

Flexible Manufacturing System Modeling: A Petrinet Based Survey Approach

Jacob John¹ and M B Kiran²

*¹Research Scholar, Dayananda Sagar College of Engineering,
Bangalore, Karnataka India 560078*

*²Pandit deendayal petroleum University,
Gandhinagar, Gujarat, India 382007*

Abstract

Flexible manufacturing system (FMS) is the one, which can be changed or adapted quickly to manufacture different components or products at different volumes of product. Flexible manufacturing system includes – a set of flexible machines, an automatic transport system and a sophisticated database with decision support system, to decide at each instant what is to be done on which machine. The objective of Flexible manufacturing system is to provide both machine flexibility and routing flexibility. In large flexible manufacturing system there are number of independent subsystems, and essential element of integration is through computer based system. Also, a detailed planning for successful coordination of complex subsystems highlights the importance of modeling of FMS. Petrinets are a set of nodes, arcs and well known for their modeling potential. These systems have significant ability to implement optimization techniques and are mainly used for modeling manufacturing plants as discrete event systems. In petrinet the two types of nodes are places and transitions, which represent the state of the system and the occurrence of events, respectively. In manufacturing systems, places would represent operations like process, transportation, reparation and transitions signifies events, such as termination of a job processing or a machine breakdown. Arcs are directed, and connect transitions with places. Tokens exist in places and represent the truth of the condition related with such a place. The firing process induces a token's flow among places. when a transition fires, tokens from all its input places are removed and put into the transition output places. A transition can only be fired if it has sufficient tokens at its input places. The main objective of this study is to carry out a detailed survey of Petrinet based modeling approaches to flexible manufacturing system.

Key words: Flexible manufacturing system, Modeling methodologies, Petri Net, Classifications.

INTRODUCTION

Performance in most manufacturing settings is affected by operative decisions related to job scheduling such as: a) selection of jobs in queue, b) priority when choosing a machine for a job processing (among parallel machines), and c) assignment of resources in the execution of a production plan. Such decisions have significant impact on systems efficiency, operational costs and service promise fulfillment, and are frequently taken intuitively by the operators, based on their experience.

In order to support this decision process and guarantee the system efficiency, computational applications need to be developed. Such applications should be able to

- (i) Provide modeling aids,
- (ii) Apply scheduling techniques and
- (iii) Define a production plan.

Petri Nets are well known for their modeling potential, and for their ability to implement optimization techniques. Karl Petri developed this technique in 1962 for communication system analysis. Its use has been extended to other application fields, like manufacturing [1].

In the industry, Petri Nets were mainly used for modeling manufacturing plants as discrete event systems. In the seventies the GRAFCET tool (Petri Net-based) was introduced in order to specify, validate and implement logic controllers in production systems. GRAFCET has been recognized in many countries [2]. Petri Nets have also been used in system design, and modeling, at companies like Microsoft Corporation, AT&T, Digital Equipment Corporation and Applied Materials Inc.[3].

When Petri Nets were introduced, many papers were: Topics included resource utilization, bottlenecks, throughput, cycle times and capacity estimations [4] [5]. In order to validate the use of Petri Nets, manufacturing systems were evaluated using different techniques like simulation, queuing theory, probability and stochastic Petri Nets [6]. On the other hand, the Petri Net potential for analyzing and modeling complex systems has encouraged its use on scheduling problems. A Beam Search algorithm was implemented on Petri Nets in order to find an optimal production schedule [7]. Later, the Branch and Bound method was used in robot task programming, truncating the net in smaller sub-nets. This technique was complemented with dispatching rules for selecting the firing transition [8] [9]. A heuristic that only generates part of the Petri Net reachability graph (the graph deploying all the net possible states) was suggested, presenting three searching types: depth search, bread search and a mixture of both [10].

SIMULATIONS WITH PETRI NETS

Petri Nets

A Petri Net is a set of nodes and arcs. There are two types of nodes: places and

transitions, which represent the state of the system and the occurrence of events, respectively. In manufacturing systems, places would represent operations (e.g., process, transportation, reparation), and transitions symbolize events, such as termination of a job processing or a machine breakdown. Arcs are directed, and connect places with transitions (or transitions with places). Tokens reside in places and represent the truth of the condition associated with such a place. The firing process induces a token's flow among places; when a transition fires, tokens from all its input places are removed and put into the transition output places. A transition can only be fired if it has been enabled (i.e. there are sufficient tokens at its input places). There are many different types of Petri Nets, though in this project only Marked Timed-Place Petri Nets (TPPNs) are of concern. Timed-Places are useful for modeling processing times, flow times or breakdowns, meaning that an amount of time may elapse until tokens enable their output transitions.

MODELING METHODOLOGIES - A SURVEY BASIC

Over the last two decades, several modeling methodologies have been developed. A coarse and non exhaustive description of several important methodologies that have been used in the manufacturing cell control domain is presented in the following sections. Modeling methodologies can be basically classified into two categories:

1. Machine-based methodologies,
2. Language-based methodologies.

Machine-based methodologies (MBMs) are based on the premise that a DES is an aggregation of processes that can be modeled entirely within a state-event framework. MBMs include automata, state charts, specification and description language (SDL), Petri nets, and discrete event control networks. MBMs provide a graphical notation that facilitates the representation of formal models.

Machine-Based Methodologies

Automata Theory

Automata theory provides a formal modeling methodology for manufacturing cells that employs abstract computing machines such as deterministic finite automata and pushdown automata. These abstract machines are inherently simple. In addition, the theory of regular languages and context-free languages can be used to describe them. Automata are graphically represented as state transition diagrams, where states are drawn as circles and transitions are represented by directed labeled arcs that connect states. The basis of automata theory is the concept of a deterministic finite automaton (DFA).

State charts

State charts comprise a formalism that extends the basic deterministic finite automata structure. A state chart consists of a set of states, which are graphically represented as

rectangular boxes, and a set of transitions between states, which are represented by directed arcs. One extension is that states are allowed to have substrates, which support the structuring of networks. Substrates are represented by the “insideness” of rectangular boxes. This is called depth. Another extension, similar to Mealy machines, is that transitions can be labeled by $(\alpha[Z]/\beta)$, where α is the input symbol that triggers the transition, Z is a condition that must be true in order to take the transition, and β is an action that is performed when the transition is made.

Discrete Event Control Networks

A discrete event control network (DECN) is a formal model of parallel computation. This formalism models systems with concurrency by capturing intended control logic, physical components, and behavioral characteristics in a network representation schema. In the network, there are three types of nodes that are connected by directed arcs: physical nodes, data nodes, and timer nodes. Messages are dispatched down the arcs as the data state of the nodes attains given values. These messages modify the data state of the receiving nodes, which in turn may dispatch messages to other nodes. Physical devices can be mapped to physical nodes. In a control situation, if the physical node is coupled to an input device, the node will receive its input from the physical device and dispatch messages to other nodes. Similarly, if the physical node is coupled to an output device, it will send a control message to the physical device based on its data state.

Language-Based Methodologies

Synchronous Languages

The family of languages is based on the synchrony hypothesis, which states that the formal model specification response to a set of inputs is instantaneous. The execution of these models is cyclic, where a cycle consists of fetching inputs and computing and emitting outputs. Each cycle is performed in one instant of time. The implication of the synchrony hypothesis, at the practical level, is that the computation and emission of the outputs must be completed before the next cycle occurs.

Process Algebras

The core of process algebra is the notion of a sequential process. A sequential process is formally defined as the four-tuple $P = (S, B, F, \text{ and } T)$, Where S is a finite set of states, B is the set of initial states, F is the set of final states, and T is a transition relation. The definition of a process is very general. It can be interpreted however, as a description of the behavior of a system of concurrent processes or programs.

CONCLUSIONS

In this paper, a survey has been presented of several important modeling methodologies that have been used in the manufacturing cell control domain. The

methodologies are classified as machine-based methodologies and language-based methodologies. The classification emphasize that the role of the manufacturing engineer is to model a physical system that meets the manufacturing cell requirements specification. Initially, the cell requirements specification is limited to flow charts or textual control specifications, the engineer must first proceed to create a formal model, a blueprint, according to which the manufacturing cell will actually be developed. Thus, the objective of the engineer is to build or create a physical system in such a way that possesses behavioral properties and qualities that are linked to one another in the same way as particular propositions in a formal system. The manufacturing cell embodied is a realization of the formal system, which constitutes the blueprint that can be implemented.

REFERENCES

- [1] Zhou M. C., DiCesare F., 1993, *Petri Net synthesis for discrete event control of manufacturing systems*, Kluwer Academic Publisher, (USA), 187 – 188.
- [2] Guia A., DiCesare F., 1993, *GRAFNET and Petri Nets in manufacturing systems*, *Intelligent Manufacturing: Programming Environments for CIM*, Springer Verlag, London, 153 – 176.
- [3] Zhou M. C., Venkatesh M., 1999, *Modeling, simulation and control of flexible manufacturing systems*, *Intelligent Control and Intelligent Automation*, Vol 6. World Scientific Publishing Co., Singapore, 43 – 44.
- [4] Zhou M. C., DiCesare F., 1991, Parallel and sequential mutual exclusions for Petri Net modeling for manufacturing systems, *IEEE Trans. on Robotics and Automation*, Vol 7, No. 4, 515 – 527.
- [5] Dubois D., Stecke K., 1983, *Using Petri Nets to represent production processes*, *Proc of the 22nd IEEE Conf. on Decision and Control*, San Antonio, TX, 1062 – 1067.
- [6] Watson J.F., Desrochers A. A., 1991, *Applying GSPN to manufacturing systems containing non-exponential transition functions*, *IEEE Trans. on System. Man. and Cyber.* Vol 21, No. 5, 1008 – 1017.
- [7] Shih J., Sekiguchi T., 1991, *A timed Petri Net and beam search based on-line FMS scheduling system with routing flexibility*, *Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation*. Sacramento, CA, 2548 – 2553.
- [8] Shen L., Chen Q., Luth J. Y., Zangh Z., 1992, Truncation of Petri Net models of scheduling problems for optimum solutions, *Proc. of Japan/USA Symposium on Flexible Automation*, 1681 – 1688.
- [9] Zhou M. C., Chiu H., Xiong H. H., 1995, *Petri Net scheduling of FMS using branch and bound method*, *Proc. of 1995 IEEE Int. Conf on Industrial Electronics, Control and Instrumentation*. Orlando, FL, 211 – 216. [10] Zhou M. C., Venkatesh M., 1999, *Modeling, simulation and control of flexible manufacturing systems*, *Intelligent Control and Intelligent Automation*, Vol 6. World Scientific Publishing Co., Singapore, 56.

- [10] Sun T. H., Cheng C. W., Fu L. C., 1994, *A Petri Net based approach to modeling and scheduling for an FMS and a case Study*, IEEE Trans. on Industrial Electronics, Vol 41, No. 6, 593 – 601.