidance in any FMS using PN's. The controller is basically an on-line monitoring system. Fig. 4.5 shows the components of the proposed controller. These components are described below:

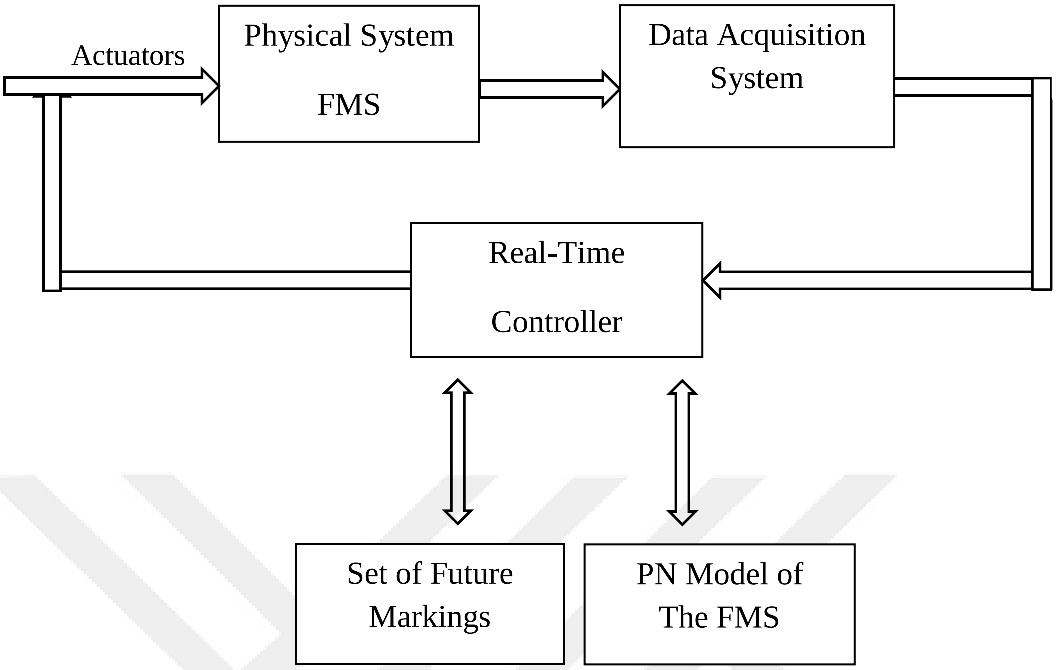


Figure 4.5: Controlling System

Physical System: This block corresponds to the actual FMS in operation.

Data Acquisition System: This unit is responsible for gathering, using various sensors, status information of all resources in the FMS. The output of this unit can be used to determine the current marking of the PNM of the FMS.

Petri Net Model: This corresponds to a data structure that efficiently stores the PNM of the FMS. The construction of this model can be carried out easily using the paradigm of union of PN's, as is detailed in this study. This data structure also includes a field for the current marking, which is updated constantly by the real-time controller.

Set of Future Markings: This is another data structure that efficiently stores the sets L,(M), L,(M);.., L,(M), where n is the look ahead employed and M is a current marking. These sets are crucially used by the real-time controller to select the immediate transitions to fire. When the marking of the PNM changes, the L sets for the new marking can be computed easily from those of the current marking.

Real-time Controller (RTC): The inputs to this unit are the look ahead to be employed and the sensor output data for the current state of the FMS. The controller has access to the two data structures, namely, PNM and the set of future markings. This unit mainly performs three functions in each iteration.

1) Determination of the current marking of the PNM.
2) Classification of the current marking into deadlock,
3) Looking ahead into the system evolution and initiation

The RTC first checks if the previous marking, say P (not to be confused with the notation for the set of places for a tangible marking, or vanishing marking of appropriate actions.

Appendix A gives (in a Pascal-like language) an algorithm that describes the working of the RTC in each iteration. It can be seen that such a controller can, in principle, be implemented for real-world FMS's such as the GE FMS.

5. ANALYSIS OF PERFORMANCE OF FMS

High-volume automated manufacturing has obvious cost advantages due to economies of scale. FMSs offer these advantages to the manufacturer with the flexibility for manufacturing a variety of parts in small quantities. In the FMS which is highly integrated to the system, performance evaluation is the main focus. However, the problem of obtaining an optimal schedule is very difficult because of the large number of factors which influence the performance measures and the inter-relationships between various components of the FMS. The better performance of the FMS results in reduced labor costs, increased output, decreased manufacturing costs, increased flexibility, and reduced production lead time.

5.1. Analysis of performance measures of FMS using PN

As described above, modeling and the optimal design of FMS is a critical and complex problem. Standard PN's do not include any time concept. Therefore, with a standard PN it is possible to describe only the logical structure of systems, and not their time evolution. In order to study performance of PN models, the duration of activities must also be taken into account and included into model specifications. Minimize the total time required and the set-up costs, meet the due date, maximize the machine utilization are the major performance criteria to evaluate FMS. TPNs are an extended class of Petri nets that allows for the description of firing times in the modeled system. In TPNs, which have been introduced by Ramchandani, each transition takes a 'real time' to fire, which determines the duration of transition's firings. The firing times may be either deterministic or stochastic. Thus, the performance evaluation can be conducted by incorporating both deterministic and/or stochastic time functions into performance measures. Viswanadham and Raghavan (2000) investigated a dynamic modeling technique for analyzing supply chain networks using stochastic timed Petri nets. Tuysuz and Kahraman (2010) presented an approach for modeling and analysis of time critical, dynamic and complex systems using stochastic Petri nets together with fuzzy sets. Liu, Lin, and Ren (2013) present a novel fault diagnosis and cause analysis model using fuzzy evidential reasoning approach and dynamic adaptive fuzzy Petri nets to address the problems.

5.2. Performance evaluation of marked graphs by linear programming

Performance evaluation is an important issue for discrete event dynamic systems which can be described and modeled through the help of PN. In this section, to obtain and analyze the performance measures of FMS, an implementation of PN is constructed. We indicated that the problem of performance evaluation of timed PN can be transformed into a simple linear programming problem. In order to analyze the behavior of FMS and to attempt a reasonable solution for production planning, the timed marked graph (TMG) (Campos, Chiola, & Colom, 1992; Murata, 1989) also well-known Subclass of PN in literature is used. If the system can be modeled as a marked graph, the following formula can be applied for performance evaluation (Yamada & Kataoka, 1994):

$$C^{*} = \max_{\gamma} \{T_{\gamma}/N_{\gamma}\} \tag{1}$$

where γ indicates a cycle in the marked graph, $T\gamma$ and $N\gamma$, respectively, are the sum of firing time and the number of tokens along γ. Max in Eq. (1) is taken over all elementary cycles of the marked graph. It has been demonstrated that all transitions have the same minimal cycle time C^* (Ramchandani, 1973). In this study, to evaluate the performance of the system, we used linear programming method because of the computational complexity of Eq. (1). The problem of performance evaluation is reduced to LP problem with k (=m - n + 1) variables and n - 1 constraints, where m and n represent the number of places and transitions in the marked graph, respectively.

Basically, TMG is a set $M = \{P, T, I, O, \tau\}$, where

$P = \{p_1, p_2, \ldots, p_m\}$ is the finite set of places, m: number of places in M

$T = \{t_1, t_2, \ldots, t_n\}$ is the finite set of transitions, n: number of transitions in M

I: P→T is the input function,

O: P→T is the output function, and $\tau \in R^n$ is the vector representing the firing time at each transition.

5.3. Problem formulation

5.3.1. The computational framework provided by LP of marked graphs

A TMG *M* may also be viewed as a directed graph G = (V, E), where V = {v_1, v_2, . . . , v_n} is the set of vertices, and E = {e_1,e_2, . . . , e_m} $\subseteq$ V x V is the set of directed edges. It is denoted $e_i = (v_p,v_q)$
if and only if $I(p_i) = t_p$ and $O(p_i) = t_q$.
The m x n incidence matrix is A = (a_{ij}), with

$$\alpha_{ij} = \begin{cases} 1 \ when\ \vartheta_j is\ the\ terminal\ node\ of\ e_i \\ -1\ when\ \vartheta_j is\ the\ original\ node\ of\ e_i \\ 0\ otherwise \end{cases}$$

In directed graphs, a circuit refers to a closed sequence of forward and backward edges. A subgraph *T* of *G* is said to be a spanning tree if *T* covers all nodes of *G* and does not include any circuits.

The set of edges other than those in *T* are denoted as T^c. This is mentioned as a cotree. The numbers of edges in *T* and T^c are, respectively, n - 1 and k = m - n + 1. Without loss of generality, it can be assume that the first k edges of E are in T^c. The k - m matrix Γ, consisting of these *k* row vectors, is referred to as the fundamental circuit matrix of *G*. By the preceding construction, Γ can be written in canonical form as $\Gamma = (I, \Gamma^*)$ where *I* is the k x k identity matrix and Γ^*denotes the last n - 1 columns of Γ. In graph theory, a cycle can be represented as a linear combination as a linear combination of γ_1, γ_2,. . . , γ_k. Let $\rho = (\rho_1, \rho_2, . . ., \rho_k)$ be the k-dimensional row vector of decision variables. Therefore,

$$\gamma = \sum_{i=1}^{k} \rho_i \gamma_i = \rho\Gamma \qquad (2)$$

For the cycles, $\rho\Gamma \geq 0$ must have been.

$\tau = (\tau_1, \tau_2, \ldots, \tau_m)$ is the constant vector with the ith element being the firing time at the input transition of the ith place of *M*; i.e. $\tau_i = \tau(I(p_i))$. Let $\tau^* = \Gamma\tau$, $\rho\tau^*$ (T_γ) is the sum of firing times along γ.

Let $\eta \in N^m$ be the initial marking of *M*, with its ith component representing the number of tokens in place $p_i \in P$, $\eta^* = \Gamma_\eta$ and $\rho\eta^*$ (N_γ) is the total number of tokens. Thus Eq. (1) can be rewritten as (Morioka & Yamada, 1991):

Maximize

$$C = \frac{\rho\tau^*}{\rho\eta^*} \qquad (3)$$

Morioka and Yamada (1991) transformed the above Eq. (3) into the following LP problem.

Maximize $C = \rho\tau^*$

subject to $\rho\Gamma^* \geq 0$

$$\begin{aligned} \rho\eta^* &\leq 1 \\ \rho &\geq 0 \end{aligned} \qquad (4)$$

Eq. (4) is a LP problem with *k* variables and *n* constraints.

5.3.2. Application

The PN flow given in Fig. 4.2 is converted to a Marked Graph and shown in Fig. 5.1. In order to avoid the complexity, numbers of places and transitions are reduced while generating the Marked Graph however the outlines of overall system structure are preserved.

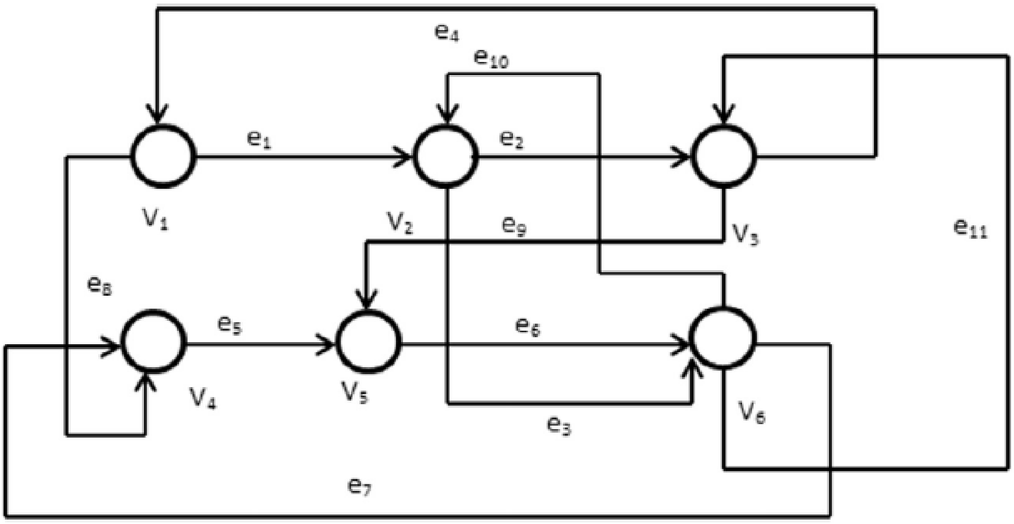


Figure 5.1: Marked graph for Valeo.

Consider the TMG of Fig. 5.1 the incidence matrix and the fundamental circuit matrix Γ, with vectors t^* and n^* for the marked graph of Valeo in Fig. 5.1, are shown as follows (see matrix (1) and matrix (2)): Performance evaluation for the marked graph of Fig. 5.1 can be executed by solving the following LP problem

	V_1	V_2	V_3	V_4	V_5	V_6
e_1	-1	1				
e_2		-1	1			
e_3		-1			1	
e_4	1		-1			
e_5				-1	1	
e_6					-1	1
e_7				1		-1
e_8	-1			1		
e_9			-1		1	
e_{10}		1				-1
e_{11}			1			-1

Matrix (1): The incidence matrix

	e_2	e_3	e_{10}	e_5	e_9	e_{11}	e_1	e_4	e_6	e_7	e_8	t^*	n^*
δ_1	1						1	-1				4	-1
δ_2		1					1			1	-1	8	3
δ_3			1				-1			-1	1	0	10
δ_4				1					1	1		15	18
δ_5					1			-1	1	-1	-1	6	-4
δ_6						1		1		-1	1	4	16

Matrix (2): Fundamental circuit matrix C, with vectors t^* and n^* for the marked graph of Valeo in Fig. 5.1

$$
\begin{aligned}
&\text{MaxC} = 4\rho_1 + 8\rho_2 + 15\rho_4 + 6\,\rho_5 + 4\rho_6 \\
&\rho_1 + \rho_2 - \rho_3 \geq 0 \\
&- \rho_1 - \rho_5 + \rho_6 \geq 0 \\
&\rho_4 + \rho_5 \geq 0 \\
&\rho_2 - \rho_3 + \rho_4 - \rho_5 - \rho_6 \geq 0 \\
&- \rho_2 + \rho_3 - \rho_5 + \rho_6 \geq 0 \\
&- \rho_1 + 3\rho_2 + 10\rho_3 + 18\rho_4 - 4\rho_5 + 16\rho_6 \leq 1 \\
&\rho_i \geq 0
\end{aligned} \tag{5}
$$

By solving Eq. (5), we obtain ρ = 1/18 (1, 1, 0, 0, 0, 1) with the maximal value of C^* = 0.89.

This solution gives the minimal cycle time C^* and also the three bottleneck cycles: $\{e_2, e_1, e_4\}$; $\{e_3, e_1, e_7, e_8\}$; $\{e_{11}, e_4, e_7, e_8\}$.

6. REASONS TO USE PETRI NETS

It is highly possible that PN can be used in different areas of the automotive industry as well as FAMSs. Workflow Management Systems (WFMS) are one of the potential to use this applications.

Reason 1: Formal semantics despite the graphical nature - The first reason for using a Petri-net-based WFMS, is the fact that business logic can be represented by a formal but also graphical language. The semantics of the classical Petri net and several enhancements (color, time, hierarchy) have been defined formally (Hee, 1994; Jensen, 1992; Murata, 1989; Reisig, 1985). In this section we will show that a Petri net can be used to model the primitives identified by the WFMC (1996). These primitives are also present in today's WFMSs. To discuss these workflow primitives we start by introducing some terminology. The objective of a WFMS is to handle cases successfully. Examples of cases are insurance claims, orders, mortgages, tax-returns and loans. A task is a piece of work whose execution contributes to the completion of a business process. Synonyms for task are process activity, logical step and work element. A task instance is a task that needs to be executed to handle a specific case. Task instances are executed by resources. A resource is a human, an application or a combination of a human and one or more applications. Synonyms for resource are actor or participant. The capabilities of a resource are given by a set of roles. Each task requires a specific role. Roles are used to map task instances to resources. A workflow procedure defines a partial ordering of tasks to handle cases of a specific type. A workflow process definition comprises a workflow procedure, a set of resources and a strategy to map task instances to resources.

One should clearly differentiate between workflow process definition and workflow process execution. Workflow process definition is concerned with the design of tasks, procedures, roles and resources using a design and analysis tool. Workflow process execution is concerned with the enactment of cases and task instances using a workflow engine.

Features of a Petri-net-based WFMS are most prominent in the design and analysis faze. Therefore, we concentrate on the workflow process definition. Figure 6 shows how the six workflow primitives identified by the WFMC (1996) can be mapped onto Petri nets. Tasks are mapped onto transitions and causal relations are modeled by places. Transition t1 models the synchronization of two sub flows (AND-join). Transitions t21 and t22 model an OR-join: two sub flows are merged into one sub flow. Transition t3 models an AND-split: a sub flow is split into two parallel sub flows. Transitions t41 and t42 model an OR-split: a selection is made between two alternative branches. Iteration can be modeled by adding a feedback transition (t52). Connecting two transitions (t61 and t62) by means of an intermediate place, results in two sequential tasks.

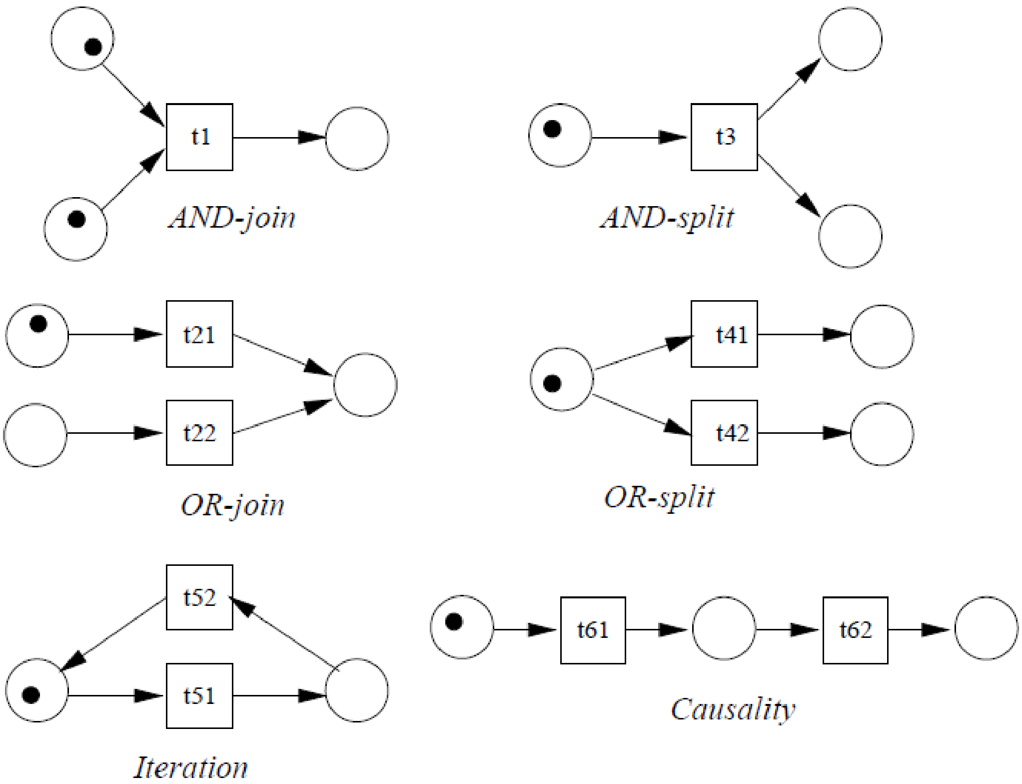


Figure 6.1: Workflow primitives

The state of a case *c* cis given by the distribution of tokens corresponding to *c* cover the places in the Petri net. To distinguish between tokens corresponding to different cases we use a high-level Petri net model (Jensen, 1992; Hee, 1994) extended with color. The color or value of a token contains information about the case the token belongs to and some additional information (e.g. routing parameters, due date, responsible or preferred resource). Note that each transition which models an AND-join requires a precondition to prevent tokens corresponding to different cases from being mixed.

The workflow primitives shown in Figure are used to define workflow procedures. However, to complete the definition of a workflow process we have to add another dimension: the dimension which takes care of the mapping of tasks to resources. The dimension of workflow procedures and the dimension of resource management are orthogonal and therefore difficult to visualize in one Petri net. Nevertheless, it is possible to model workflow procedures and resource management in an integrated way by using an high-level Petri net extended with color and hierarchy. For more information the reader is referred to Aalst, Van Hee &Houben (1995) and Aalst & Hee (1996, 1995).

Reason 2: State-based instead of event-based - In contrast with many other process modeling techniques, the state of a case can be modeled explicitly in a Petri net. Process modeling techniques ranging from informal techniques such as dataflow diagrams to formal techniques such as process algebra's are event-based, i.e., transitions are modeled explicitly and the states between subsequent transitions are modeled implicitly. Today's WFMSs are typically event-based, i.e., tasks are modeled explicitly and states between subsequent tasks are suppressed. Figure shows a typical diagram which defines a workflow procedure. The tasks A, B, C, D, E, F, G and H are represented explicitly in contrast to the state.

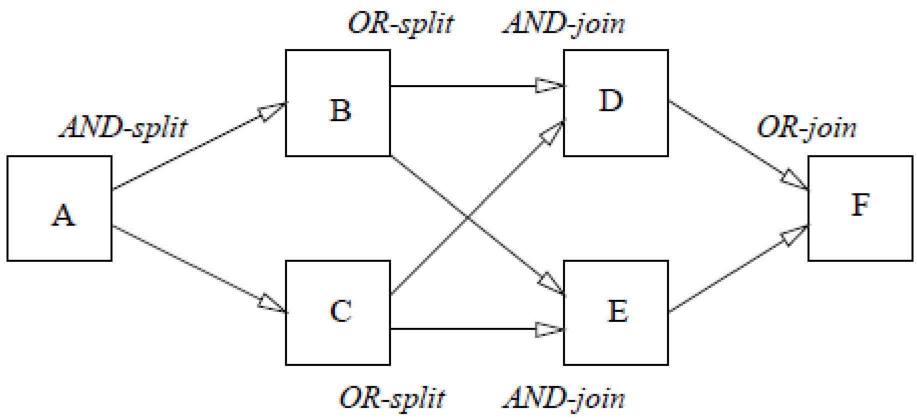


Figure 6.2: An event-based description of a workflow procedure

First of all, a state-based description allows for a clear distinction between the enabling of a task and the execution of a task. Since the enabling of a task does not imply that the task will be executed immediately, it is important to have this distinction. To illustrate this, we need to discuss the triggering of tasks in more detail. The execution of a task instance for a specific case starts the moment the task instance is triggered. A task instance can only be triggered if the corresponding case is in a state which enables the execution of the task.

Automatic: a task is triggered the moment it is enabled. This kind of triggering is used for tasks which are executed by an application which does not require human interaction.

7. CONCLUSION

Performance evaluation is an important issue for discrete event dynamic systems which can be described and modeled through the help of PN. In this research, we have shown that how PN models can be systematically built for FMSs in Automotive Industry and how they can be analyzed to gain insights into the qualitative weighted behavior of FMSs. The principles of PN can be used to model and analyze any given FMS. However, if the FMS is complex, the PN will be quite huge and analysis will not be easy. One way to overcome this problem is to use timed graphs instead of ordinary PNs.

In this thesis, we modeled the illustrative example as a TMG, a well-known subclass of PN in literature, and we showed that the problem of performance evaluation can be reduced to a simple LP problem with m - n + 1 variables and n constraints, where m and n represent the number of places and transitions in the marked graph, respectively.

The presented PN based method is illustrated by modeling a real-time scheduling and control for FAMS in Valeo Turkey.

We indicated that the problem of performance evaluation of timed PN can be transformed into a simple linear programming problem. In order to analyze the behavior of FMS and to attempt a reasonable solution for production planning, the TMG is used. The aim of the TMG method is to search for solutions to attain higher speeds and more flexibility and thus to increase manufacturing productivity. Marked graph has many benefits in modeling, qualitative analysis, performance evaluation and code generation through its design.

REFERENCES

Acree, E.S. and Smith, M.L. (1985). Simulation of a flexible manufacturing system-Application of computer operating system techniques, *Proc. 18th ZEEE Simulation Symp*., pp. 205-216.

Aized, T. (2009). Modelling and performance maximization of an integrated automated guided vehicle system using colored Petri net and response surface methods. *Computers & Industrial Engineering* 57(3): 822–831.

Al Hazza, M.H.F., Taha, A.H., Adesta, E.Y.T. and Albakri, A. (2015). Modelling and analysing deadlock in flexible manufacturing system using untimed Petri Net, in *4th International Conference on Advanced Computer Science Applications and Technologies*.

Alla, H., Ladet, P., Martinez, J. and Silva, M. (1985). Modelling and validation of complex systems by Petri nets: Application to FMS, Springer-Verlag New York.

Angraa, S., Chandab, A.K. and Chawlaa, V.K. (2018). Comparison and evaluation of job selection dispatching rules for integrated scheduling of multi-load automatic guided vehicles serving in variable sized flexible manufacturing system layouts: A simulation study, *Management Science Letters* 8(2018): 187-200.

Bashir, M., Zhou, J. and Muhammad, B.B. (2021). Optimal supervisory control for flexible manufacturing systems model with petri nets: A place-transition control, IEEE.

Beck, C.L. and Krogh, B.H. (1986). Models for simulation and discrete control of manufacturing systems. Proc. Znt. Conf. Robotics Automat.

Blackburn, I.D. (1991). Time-based competition the next battleground in American manufacturing. Chicago Business One Irwin.

Blaga, F., Pop, A., Indre, C., Hule, V. and Stănăşel, I. (2022). Evaluation of the performance of a robotized manufacturing cell through modeling and simulation with petri nets. *Academic Journal of Manufacturing Engineering* 20(4).

Chawla, V.K., Chanda, A.K., Angra, S. and Rani, S. (2019). Effect of nature-inspired algorithms and hybrid dispatching rules on the performance of automatic guided vehicles in the flexible manufacturing system, *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 41: 391.

Chen, J.H., Fu, L.C., Lin, M.H. and Huang, A.C. (2001). Petri-net and GA-based approach to modelling, scheduling, and performance evaluation for wafer fabrication, *IEEE Transactions on Robotics and Automation* 17(5): 619-636.

Chena, Y.F., Li, Z.W., Al-Ahmari, A., Wua, N.Q. and Qu, T. (2017). Deadlock recovery for flexible manufacturing systems modeled with Petri nets, *Informations Sciences* 381: 290-303.

Cherif, G., Leclercq, E. and Lefebvre, D. (2023). Scheduling of a class of partial routing FMS in uncertain environments with beam search. *Journal of Intelligent Manufacturing* 34: 493–514.

Cohan, E.G., Elphick, M.J. and Shoshani, A. (1971). System deadlocks, *ACMComput. Surveys* 3(2): 67-78.

Commaner, F. (1972). Deadlocks in Petri nets, Wokefield, Applied Data Research Inc.

Dennis, J.B. (1970). *Record protect mac conf concurrent systems and parallel computation*.

Duan, W., Zhong, C., Wang, X., Rehman, A.U., Umer,U. and Wu, N. *A deadlock prevention policy for flexible manufacturing systems modeled with petri nets using structural analysis*" IEEE.

Fundamentals of a theory of a synchrons us Information flow (1963). *Proc. IFIP, Congress* 62: 386-390.

Gamila, M.A. and Motavalli, S. (2003). A modeling technique for loading and scheduling problems in FMS. *Robotics and Computer-Integrated Manufacturing* 19(1–2): 45–54.

Habermann, A.N. (1977). System deadlocks, *Current Trends in Programming Methodology*, Vol. 11, K. M. Chandy and R. T. Yeh, (Eds.), Prentice-Hall Englewood Cliffs, NJ, pp. 256-297.

Hall, A.W., Saint, H., Shapiro, R. and Warshall, S. (1968). Final report of the ınformation systems theory protect, Tech. Rep RADC-TR-68-305, Griffiss Air Force Base New York.

Han, S., Chang, T.W., Hong, Y.S. and Park, J. (2020). Reconfiguration decision-making of IoT based reconfigurable manufacturing systems. *Applied Sciences* 10(14): 4807-4824.

Ho, Y.C. (1987). Performance evaluation and perturbation analysis of discrete event dynamic systems, *ZEEE Trans. Automat. Contr*. 32(7): 63-472.

Jain, M., Sandhya Maheshwari, K. P. and Baghel, S. (2008). Queueing network modelling of flexible manufacturing system using mean value analysis, *Applied Mathematical Modelling* 32(5): 700–711.

Javaid, M., Haleem, A., Singh, R.P. and Suman, R. (2022). Enabling flexible manufacturing system (FMS) through the applications of industry 4.0 technologies, *Internet of Things and Cyber-Physical Systems* 2: 49-62.

Kaid, H., El-Tamimi, A.M., Nasr, E.A. and Al-Ahmari, A. (2015). Applications of petri nets based models in manufacturing systems: A review, *Proceedings of the 2015 International Conference on Operations Excellence and Service Engineering Orlando*, Florida, USA.

Kamath, M. and Viswanadham, N. (1986). Applications of Petri net based models in the modelling and analysis of flexible manufacturing systems. *Proc. 1986ZEEE Conf. Robotics Automat*.

Kuo, C.H., Huang, H.P. and Yeh, M.C. (1998). Object-oriented approach of MCTPN for modeling flexible manufacturing system, *International Journal of Advanced Manufacturing Technology* 14: 737-749.

Lee, J. and Korbaa, Q. (2004). Modeling and scheduling of ratio-driven FMS using unfolding time Petri nets, *Computers & Industrial Engineering* 46(2004): 639–653.

Leonard, R. and Brown, C.C. (2015). Hierarchical control structure in flexible manufacturing systems, Macmillan Education UK.

Lin, J.T. and Lee, C.C. (1997). *Petri net-based integrated control and scheduling scheme for flexible manufacturing cells*, *Computer Integrated Manufacturing Systems* 10(2): 109-122.

Lu, Y., Chen, Y.F., Li, Z.W. and Wu, N.Q. (2022). An efficient method of deadlock detection and recovery for flexible manufacturing systems by resource flow graphs, *IEEE Transactions on Automation Science and Engineering* 19(3).

Manu, G. and Vijay Kumar, V. (2020). Optimization of system utilization and throughput in FMS using taguchi and ANOVA approach, Elsevier BV.

Martinez, J., Alla, H. and Silva, M. (1986). Petri nets for specification of FMS's, *Modelling and Design of FMS,* A. Kusiak, (Ed.), Elsevier New York, pp. 389-406.

Mejíaa, G. and Pereirab, J. (2020). Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets, *Journal of Manufacturing Systems* 54(2020): 272-284.

Murata, T. (1989). Petri nets: Properties, analysis and applications, *Proceedings of IEEE* 77(4): 541-580.

Narahari, Y. (1987). *Petri net-based techniques for modelling, analysis, and performance evaluation*, Doctoral dissertation, Dept. Comput. Sci. Automat., Indian Inst. Sci., Bangalore.

Narahari, Y. and Viswanadham, N. (1985). A Petri net approach to modelling and analysis of flexible manufacturing systems, *Annals Oper. Res*. 3: 449-472.

Parhi, S. and Srivastava, S.C. (2018). A highly efficient and optimised simulation based multi objective decision-making for FMS control, *International Journal of Emerging Research in Management and Technology* 6: 98.

Patalas-Maliszewska, J., Wisniewski, R., Topczak, M. and Wojnakowski, M. Design optimization of the Petri net-based production process supported by additive manufacturing technologies, *Bulletin of The Polish Academy of Sciences Technical Sciences* 70(2).

Peterson, J.L. (1981). Petri net theory and the modeling of systems, Englewood Cliffs, NJ Prentice Hall, Inc.

Peterson, J.L. and Silberschatz, A. (1985). *Operating system concepts,* Addison-Wesley Reading, MA.

Petri, C.A. (1962). *Kommunikation mit automaten.* Doctoral dissertation, Rheinisch-Westfalisches Institut für Instrumentelle Mathematik an der Universität Bonn, Bonn, 1962.

Popescu, M., Ungureanu-Anghel, D. and Filip I. (2013). Designing complex petri nets using submodels with application in flexible manufacturing systems, *8th IEEE International Symposium on Applied Computational Intelligence and Informatics*.

Ram, R. and Viswanadham, N. (2017). Performance evaluation of cellular flexible manufacturing systems: A decomposition approach, *IFAC Proceedings* 22: 37-44.

Reisig, W. (1985). *Petri nets: An introduction. EATCS Monographs on Theoretical Computer Science*. Springer-Verlag Berlin.

Rybicka, J. (2017). Modelling flexible manufacturing systems through discrete event simulation, Cranfield University.

Santarek, K., Buseif, I. M. (1998). Modeling and design of flexible manufacturing systems using SADT and Petri nets tools, *Journal of Materials Processing Technology* 76(1998): 212–218.

Saygin, C., Chen, F.F. and Singh, J. (2001). Real-time manipulation of alternative routings in flexible manufacturing systems: A simulation study, *International Journal of Advanced Manufacturing Technology* 18: 755-763.

Singha, R., Singh, R. and Khan, B.K. (2016). Meta-hierarchical-heuristic-mathematical-model of loading problems in flexible manufacturing system for development of an intelligent approach. *International Journal of Industrial Engineering Computations* 7(2016): 177-190.

Sivagnanavelu, D.G. (2000). *Dynamic scheduling of flexible manufacturing systems: A study of machine and material handling control strategies,* Master's thesis, Concordia University, Montreal.

Viswanadham, N. and Narahari, Y. (1987). *Coloured Petri net models for automated manufacturing systems, Proc. 1987 IEEE Znt. Conf. Robotics Automat*.

Wang, X., Xing, K., Feng, Y. and Wu, Y. (2021). Scheduling of flexible manufacturing systems subject to no-wait constraints via petri nets and heuristic search, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51(10).

Wang, Z., Pang, C.K. and Ng, T.S. (2019). Robust scheduling optimization for flexible manufacturing systems with replenishment under uncertain machine failure disruptions. *Control Engineering Practice* 92(2019): 104094.

Wiswanadham, N., Narahari, Y., Johnson, T.L. (1990). Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petri net models, *IEEE Transactions on Robotics and Automation* 6(6).

Yadav, A. and Jayswal, S.C. (2017). Modelling of flexible manufacturing system: A review. *International Journal of Production Research* 56(7): 2464–2487.

Zhanga, S., Li, X., Zhanga, B. and Wang, S. (2020). Multi-objective optimisation in flexible assembly job shop scheduling using a distributed ant colony system, *European Journal of Operational Research* 283(2020): 441-460.

Zhou, M.C. and DiCesare, F. (1993). Petri net synthesis for discrete event control of manufacturing systems, Kluwer Academic Publishers London.

Zhou, M.C. and Venkatesh, K. (1998). *Modeling, simulation and control of flexible manufacturing systems: A Petri net approach*, World Scientific Singapore.

Zhou, M.C. and Xiong, H.H. (2001). Petri net modeling and scheduling of automated manufacturing systems in computer aided design, engineering, and manufacturing: Systems techniques and applications, Vol. 4, *Optimization Methods for Manufacturing*, C.T. Leondes (Ed.), CRC Press, Taylor & Francis Group, LLC, pp. 1-23.

Zhou, M.C., DiCesare, F., and Desrochers, A.A. (1992). A hybrid methodology for synthesis of petri nets for manufacturing systems, *IEEE Transactions on Robotics and Automation* 8(3): 350-361.

APPENDICES

Appendix A. Informal Algorithm for the Real-Time Controller (Viswanadham, Narahari, Johnson, 1990)

```
Algorithm: Real-Time Controller
Input:   (1) Petri Net Model of the FMS. (2) n, the look-ahead.
         (3) P, the previous marking. (4) L1(P), L2(P). ..., L,(P).
Output:  Appropriate Scheduling Decision or Deadlock Recovery
         Action
Local Variables: i (integer); deadlock-flag (boolean);
begin
        if      P tangible then compute the current marking M after
                reading appropriate sensor outputs
        else    compute the current marking M by firing in P the
                transition that was selected to fire;
        Compute E,the set of enabled transitions in M;
        if      E =,Ø {M is a deadlock) then initiate deadlock recovery
        else if E contains only timed transitions JM is tangible) then
                initiate monitoring of activities in progress
        else    -{M vanishing} begin
                i:= 1;
                repeat
                        compute Li(M) using Li+, (P);
                        if      L.(M) contains only deadlocks then deadlock
                                flag:=true
                        else    i: =i+l
                until   (i =n+l or deadlock-flag);
                if      deadlock-flag then initiate appropriate advance
                        deadlock-recovery
                else begin      -{Ln(M) contains at least one safe state
                                or one blocked state)
                if      L (M) contains at least one safe statethen
                        select for firing an immediate
                        transition that leads to one of these
                        safe states
                else    select for firing an immediate transition
                        that leads to one of the blocked states

                end

        end

end
```

BIOGRAPHICAL SKETCH

Özkan BAŞAK has graduated from Yıldız Technical University as a Mechanical Engineer in 2006 and as an Electrical Engineer from the same university in 2007.

In 2007, he started the Industrial Engineering graduate program at Galatasaray University.

Between 2007 and 2010, he worked as a Credit Risk and Data Analysis Specialist at Fortis Bank Headquarter in Brussels.

In 2010, he decided to end his career in finance and opened a new chapter in the automotive industry.

Since 2010, he has been working at Valeo Group, he led diverse roles respectively as a Quality Engineer, Quality Manager, Project Manager, Lean Production Manager, Country Operational Excellence Director, Factory Manager and meanwhile Country Ethics & Compliance Champion, and he is currently managing the Sales operations.

He has been married since 2018 and has a 3 years old daughter.