A comparative study on Petri Nets in manufacturing applications

[1] Sayed Taha Mohamed, [2] Mohamed Abdel Gawad Mostafa, [3] Ahmed Fathi Mohamed

[1]Production Engineering and Mechanical Design Dept., Faculty of Engineering, Minia University, Egypt
[2] Industrial Engineering Dept., faculty of Engineering, Zagazig University, Egypt
[3] Mechanical Engineering Dept., College of Engineering and Islamic Architecture, Umm Al-Qura University, KSA.
Afmohamed277@gmail.com

Abstract: The main aim of this paper is to show the Petri Nets facilities as a comprehensive approach to manufacturing applications for design, specification, simulation and verification of systems. So, a comparison study between the traditional Petri Net (PN) and Colored Petri Net (CPN) as a graphical and mathematical modeling tool was considered for systems that exhibit properties such as sequencing, concurrency merging and synchronization. We can observe that, by using CPN the description and analysis is more compact and the data manipulations can be described in much more direct way. In tern, CPN is a very useful tool for manufacturing systems of large size and high complexity. Petri net is involved to make the implementation of the job shop scheduling specially in any automation system. the effectiveness and efficiency of the proposed approach is illustrated in case study.

[Sayed Taha Mohamed, Mohamed Abdel Gawad Mostafa, Ahmed Fathi Mohamed. **A comparative study on Petri Nets in manufacturing applications**. *Life Sci J* 2013;10(1):1496-1502] (ISSN:1097-8135). http://www.lifesciencesite.com. 20

Key words: Manufacturing modelling; Petri net; Colored Petri net.

1. Introduction

Petri Nets were devised in Germany 1962 by **Petri**, as a tool for modeling and analyzing processes assistance in the field of automata [1]. One of the strengths of this tool is the fact that it enables processes to be described graphically. Despite the fact that Petri Nets are graphical, they have a strong mathematical basis **[2].** Unlike many other schematic techniques, they are entirely formalized **[3].** So, it is often possible to make strong statements about the properties of the process being modeled. There are also several analysis techniques and tools available which can be applied to analyze a given Petri Net **[4].**

In tern, Petri Nets were recognized as an appropriate tool for modeling and analysis of manufacturing systems that exhibited properties such as sequencing, conflict, concurrency and synchronization **[5, 6, and 7].** Over the years, the model proposed by Carl Adam Petri has been expanded upon in many different ways. So, it is possible to model complex processes in an accessible way. Since the late of 1970s, the European have been very active in organizing workshops, advanced courses and publishing conference proceedings on Petri Net. From July 1985 to June 1997 a global interest were focused in the advancements of timed and stochastic Petri Nets and their applications in the design and performance evaluation of systems **[8].** However, in modeling a practical system using PN, one of the problems normally encountered is the rapid growth in the net size. So several PN classes have been proposed to reduce the net complexity, e.g. the introduction of object oriented PNs (OOPNs) **[9],** colored PNs (CPNs) **[10, 11 and 12]** and hierarchical PNs **[6].** Several authors have extended the traditional PN model with color in other words typed tokens **[10]** in these models; tokens have a value often referred to as color. There are several reasons for such an extension. One of these reasons is the fact that uncolored PN tend to become too large to handle. Another reason is the fact that tokens often represent objects or resources in the modeled system. As such these objects may have attributes, which are not easily represented by a simple PN token. These CPNs allow the modeler to make much more succinct and manageable descriptions.

The purpose of this paper is to present a comparison between traditional PN and CPN that is the field which can reduce the complexity of the management problems in manufacturing systems. The second section of the paper is devoted to the definitions, concepts and properties to be used in traditional PN. CPN definitions, concepts and properties are presented in the third section. The fourth section introduces a case study to compare the two categories of Petri Nets. Finally, section five presents the concluding remarks.

2. Traditional Petri net (PN)

A Petri net consists of places and transitions. We indicate a place using a circle. A transition is shown as a rectangle or bar. Figure 1 shows a simple Petri net, consisting of three places (P_1, P_2, and P_3) and three transitions (T_1, T_2, and T_3). Places and transitions in a Petri net can be linked by means of directed arcs. These arcs maybe weighted, this weight represents the number of tokens which can be move through these arcs. The arc is labeled by its weight; arc without label has a weight equal one. In figure 1-a, for example, the place p_1 and the transition t_1 are linked by an arrow (arc with weight equal to one) pointing from the former to the latter.

There are two types of arcs; those which run from a place to a transition called input arcs and those which run from a transition to a place called output arcs. Arcs from a place to a place or a transition to a transition are not possible. In tern upon the arcs, we can determine the input and the output places of a transition.

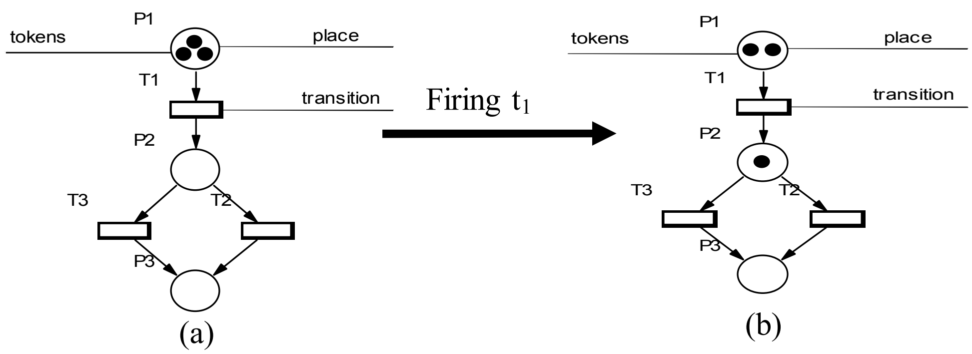


Figure 1: A traditional Petri net

Places can contain tokens. These are indicated using black dots. In figure 1 the place p_1 contains three tokens. The structure of a Petri net is fixed; however, the distribution of its tokens among the places can change. The transition t1 can thus take tokens from the P_1 (input place) and put them in P_2 (output place). We call this the firing of the transition.

3. Qualitative properties of Petri nets

3.1 Boundedness

Marking M_o is K-bounded if there exists a positive integer K, such that for every reachable marking M the number of tokens in each place is bounded by K. If K=1, the marking is said to be safe. In a manufacturing environment, the boundedness or safeness of a Petri net indicates the absence of overflows in the modeled system.

3.2 Liveness

A Petri net is live given initial marking Mo if there always exist a firing sequence σ to enable each transition in the net for any marking in $R(M_o)$ (set of all markings reach from the initial marking M_o). A transition that can not fire is redundant transition and can be eliminated from the net. Liveness is tied to the concept of deadlocks and deadlock-freeness, as related to the modeling of production system.

3.3 Reversibility

A Petri net is reversible if for every $M \in R(M_o)$ then $M_o \in R(M)$ i.e. the initial marking is reachable from all reachable markings. Reversibility means reinitializability. It implies that the system will finally return to its initial state from any current state.

3.4 Conservative

A Petri net is conservative if the number of token in the net is constant. This implies that each transition in such a net is conservative, in the sense that the number of inputs of each firable transition is equal to the number of outputs of that transition. More generally, weights can be defined to each place allowing the number of tokens to change as long as the weighted sum is constant. This is an important concept, since, if tokens are to represent resources, then it follows that since resources can neither be created nor destroyed, tokens should also be neither created nor destroyed

4. Analysis of Petri nets

Once a system has been modeled by using a Petri net, it is desirable to analyze the net to determine which properties the net possesses. Two analysis approaches will be discussed later. The success of any model depends on the next two factors: its modeling power and its decision power. Modeling power refers to the ability to correctly represent the system to be modeled; decision power refers to the ability to analyze the model and determine properties of the modeled system **[13].** The modeling power of Petri net has been examined in the previous sections, and in this section we take into consideration the analysis techniques of PNs.

4.1 Reachability graph (RG)

The Reachability graph represents all of the possible reachable markings. Starting with the initial marking M_o. If we able to compute all reachable markings, $M \in RG(N, M_o)$, and their Reachability relationships, all qualitative behavioral properties should be analyzable **[14].** A major problem arises in systems is unbounded system (infinite states), because it can not easily be represented by enumeration, so finite representations have been proposed. The main limitation of this approach is its computational complexity, so called state explosion problem; the number of markings can be exponential with respect to the size of the net (measured, for example, by the number of places). Once the Reachability graph for a Petri net is obtained, several analyses can be performed.

Boundedness: A PN is K-bounded iff the symbol ω (ω implies a loss of information about the actual number of tokens involved) never appears in its Reachability graph. The upper bound K is determined by searching all the nodes for the largest number of tokens. The net is safe if K =1.

Conservativeness: A PN is conservative iff it is bounded and the weighted sum of the tokens in every node of the Reachability graph is constant.

Reversibility: A PN is reversible with respect to an initial marking M_o, iff every node in the Reachability graph is in a directed circuit containing M_o. A PN is partially reversible if a directed circuit containing Mo includes only some of the nodes.

Liveness: A PN is live with respect to an initial marking M_o, if it is reversible and the Reachability graph has a directed circuit, not necessarily elementary circuit containing all the transitions infinitely often. This is sufficient but not necessary condition for Liveness. There are PNs that are live but not reversible.

4.2 Incidence Matrix

In this section we present another technique for analyzing Petri nets based on matrix linear-algebra. This method has a major advantage over the Reachability graph.

P-invariant: An m x 1 vector x is a P-invariant if $x^tA = 0$ Where A is the incidence matrix [7, 12] and m denotes the number of places. In other words, $x^t M = x^t M_o$. This implies the weighted sum of the markings is constant if P-invariants are chosen as weights.

T-invariant: An n x 1 vector y is a T-invariant if $Ay = 0$ Where A is the incidence matrix and n denotes the number of transitions. Thus if a firing vector equal to a T-invariant in the state equation it returns the marking to itself.

By using incidence matrix we can perform several analyses for a PN as follows:

Boundedness: A PN is structurally bounded if there is exists a nonzero (n x 1) vector x for non-negative integers such that $x^tA \leq 0$.

Conservativeness: A PN is conservative iff there exists a nonzero (n x 1) vector x of non-negative integers such that $x^tA = 0$.

Reversibility: A PN is not reversible if $Ay = 0$ gives only trivial solutions. This is a sufficient but not necessary condition. So, a non-trivial solution only guarantees partial reversibility.

Liveness: A PN is live if all places are covered by P-invariants, all the P-invariants are marked with tokens, and none of the siphons (a siphon P_s is a set of places, such that ${}^{\bullet}P_s \subseteq P^{\bullet}{}_s$. it is mean every transition that outputs to one of places in P_s also inputs from one of places in P_s. so, a siphon having lost all of its tokens can never obtain a token again.) is ever cleared of tokens **[15].**

5. Colored Petri Nets (CPNs)

CPNs were originally introduced by Kurt Jensen in his Ph.D. thesis that was published in 1980. CPN can be considered as a graphical oriented language for design, specification, simulation and verification of systems. It is in particular well suited for systems in which communication, synchronization and resource sharing are important. Typical examples of application areas are communication protocols, distributed systems, imbedded systems, automated manufacturing and work flow analysis. In the beginning, only small, unstructured set of colors were used such as enumerating a fixed set of processes. Later it was generalize, in such a way arbitrary complex data types can be used as color sets. Now it is not at all atypical to have tokens that carry a complex data value, e.g., a list of many thousand records, involving fields of many different types **[2, 10 and 11].** In addition formal expressions are attached to the arcs of the Petri Net model in order to constrain and specify the flow of tokens through the transition firing process. It is possible to attach a Boolean expression to each transition. The Boolean expression is called a guard. It specifies that we only accept bindings for which the Boolean expression evaluates to true. Figure 2 shows the elements of CPN.

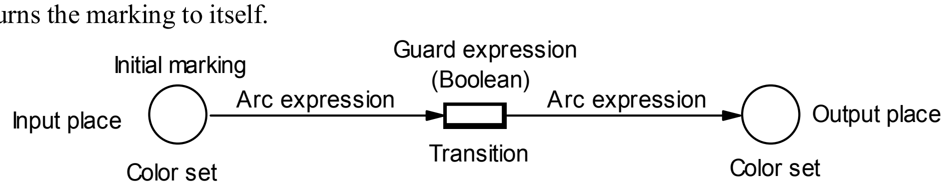


Figure 2: The elements of Colored Petri Nets

To enabled, a transition must have sufficient tokens on its input places, and these tokens must have token values that match the arc expressions and binding any variable to a value in its type. For example, figure 3 shows a simple CPN contains three places p_1 can contain tokens of data type (Integer, String) and it has two tokens one of (1, A) and one of (2, B) as initial marking. Numbers in front of the small backslash indicate the exactly number of token found in the place. p_2 can contain tokens of data type (Integer) and it has one token of (1) as initial marking. p3 can contain tokens of data type (string) and it has no token. Transition t has three surrounding arcs and three arcs expressions involve the variables n of type integer and p of type string. For the transition to fire, the two variables must be bind to values in their types in such a way that the arc expression of each incoming arc evaluates to a token value that is present on the corresponding input place. Since p_2 contains only one token with value 1, it is obvious that n must bound to 1. Then we see that p must be bound to A, since p_1 only has one token in which the first element of the pair is 1. With binding n=1 and p=A transition t is enabled, because there is token 1 on place p_2 and a token (1, A) on place p_1. When transition fires it removes the two specified tokens from the input places, and simultaneously it produce a token of A on place P_3. We could add a guard for example n<10 to the transition t this would prevent firing execution more than 9 times.

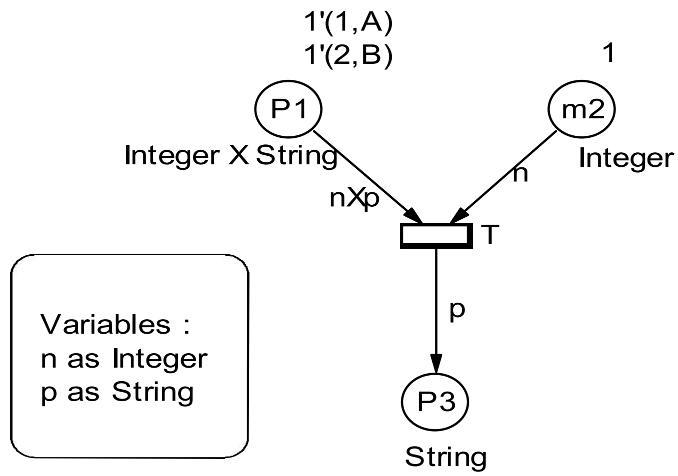


Figure 3: simple CPN

The information attached to the tokens is usually altered and modified when a transition is fired. All these features enable CPNs to combine and group several similar subnets into a single net. Thus CPNs can be used to construct more compact and concise models than traditional Petri Nets. CPN is a modeling language at the same time theoretically well found and versatile enough to be used in practice for systems of the size and complexity we find in typical industrial projects. To achieve this, CPN combined with a programming language. CPN provide the primitives for the description of the sequencing and synchronization of concurrent processes, while programming language provide the primitives for the definition of data types and manipulation of data values.

6. Case study

In this section it is proposed to use classical Petri Net and colored Petri net during the modeling stage. Considering the information given in Table 1, which is concerned with a manufacturing system, that is composed of three machines (M_1, M_2, and M_3) **[16, 17, 18, 19 and 20]**. Correspondingly, the process planes for two different products J_1 and J_2 are given in table 1.

Table 1: The model data

	Jobs	
Operation No.	J_1	J_2
1	M_1 / M_2	M_1 / M_3
2	M_2 / M_3	M_1 / M_2 / M_3

The preceding model can be depicted using classical Petri Net as shown in Figure 4 and the corresponding description of this model shown in table2.

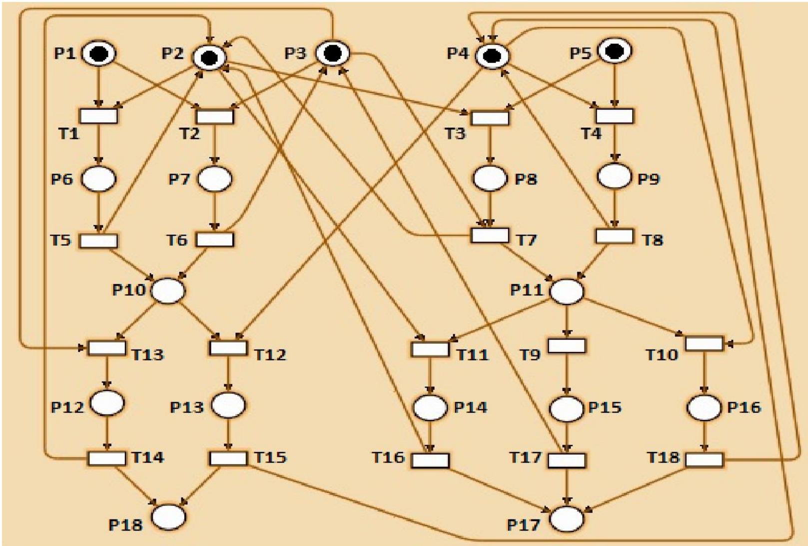
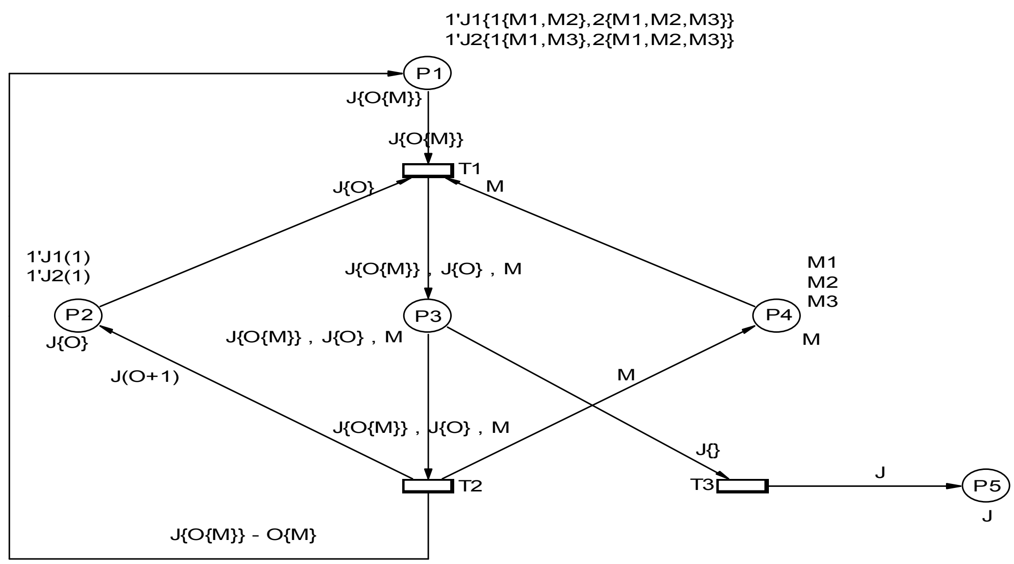


Figure 4: The classical Petri Net construction for the preceding industrial model

Table 2: Descriptions for places and transitions showing in Figure 4

Places	Description of places	Transitions	Description of transitions
P_1	Raw material of job one available	T_1	Start of machine one operation for job one process one
P_2	Machine one available	T_2	Start of machine two operation for job one process one
P_3	Machine two available	T_3	Start of machine one operation for job two process one
P_4	Machine three available	T_4	Start of machine three operation for job two process one
P_5	Raw material of job two available	T_5	End of machine one operation for job one process one
P_6	Machine one operating for Job one process one	T_6	End of machine two operation for job one process one
P_7	Machine two operating for Job one process one	T_7	End of machine one operation for job two process one
P_8	Machine one operating for Job two process one	T_8	End of machine three operation for job two process one
P_9	Machine three operating for Job two process one	T_9	Start of machine two operation for job one process two
P_{10}	Intermediate place for job one between process one and process two	T_{10}	Start of machine three operation for job one process two
P_{11}	Intermediate place for job two between process one and process two	T_{11}	Start of machine one operation for job two process two
P_{12}	Machine two operating for Job one process two	T_{12}	Start of machine two operation for job two process two
P_{13}	Machine three operating for Job one process two	T_{13}	Start of machine two operation for job one process two
P_{14}	Machine one operating for Job two process two	T_{14}	End of machine two operation for job one process two
P_{15}	Machine two operating for Job two process two	T_{15}	End of machine three operation for job one process two
P_{16}	Machine three operating for Job two process two	T_{16}	End of machine one operation for job two process two
P_{17}	Final product for job one	T_{17}	End of machine two operation for job two process two
P_{18}	Final product for job two	T_{18}	End of machine three operation for job two process two

In this situation it is proposed to use CPN during the modeling stage considering the information given in table 1, which is modeled using classical Petri Net earlier in this section. This model can be depicted using CPN as shown if Figure 5 and the corresponding description of this model shown in table 3.



Variables and Types:
- M as Machine
- O as Operation (Process)
- J as Job
- J {O {M}} as job J with a list of its processes O and each process with a list of convenient machines M
- J {O} as a Process O of job J
- J {} as job J achieved all of its processes
- Operator (-) such as J {O {M}} - O {M} mean that remove the set of process O from the multi set of Processes of job J

Figure 5: The CPN construction for the industrial model shown in table 2

Table 3: Descriptions for places and transitions showing in Figure 5

Places	Description of places	Transitions	Description of transitions
P_1	Raw material of available jobs with a list of the operation make on the job	T_1	Start of machining operation for process O on machine M of job J
P_2	Counter for each job operations	T_2	End of machining operation
P_3	Machining operation	T_3	Taking the final product to the output buffer
P_4	The available machines		
P_5	The output buffer		

7. Comparison between PN and CPN

The main difference between CPNs and classical Petri Nets is that in CPNs places describe the state or resources of the system and depicted as a circle has a data type and data values which called colors. These colors represent the tokens. In tern tokens are identifiable and carry information or data values called colored tokens, hence it can be distinguished from each other. Previous industrial model shown in Figure 5 at which transition t_1 handle all starting operations on all available machines for all jobs, in contrast with classical PN that constructed in Figure 4, a particular Job, particular operation and particular machine has an individual transition.

Attaching a data value to each CPN token allows us to use much fewer places than needed in a classical PN. In CPN we can attach data values to the individual tokens. In classical PN the only way we can distinguish between tokens is by positioning them at different places. When a CPN uses complex types (such as integers, reals, strings, products, records and lists), the equivalent classical PN often has infinite or astronomical number of places.

The use of variables in arc expression means that each CPN transition can occur with different bindings, i.e., in many slightly different ways in a similar way as a procedure can be executed with different parameters. Hence, in CPN we can use a single transition to describe a class of related activities, while in classical PN we need a transition for each instance of such an activity.

Finally, we must stress on the fact that CPNs have the same modeling powers of ordinary Petri Nets. As mentioned earlier, the main advantage of

CPNs is that their formalism assists in constructing more compact Petri Net models which in turn help in simplifying the analysis process.

8. Conclusion

Petri nets are a general graphical tool very well suited to the description of distributed, sequencing and concurrent systems which exhibit synchronization and contention for share resources. In tern Petri nets have been claimed to be an ideal modeling tool for manufacturing systems. We can observe from the previous discussion that the main problem that hinders the use of classical Petri Net, as modeling tool, to model complex systems is usually a tedious task due to the large size of the resulting Petri Net model. Moreover, the huge size of the final Petri Net model increases the difficulty of analyzing and validating the model. Therefore, an important extension has been proposed to facilitate the modeling of complex systems. Typical extension is the addition of color that makes classical Petri net suitable for the representation and study of the complex manufacturing systems and called colored Petri net. Colored Petri nets inherit all the advantages of the classical Petri nets, such as the graphical and precise nature, the mathematical foundation and the analysis methods.

References

[1] Petri, C. A. "Communication with automata", English Translation, New York: Griffiss Air Force Base. Tech. Rep. RADC-TR-65-377, Vol. 1, 1962.

[2] Peterson J. L., "Petri nets.", ACM Computing Surveys, vol. 9, no. 3, pp. 223-252, 1977.

[3] Viswanadham, N. and Narahari, Y., "Performance modeling of Automated Manufacturing Systems", Prentice Hall, Englewood Cliffs, NJ 07632, 1992.

[4] Zuberek W. M., "Application of timed Petri nets to modeling and analysis of flexible manufacturing cells", Technical report 9503, Dept. of Computer science, Memorial University of Newfoundland, Canada, 1995.

[5] Sun, T. H., Cheng, C. W. & Fu, L.C. "A Petri net based approach to modeling and scheduling for an FMS and a case study.", IEEE Transactions on Industrial Electronics, pp 593-601, 1994.

[6] Zuberek W. M., "Scheduling of flexible manufacturing cells and their timed color Petri net models.", Proceeding of 1995 IEEE international conference on systems, pp 2142-2147, 1995.

[7] Zuberek W. M., "hierarchical derivation of schedules for manufacturing cells.", 9th Symp. On information control in manufacturing, France, pp 423-428, 1998.

1/8/2013

[8] Murata T., "Petri nets: properties, analysis and applications.", Proceedings of the IEEE, vol. 77, no. 4, 1989.

[9] Jiang Z., Zuo M. J. and Fung R. Y. K. "Stochastic Object-Oriented Petri nets (SOPNs) for reliability modeling of manufacturing systems.", Proceeding of the 1999 IEEE Canadian Conference on Electrical Computer Engineering, 1999.

[10] Jensen K., "An introduction to the theoretical aspects of coloured Petri nets.", Lecture Notes in Computer Science, Vol. 803, pp 230-272. Springer-Verlag, 1993.

[11] Jensen K., "A Brief Introduction to Coloured Petri Nets.", Lecture Notes in Computer Science, Vol. 1217: Tools and Algorithms for the Construction and Analysis of Systems. Proceedings of the TACAS'97 Workshop, Enschede, The Netherlands 1997, pp 201-208. Springer-Verlag, 1997.

[12] Jensen K., "An Introduction to the Practical Use of Coloured Petri Nets.", Lecture Notes in Computer Science, Vol. 1492: Lectures on Petri Nets II: Applications. Springer-Verlag, 1998.

[13] Zuberek W. M., "Cluster Tools With Chamber Revisiting-Modeling and Analysis Using Timed Petri Nets", IEEE transactions on semiconductor manufacturing, vol. 17, No. 3, August 2004.

[14] Mahsa G., Parisa A. B., Peter L. and Tianlong G., "Petri-net based formulation and algorithm for short-term scheduling of batch plants", Elsevier Ltd., Computers and Chemical Engineering 29 pg. 249–259, 2005.

[15] Bobbio, A., "System modeling with Petri nets", System Reliability Assessment, Kluwer P.C., pp 102-143, 1990.

[16] Hatono, I., Yarnagata, K., & Tamura, H. "Modeling and on-line Scheduling of flexible manufacturing systems using stochastic Petri nets.", IEEE Transactions on Software Engineering pp. 126-132, 1991.

[17] Lee, D.Y. & DiCesare, F. "Scheduling flexible manufacturing systems using Petri nets and heuristic search". IEEE Transactions on Robotics and Automation, pp. 123-132, (1994a April).

[18] Lee, D.Y. & DiCesare, F. "Integrated scheduling of flexible manufacturing systems employing automated guided vehicles", IEEE Transactions on Industrial Electronics, pp, 602-610, (1994b December).

[19] Lee, D.Y. & DiCesare, F. "Scheduling flexible manufacturing systems with the consideration of setup times", Proceedings of the 32nd Conference on Decision and Control. San Antonio, TX, pp. 3264-3269, (1993 December).

[20] Lee, D.Y. & DiCesare, F. "FMS scheduling using Petri nets and Heuristic Search" Proceedings of the 1992 IEEE International Conference on Robotics and Automation. Nice, France, pp. 1057-1062, (1992 May).