B.V. Sudheer Kumar, C.S.P. Rao / IOSR Journal of Engineering (IOSRJEN)
www.iosrjen.org ISSN : 2250-3021
Vol. 2 Issue 2, Feb.2012, pp. 343-347

Modeling and optimization of process plans using Petri nets and INA

B.V. Sudheer Kumar[1], C.S.P. Rao[2]
[1]Research Scholar, Dept. of Mechanical Engineering, NIT Warangal, A.P, India-506021
[2]Professor, Dept. of Mechanical Engineering, NIT Warangal, A.P, India-506021

ABSTRACT

The present paper deals with the development and optimization of computer aided process plans with respect to time and cost. The list of features derived from the CAD model is converted into xml format. This xml file is given as input to the P3 software to generate the stochastic Petri net model automatically. By analyzing this model, all possible process plans can be obtained in the form of reachability graph. Then necessary data like machining time and cost of all operations will be calculated. This data is used to find out the optimal process plans in terms of cost and time by using INA software. The developed methodology has been successfully tested with a number of components.

Keywords: **CAD model, Computer aided process plans, INA software, Optimized process plans, stochastic petri nets.**

I.INTRODUCTION

Every manufactured product has some planning associated with it. In manufacturing systems, this is technically referred to as process planning. One of the first tasks of the manufacturing personnel when they receive new drawings is to perform the process plan. This task, when completed, generally directs both the organization of needed resources and the actual production of the product. The practice of process planning in manufacturing provides precise and clear sequential directions about how the product is to be routed and fabricated in a manufacturing facility. The first step in preparing a process plan is to secure a good drawing or drawings of the product. Because the drawings represent the initial ideas and plans for the product, In many organizations, CAD software is available and can be used to transform the ideas into dimensioned drawings with achievable specifications. The dimensioned drawings should contain important information including the following. complete and clear graphics, material types, part name, drawing number, owner name, date, units, appropriate set of views showing all required dimensions, tolerances with reasonable values for each dimension, clear titles and labels, and should be easy to read. In next step the drawings should be carefully studied the way one reads a manual, to understand all the details contained in them. It is important to separate the drawings into their parts at this juncture. Each part should have a clear label so that it can be identified. After separating the parts, the reader should try to answer such questions as: “How should each of these parts be processed?”; “What types of tools and machines will be needed to process each piece?”; “How many units of each part should be fabricated?”, and “How long should it take to process each piece?” The identified tasks or processes required to fabricate each part should be listed below it. This step requires a careful study of each part and determining the various manufacturing processes needed to fabricate it into the shape shown in the drawing. Successful completion of this step often requires a good knowledge of manufacturing processes and shop processing equipment.
The listed tasks are then sequenced so that they follow the order in which they will be performed. The sequencing is important because the listing simply listed the identified processes required to fabricate each part, but did not arrange them in the order or sequence in which they will take place during the fabrication process. Again, successful completion of this step will require a good knowledge of manufacturing and shop processing equipment. It is important to number the sequenced processes at this stage. The numbers indicate the sequence in which the processes will take place. Sometimes a part can have a flexible sequence of operations. When such a situation arises, the process planner should employ the sequence that will yield greater benefit to the person, company or customer.

Assign Time Data, Equipment, and Tooling to the Sequenced Processes: To complete the planning, it is necessary that the machines and tooling needed to process each part, as well as the time it takes to complete each process are assigned. Determining how much time a process takes to complete (also called machining time) is very important. The machining time along with all machines and tools must be clearly identified. The assigned time for each process should be a reasonably good estimate of how much time that process should take to complete. This process times data can be used for time study results and experts’ opinions. The estimated time should include times taken to retrieve tools, set up equipment and perform other related but not specified tasks. The total time is also calculated for each part.

Operations such as basic inspection and setup of needed machines are sequentially included to systematically show, when each process takes place. Process selection is influenced by several factors, including required quantity, materials that the parts are made of, surface finish requirements, as well as the specified tolerances. Selecting a process without considering the influence of these factors on it could adversely affect the cost, quality, and ease of manufacturing the parts.

II. TOOLS AND TECHNIQUES

2.1 Petri Nets: Petri nets were introduced in 1962 by Carl Adam Petri. Petri nets are a powerful modeling formalism in many engineering disciplines. Petri nets combine a well defined mathematical theory with a graphical representation of the dynamic behavior of systems. The theoretic aspect of Petri nets allow precise modeling and analysis of system behavior, while the graphical representation of Petri nets enable visualization of the modeled system and state changes. This combination is the main reason for the great success of Petri nets [1]. Consequently, Petri nets have been used to model various kinds of dynamic event-driven systems like computers networks, communication systems, manufacturing plants, command and control systems, real-time computing systems, logistic networks are some important examples. This wide spectrum of applications is accompanied by wide spectrum of different aspects which have been considered in the research on Petri nets. A Petri net is a particular kind of bipartite directed graphs populated by three types of objects. These objects are places, transitions, and directed arcs. Directed arcs connect places to transitions or transitions to places. In its simplest form, a Petri net can be represented by a transition together with an input place and an output place [2]. This elementary net may be used to represent various aspects of the modeled systems. For example, a transition and its input place and output place can be used to represent a data processing event, its input data and output data, respectively, in a data processing system. In order to study the dynamic behavior of a Petri net modeled system in terms of its states and state changes, each place may potentially hold either none or a positive number of tokens. Tokens are a primitive concept for Petri nets in addition to places and transitions. The presence or absence of a token in a place can indicate whether a condition associated with this place is true or false, for instance.

A Petri net is formally defined as a 5-tuple N = (P, T, I, O, M0), where

(1) P = {p1, p2, …, pm} is a finite set of places;

(2) T = {t1, t2, …, tn} is a finite set of transitions, PU T ≠ Φ, and P ∩ T = Φ ;

(3) I: P × T → N is an input function that defines directed arcs from places to transitions, where N is a set of nonnegative integers;

(4) O: T × P → N is an output function that defines directed arcs from transitions to places; and

(5) M0: P → N is the initial marking.

A marking in a Petri net is an assignment of tokens to the places of a Petri net. Tokens reside in the places of a Petri net. The number and position of tokens may change during the execution of a Petri net. The tokens are used to define the execution of a Petri net. Most theoretical work on Petri nets is based on the formal definition of Petri net structures. However, a graphical representation of a Petri net structure is much more useful for illustrating the concepts of Petri net theory. A Petri net graph is a Petri net structure as a bipartite directed multigraph [3]. Corresponding to the definition of Petri nets, a Petri net graph has two types of nodes. A circle represents a place; a bar or a box represents a transition. Directed arcs (arrows) connect places and transitions, with some arcs directed from places to transitions and other arcs directed from transitions to places. An arc directed from a place p_j to a transition t_i defines p_j to be an input place of t_i, denoted by $I(t_i, p_j) = 1$. An arc directed from a transition t_i to a place p_j defines p_j to be an output place of t_i, denoted by $O(t_i, p_j) = 1$. If $I(t_i, p_j) = k$ (or $O(t_i, p_j) = k$), then there exist k directed (parallel) arcs connecting place pj to transition t_i (or connecting transition t_i to place p_j). Usually, in the graphical representation, parallel arcs connecting a place (transition) to a transition (place) are represented by a single directed arc labeled with its multiplicity, or weight k [4]. A circle contains a dot represents a place contains a token.

The execution of a Petri net is controlled by the number and distribution of tokens in the Petri net. By changing distribution of tokens in places, which may reflect the occurrence of events or execution of operations, for instance, one can study the dynamic behavior of the modeled system. A Petri net executes by firing transitions. We now introduce the enabling rule and firing rule of a transition, which govern the flow of tokens:

(1) Enabling Rule: A transition t is said to be enabled if each input place p of t contains at least the number of tokens equal to the weight of the directed arc connecting p to t, i.e., $M(p) \geq I(t, p)$ for any p in P.

(2) Firing Rule: Only enabled transition can fire. The firing of an enabled transition t removes from each input place p the number of tokens equal to the weight of the directed arc connecting p to t. It also deposits in each output place p the number of tokens equal to the weight of the directed arc connecting t to p.

Mathematically, firing t at M yields a new marking

$M'(p) = M(p) - I(t, p) + O(t, p)$ for any p in P [5].

Notice that since only enabled transitions can fire, the number of tokens in each place always remains non-negative when a transition is fired. Firing transition can never try to remove a token that is not there.

A transition without any input place is called a source transition, and one without any output place is called a sink transition. Note that a source transition is unconditionally enabled, and that the firing of a sink transition consumes tokens, but doesn't produce tokens.

2.2 Integrated Net Analyzer

INA software is used to obtain the optimal process plans. Generally two different optimal process plans may exit. The first process plan is optimal with respect to manufacturing cost the second one with respect to manufacturing time

III. PROPOSED METHODOLOGY

This section explains the proposed method used to produce the automated optimal process plans. From the CAD model features are extracted and converted to xml format. This xml file can be used as input to the p3 software. The p3 software generates Petri net model which represents the manufacturing data. After analyzing the Petri net model reachability graph can be obtained. This graph gives the complete set of possible process plans [6]. The process plans, along with machining data and cost can be given as input to INA software. The INA software generates optimal process plans in terms of manufacturing time and cost [7]. More clear and piece wise explanation is given in the subsequent sections with the help of an industrial component used in milling machines.

IV. RESULTS & DISCUSSION

The proposed method has been tested with the help of several components. In the present paper, the procedure is explained with the help of one component.

Case study:

The **Figure 1** shows 3D model of the component used in the present paper. This Figure 1 contains two T-slots, three through holes, and a through slot. This information is obtained from the CAD model and converted to xml file, which is given as input to p3 software. In the present work the xml file is taken as input and the process continues. The **Table 1** shows the fragment of xml file.

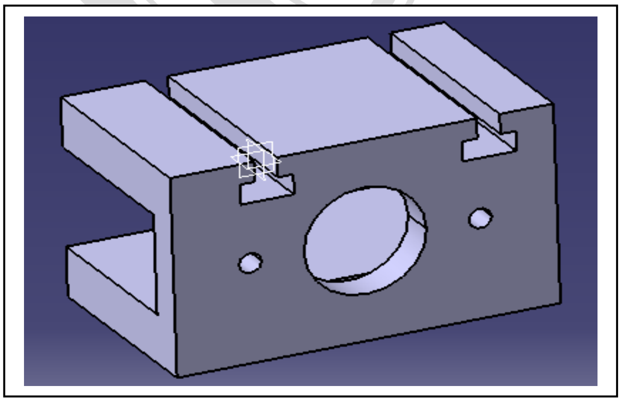
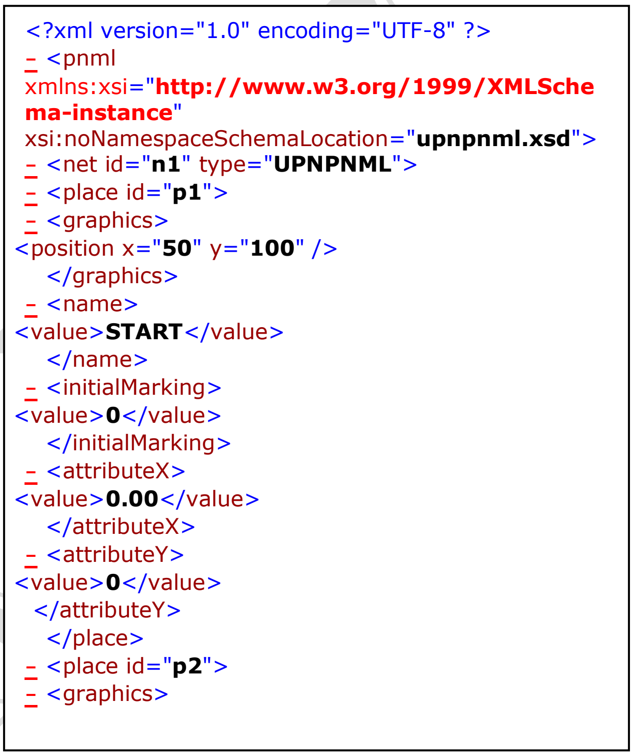


Figure 1: CAD Model

The xml file is given as input to p3 software which produces the Petri net model. The Petri net model shown in Figure 2 contains places and transitions connected by means of arcs. In this figure the places indicates resources (machine tools, cutters and setups) and transitions indicate machining operations. A token in a place shows that the resource is available. The Petri net model can be simulated and the reachability graph was analyzed for all possible process plans.

Table 1: table showing the fragment of xml file



```
 <?xml version="1.0" encoding="UTF-8" ?>
 - <pnml
 xmlns:xsi="http://www.w3.org/1999/XMLSche
 ma-instance"
 xsi:noNamespaceSchemaLocation="upnpnml.xsd">
 - <net id="n1" type="UPNPNML">
 - <place id="p1">
 - <graphics>
<position x="50" y="100" />
   </graphics>
 - <name>
<value>START</value>
   </name>
 - <initialMarking>
<value>0</value>
   </initialMarking>
 - <attributeX>
<value>0.00</value>
   </attributeX>
 - <attributeY>
<value>0</value>
  </attributeY>
   </place>
 - <place id="p2">
 - <graphics>
```

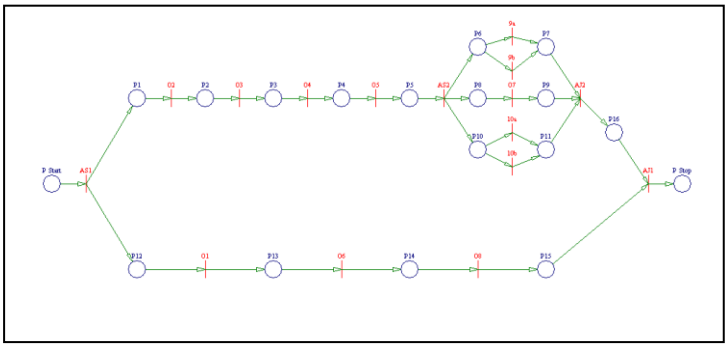


Figure 2: Petri Net model

The reachability graph generated from p3 software indicates all possible process plans.

The table indicating the required operations to produce the part with alternatives is given in the Appendix-A.

Assumptions for calculating the cost and time of operations are as follows.

Machining cost on vertical milling machine [M1] is Rs.25.00 per minute

Machining cost on radial drilling machine [M2] is Rs.15.00 per minute

Machine changing cost either from M1 to M2 or M2 to M1 is same and the cost is Rs. 1000.00

Cost of cutter change on both the machines are same and is Rs. 250.00

Cost of set up change on both the machines are same and is Rs. 50.00

Initial set up cost, cutter cost on both the machines is considered as Rs.0/-

Time required to change set up on both machines is 1 minute

Time required to change cutters on both machines is 3 minutes

Time required to change the work piece from one machine to other is 10 minutes

These process plans along with machining data and cost are given as input to INA software. The INA software generates optimal process plans in terms of cost and time (refer to Table 3).

Table 3: Table showing optimal process plan based on cost

Optimal sequence of operations based on minimum cost	Cost in Rs.
O1	150
O6	200
O8	400
O3	450
O5	200
O10a	50
O9a	50
O7	1050
O2	390
O4	140
Minimum cost of production	3080

In this example the process plan with minimum cost is found to be Rs. 3080/-

Table 4: Table showing optimal process plan based on time

Optimal sequence of operations based on minimum time	Time in minutes
O1	6
O6	7
O3	7
O5	7
O2	4.6
O4	4.6
O8	10
O10a	6
O9a	5
O7	34
Minimum time of production	91.2

In this example the process plan with minimum time is found to be 91.2 minutes.

CONCLUSIONS

In the present paper an attempt has been made to establish a procedure to obtain optimal process plans from the 3D CAD model. Two softwares, namely p3 and INA have been used to perform the said task. The established method has been tested successfully for several components.

REFERENCES

[1] Jiacun Wang, Petri Nets for Dynamic Event-Driven System Modeling,

[2] Dimitris Kiritsis, Michel Porchet.A generic Petri net model for dynamic process planning and sequence optimization, Advances in Engineering software 25(1996), 61-71

[3] Ajmone Marsan, M. 1990. Stochastic Petri nets: an elementary introduction. *Advances in Petri Nets, LNCS* 424.

[4] Genrich, J.H., and K. Lautenbach, 1981. System modeling with high-level Petri nets. *Theoretical Computer Science* 13: 109-136.

[5] Haas, P. Stochastic Petri Nets: Modelling, Stability, Simulation. 2002. New York: Springer-Verlag.

[6] Jensen, K. 1981. Colored Petri nets and the invariant-method, *Theoretical Computer Science* 14: 317-336.

[7] Jensen, K., 1997. *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use (3 volumes). London:* Springer-Verlag.

[8] Lin, Chuang, Liqin Tian and Yaya Wei. 2002. Performance equivalent analysis of workflow systems, *Journal of Software* 13(8): 1472-1480.

[9] Lindemann, C. 1998. Performance Modelling with Deterministic and Stochastic Petri Nets. John Wiley and Sons.

[10] Mandrioli, D., A. Morzenti, M. Pezze, P. Pietro S. and S. Silva. 1996. A Petri net and logic approach to the specification and verification of real time systems. In: *Formal Methods for Real Time Computing* (C. Heitmeyer and D. Mandrioli eds), John Wiley & Sons Ltd.

[11] Molloy, M. 1982. Performance analysis using stochastic Petri nets. IEEE Transactions on Computers 31(9): 913-917.

[12] Murata, T. 1989. Petri nets: properties, analysis and applications. *Proceedings of the IEEE* 77(4): 541-580.

[13] Natkin, S. 1980. Les Reseaux de Petri Stochastiques et Leur Application a I'evaluation des Systemes Informatiques. These de Docteur Ingegneur, Cnam, Paris, France.

[14] Peterson, J. L. 1981. Petri Net Theory and the Modeling of Systems. N.J.: Prentice-Hall.

[15] Ramamoorthy, C., and G. Ho. 1980. Performance evaluation of asynchronous concurrent systems using Petri nets. *IEEE Transaction on Software Engineering* 6(5): 440-449.

[16] Tsai, J., S. Yang, and Y. Chang. 1995. Timing constraint Petri nets and their application to schedulability analysis of real-time system specifications. *IEEE Transactions on Software Engineering* 21(1): 32-49.

[17] Wang, J. 1998. *Timed Petri Nets, Theory and Application.* Boston: Kluwer Academic Publishers.

[18] Wang, J. 2006. Charging information collection modeling and analysis of GPRS networks. *IEEE Transactions on Systems, Man and Cybernetics, Part C* 36(6).

[19] Venkatesh, K., M. C. Zhou, and R. Caudill. 1994. Comparing ladder logic diagrams and Petri nets for sequence controller design through a discrete manufacturing system. *IEEE Trans. on Industrial Electronics* 41(6): 611-619.

APPENDIX-A

Operation	Set up	Machine tool	Cutter	Machining time in min
Bottom face machining [O1]	S1	M1	Flat End Mill [T1]	6
Side 0° machining [O2]	S2	M1	Flat End Mill [T1]	3.6
Side 90° machining [O3]	S3	M1	Flat End Mill [T1]	6
Side 180° machining [O4]	S4	M1	Flat End Mill [T1]	3.6
Side 270° machining [O5]	S5	M1	Flat End Mill [T1]	6
Top face machining [O6]	S6	M1	Flat End Mill [T1]	6
Through slot [O7]	S3	M1	Slot mill [T2]	30
2 xT-slots machining [O8]	S6	M1	T-Slot cutter [T3]	6
Drilling Φ32 hole [O9a]	S5	M1	Drill Φ32 [T4]	2
Drilling Φ32 hole [O9b]	S1	M2	Drill Φ32 [T4]	1
Drilling Φ6x2 holes [O10a]	S5	M1	Drill Φ6 [T5]	2
Drilling Φ6x2 holes [O10b]	S1	M2	Drill Φ6 [T5]	1