Safety Discrete Event Models for Holonic Cyclic Manufacturing Systems

Calin Ciufudean* and Constantin Filote*

"Stefan cel Mare" University, University str. 9,
720225 Suceava, Romania
{Calin Ciufudean,Constantin Filote}calin@eed.usv.ro,
filote@eed.usv.ro

Abstract. In this paper the expression "holonic cyclic manufacturing systems" refers to complex assembly/disassembly systems or fork/join systems, kanban systems, and in general, to any discrete event system that transforms raw material and/or components into products. Such a system is said to be cyclic if it provides the same sequence of products indefinitely. This paper considers the scheduling of holonic cyclic manufacturing systems and describes a new approach using Petri nets formalism. We propose an approach to frame the optimum schedule of holonic cyclic manufacturing systems in order to maximize the throughput while minimize the work in process. We also propose an algorithm to verify the optimum schedule.

Keywords: Petri net, event graph, holonic cyclic manufacturing system, minimum cyclic time, kanban system.

1 Introduction

The explosive development of information and communication technology networks (ICTNs) is designing and implementing more sophisticated and flexible approaches for measuring and controlling manufacturing processes. It does this by linking high level systems, such as enterprise research planning (ERP) and customer relation ship management (CRM) to low-level, real-time control devices such as digital signal processing (DSP) programmable logic controllers (PLCs), robots, conveyors, milling machines, etc [1-3]. For these approaches, the supervisory controllers can be distributed across several different computational devices. Following the holonic approach, the supervisory controller can be considered as a single holon associated with a physical device, although atomically it consists of several sub-holons, one for each computational device involved in the control of physical device [3]. For a whole-parted system Koestler denoted the term "holon", derived from the Greek word *holos* = whole with the suffix –on (like neutron or proton) pointing out the part characteristic [4]. A holon could compile strategy out of its rules, which fits to its intentions and

* Calin Ciufudean and Constantin Filote are with the Electrical Engineering and Computer Science Department of the "Stefan cel Mare" University.

V. Mařík, T. Strasser, and A. Zoitl (Eds.): HoloMAS 2009, LNAI 5696, pp. 225–233, 2009.

goals and with the interpolation of the environment. Typical holonic applications or models are applied mostly to the production processes, called holonic manufacturing systems (HMSs); we notice that similar to these approaches, they are modeled domain specifically [5, 6]. A complex holonic structure has a hierarchical order, rules and strategies framed by communication and reaction abilities. Most of the time, the hierarchies are looked at as rigid and inflexible shapes. However, some approaches are not following this like the approaches described in [6], and the one proposed here. An interesting situation occurs in both designing and controlling flexible manufacturing systems (FMSs), where there is no hardware support to ensure synchronization of the common resources flows (e.g. practical situations where the outcome that changes the states of FMS depends on the global sequence of events). In order to model and to ensure an optimized control of these situations, we propose an approach based on fundamental class of Petri nets that are well known to produce safe models and safe protocols for common flows of resources in large and complex systems.

In this paper, we focus our discussion on techniques for the predicting and verifying the performance of holonic distributed systems. We consider a holonic distributed system as a loosely or a tightly coupled processing element working cooperatively and concurrently on a set of related tasks. In general, there are two approaches for performance evaluation [7]: deterministic models and probabilistic models. In deterministic models, it is usually assumed that the task arrival times, the task execution times, and the synchronization involved are known in advance to the analysis. This approach is very useful for performance evaluation of real-time control systems with hard deadline requirements. In probabilistic models, the task arrival rates and the task service time are usually specified by probabilistic distribution functions. Probabilistic models usually give a gross prediction on the performance of a system and are usually used for the early stages of system design when the system characteristics are not well understood. We analyze the performance of holonic distributed systems and, in order to model clearly the synchronization involved in concurrent systems, the Petri net model is chosen. In this paper we consider event graphs as Petri nets in which each place has one input transition and one output transition. It has been shown that distributed systems can be modeled as event graphs [8, 9]. When the manufacturing times are deterministic (respectively stochastic), the cycle time (respectively the mean cycle time) of the model is the period (respectively the mean period) required to manufacture a given set of parts which fits with the required ratios. The smaller the cycle time (respectively the mean cycle time) the higher the productivity of the system.

When the firing times of transitions are deterministic, it is possible to define the cycle time of an elementary circuit. This is given by the ratio between the sum of the firing times associated with the transitions of the circuit and the number of the tokens in the places of the circuit, which is constant (we consider strongly connected graphs with the number of tokens in any elementary circuit, constant):

$$C_i = \frac{T_i}{N_i} \tag{1}$$

Where i = 1, 2, ... , n number of elementary circuits of the graph; C_i = cycle time of elementary circuits *i*; $T_i = \sum_{i}^{n} r_i$ = sum of the execution times of the transition in circuit *i*; $N_i = \sum_{i}^{n} M_i$ = total number of tokens in the places in circuit *i*.

In this case it has been proven that the cycle time of a strongly connected event graph is equal to the greatest cycle time of all elementary circuits. Furthermore, given a value C* greater than the largest firing time of all transitions, an algorithm has been proposed in [10] to reach a cycle time less than C*, while minimizing a linear combination of the token number in the places. The coefficients of the linear combination are the elements of a p-invariant. When the event graph is the model of a ratio-driven distributed system (such as manufacturing system), C* has to be greater than the largest cycle time of all command circuits [11]. A command circuit is an elementary circuit, which joins the transition that models the operations performed on the same machine [23 - 25]. Such a circuit contains one token to prevent more than one transition firing at any time in each elementary circuit.

In other words, C* must be greater than the time required by the bottleneck machine to perform a sequence of parts which fits with the production ratios. In the case of random firing times, it is no longer suitable for the elementary circuits to evaluate the behavior of the event graph in order to reach a given performance. Thus, the results presented in this paper, which aim at reaching a given mean cycle time in a steady state while minimizing a linear combination of the place markings, are particularly important at the preliminary design level of manufacturing systems working on a ratio-driven basis.

We also consider a few techniques for scheduling and verifying the throughput of holonic cyclic manufacturing systems (HCMSs). A holonic cyclic production system manufactures a set of products at a constant frequency. We notice that any holonic cyclic production system can be modeled as an event graph, and therefore it is possible to drive the optimal throughput using the properties of the event graphs [12 - 14] by scheduling the tasks of the HCMSs in order to minimize the work in process. One may notice that nowadays often these production systems display a distributed configuration, e.g. they can be seen as distributed systems. Section 2 frames the optimum behavior of the Petri net model of a holonic cyclic manufacturing system. In section 3 we give a sufficient condition for optimality, e.g. an algorithm that maximizes the throughput of scheduling production lines and we propose an algorithm to evaluate the bounds used to calculate the average cycle time. Section 4 concludes the paper.

2 Framing the Mean Cycle Time

In this paper we examine the holonic cyclic manufacturing systems (HCMSs), e.g. production tasks including assembly/disassembly or fork/join ones, with unreliable machines displayed in Jakson (infinite - capacity) networks and we consider the problem of maximizing the throughput by achieving a balance among the processing and repair of all machines under certain economical criteria.

It has been proven [14] that in an event graph a marking belonging to the optimal solution under a periodic operational mode (POM) is an optimal solution under the earliest operational mode (EOM). So, we consider the earliest operational mode of the event graph, and we assume only non pre-emptive transition firings. We further assume that, when the transition fires, the related tokens remain in the input places until the firing process ends. They then disappear, and one new token appears in each output place of the transition. We use the following notations: M_i = the marking of the elementary circuits, $i \in N$; $X_t^k \in R^+$ = random variable generating the time required for the k^{th} firing of the transition t, $k \in N$; $I_t(n)$ = instant of the n^{th} firing initiation of the transition t; E = set of elementary circuits; s(e) = sum of the random variables generating the firing; $\sum_{t \in e} X_t^1$ = sum of times of the transitions belonging to e; E_t = set of elementary circuits containing the transition t.

We assume that the sequences of transition firing times are independent sequences of integrable random variables. It was proven in [15] that there is a positive constant $s(M_0)$ so that:

$$\lim_{n \to \infty} \frac{I_t(n)}{n} = C_m \tag{2}$$

Where: C_m = the average cycle time of the event graph.

Furthermore, we denote by m_t the mean value of X_t^k and by q_t the standard deviation of X_t^k, i.e., $m_t = F[X_t^k]$ and $q_t^2 = F[(X_t^k - m_t)^2]$.

2.1 The Lower Bound of the Mean Cycle Time

The cycle time of the deterministic problem obtained by replacing the random variables which generate the firing times by their mean values, is a lower bound of the mean cycle time [16]. The following relation proven in [17] provides a better lower bound for the value of the mean cycle time than the previous one:

$$C^{**} \geq \max_{e \in E} F[\max\{ \frac{s[e \setminus \{t^*(e)\} + m_{t^*(e)}]}{M_0(e)},\ m_{t^*(e)} \}] \tag{3}$$

Where $t^*(e)$ is a transition, belonging to event e, with the greatest average firing time, i.e., $m_{t^*(e)} = \max_{t \in e} m_t$, and $s[e \setminus \{t^*(e)\}]$ is the sum of the firing times of the transitions belonging to e except $t^*(e)$.

2.2 The Upper Bound of the Mean Cycle Time

With M_0 being the initial marking, we derive a marking M_1 from M_0 by leaving the places which are empty in M_0, empty in M_1, and by reducing to one the number of tokens in the places containing more than one token in M_0.

Thus, $M_1(p) \leq M_0(p)$ for any set of places of the strongly connected event graph. An earliest operation mode running with the initial marking M_1 leads to a greater mean cycle time than the one obtained when starting from M_0. Then, starting from M_1, we apply to the event graph the earliest operation mode, but we block the tokens as soon as they reach a place already marked in M_1. This operation mode is referred to as the constrained mode [18]. We denote by C* the mean cycle time obtained by using the constrained operation mode when M_1 is the initial marking. We know [8, 9] that C* is greater than the mean cycle time obtained by using the earliest operation mode starting from M_1 which, in turn, is greater than the mean cycle time obtained with the earliest operation mode when the initial marking is M_0. Thus, C* is an upper bound of the solution to our problem (i.e. the mean cycle time obtained starting from M_0 when using the earliest operation mode). The following relation defines this upper bound:

$$C^* = F[\max_{z \in Z} s(z)] \tag{4}$$

Where, Z is the set of directed paths verifying the following properties:

- the origin and the extremity of any path is a marked place;
- there is no marked place between the origin and the extremity of the path.

3 The Evaluation of the Optimum Schedule of a HCMS

The problem is to determine the time at which each task should begin during the cycle. We assume that the set of tasks does not change, the tasks are not preemptive, and the task processing is deterministic. We determine two bounds for the mean cycle time of a Petri net model of a HCMS. The next step is to give an algorithm for optimizing the Petri net model in order to determine a feasible scheduling for the production cycle and to minimize the work in process, e.g. the number of cycles necessary to complete one item of each product, given that schedule [19-22].

Algorithm:

1. The tasks of the HCMS that require resources are scheduled in any order. We notice by s_i and by e_i the start, respectively the end time of task *i*. All tasks start in the interval $[0, C^*]$, and the resources can perform maximum one task at a time, where C^* is given by relation (4).
2. For each product p = 1, 2, …, n go to step 3 for the first task and go to step 4 for the rest of the tasks.
3. Let *i* be the first task in the sequence of tasks T_k, k = 1, 2, …, m necessary to produce one unit of k. Assign the table $T_k(i) = 0$ for this task.
4. For each product label task $T_k(j)$ as follows: if $s_j > e_i$ then set $T_k(j) = T_k(j) + 1$, otherwise set $T_k(j) = T_k(i)$.
5. For each product calculate the production time as follows:
   $C_p = [T_p(l) - T_p(f)] \cdot C_{med} + s_l - s_f$, where f is the first task, and l is the last one, in the sequences of tasks T_k, and $C_{med} = (C^* + C^{**})/2$, with C^{**} and C^* given by relation (3) , respectively relation (4).

6. For the entire production the total average work in process is $\sum_{p=1}^{n} \frac{C_p}{C_{med}}$, where p=1, 2, …, n is the number of products in a complete production cycle.

In the reminder of the previous section, we compare the previous bounds with the existing ones. Under the assumption of non-preemptive transition firing, it was proven in [16] that:

$$C^{**} \geq \max_{e \in E} F[\max\{ \frac{s\lfloor e \setminus \{t^*(e)\} + m_{t^*(e)} \rfloor}{M_0(e)}, m_{t^*(e)} \}] = \text{old lower bound} \quad (5)$$

$$C^* \leq \sum_t m_t = \text{old upper bound} \quad (6)$$

The following relations show that the new bounds are better than the old ones. But how close are they to the optimal solution? In order to answer this question we give the next algorithm, inspired from the operational research area, for verifying the system's performance:

a) Express the token loading in a (p x p) matrix P, where p is the number of places in the Petri net model of the system. Entry (A, B) in the matrix equals x if there are x tokens in place A and place A is connected directly to place B by a transition; otherwise (A, B) equals 0.

b) Express the transition time in a (p x p) matrix Q. The entry (A, B) in the matrix equals to the mean values of the random variables which generate the firing times (i.e., X_t^k) if A is an input place of the transition "i" and B is its output place. Entry (A, B) contains the symbol "w" if A and B are not connected directly as described above.

c) Compute matrix $C_{med} \cdot P$ - Q (with $p - w = \infty$, and $C_{med} = (C^* + C^{**})/2$, for $p \in N$), than use Floyd's algorithm to compute the shortest distance between every pair of nodes using matrix CP-Q as the distance matrix. The result is stored in matrix S. There are three cases.

1) All diagonal entries of matrix S are positive (i.e., $C_{med} \cdot N_k - T_k > 0$ for all circuits - see relation (1)) the system performance is higher than the given requirement;
2) Some diagonal entries of matrix S are zero's and the rest are positive (i.e., $C_{med} \cdot N_k - T_k = 0$ for some circuits, and $C_{med} \cdot N_k - T_k > 0$ for the other circuits) - the system performance has just reached the given requirement;
3) Some diagonal entries of matrix S are negative (i.e., $C_{med} \cdot N_k - T_k < 0$ for some circuits) - the system performance is lower than the given requirement.

In addition we may say that when a decision-free system runs at its highest speed, $C_{med} \cdot N_k$ equals to T_k for the bottleneck circuit. This implies that the places in the bottleneck circuit will have zero diagonal entries in matrix S. The system performance can be improved by reducing the execution times of some transitions in the circuit, or introducing more concurrency in the circuit (by modifying the initial marking), or increasing the mean cycle time (by choosing another average value for it).

4 The Evaluation of Kanban Systems

As we well know, an event graph can be used to model a kanban system. An example of a simple production line will be used to exemplify the above discussed problems. The production line consists of two machining tools (M_1 and M_2), two robot arms and two conveyors. Each machining tool is serviced by a dedicated robot arm, which performs load and unload tasks. One conveyor is used to transport work-pieces, a maximum of two at a time. The other conveyor is used to transport empty pallets. There are three pallets available in the system. Each work-piece is processed on M_1 and M_2, in this order.

The stochastic timed Petri net model of this system is shown in Fig.1. The initial marking of the net is $(300012111)^T$. When the time delays are modeled as random variables, it has become a convention to associate the time delays with the transition only. The involved transition has the associated time delays expressed in time units. The random variables X_1, X_2, X_3, X_4 are assigned to the transitions t_1, t_2, t_3, t_4, respectively. X_1 is uniformly distributed on [0.2], X_2 and X_3 are random variables with $F[X_2] = 11$ t.u. and $F[X_3] = 1$ t.u. X_4 is a constant and equals to 17 t.u. The Petri net model contains four loops.

The time delays associated with these loops, as well as their token contents are:

1) loop: $t_1 p_2 t_2 p_3 t_3 p_4 t_4 p_1 t_1$, loop delay: 30 u.t., token sum: 3, cycle time: 10 u.t.
2) loop: $t_1 p_2 t_2 p_5$ (or p_8) t_1, loop delay: 12 u.t., token sum: 1, cycle time: 12 u.t.
3) loop: $t_2 p_3 t_3 p_6 t_2$, loop delay: 2 u.t., token sum: 2, cycle time: 1 u.t.
4) loop: $t_3 p_4 t_4 p_7$ (or p_9) t_3, loop delay: 18 u.t., token sum: 1, cycle time: 18 u.t.

Thus, the minimum cycle time is 18 time units. This means that it takes a minimum of 18 time units to transform a raw workpiece into a final product. Computing the lower-bound and the upper bound of the cycle time of the event graph from Fig.1. by using the relations (3) and (4) , we obtain the values: $C^{**} \geq 12$ u.t.; $C^{*} \leq$ 18 u.t.

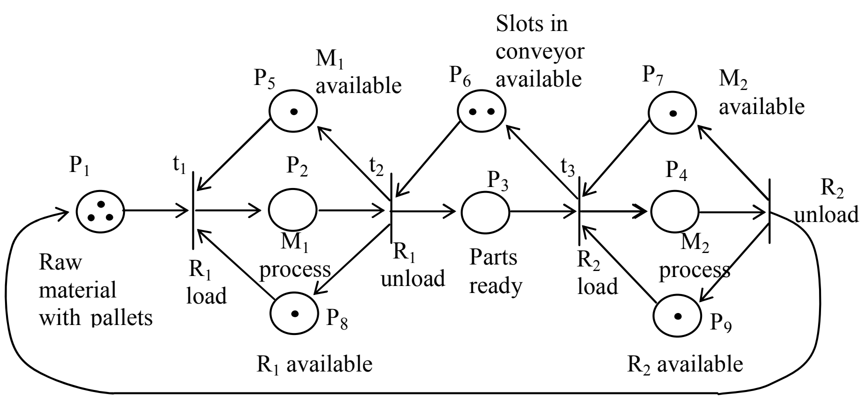

Fig. 1. Stochastic Petri net model for a manufacturing system

5 Conclusion

This work establishes a necessary relation between diagnosis and performances of holonic distributed systems, such as discrete event manufacturing systems. We have proposed a new architecture for the diagnosis and performance evaluation of holonic distributed systems using Petri nets formalisms and their properties [19, 20].

An important result in this paper is that it is always possible to reach a mean cycle time as close as possible to the greatest mean firing time using a finite marking, assuming that a transition cannot be fired by more than one token at each time. This result holds for any distribution of the transition firing time. An algorithm for verifying the distributed systems performance was introduced. An approach for computing the upper and lower bounds of the performance of a conservative general system is mentioned. However, the bounds produced may be loose.

Further research will focus on the conditions under which a mean cycle time can be reached with a finite marking.

Acknowledgments. The research was supported by Romanian National Council for Projects Management within Grant PN II No.71-065/2007.

References

1. Brusey, J., McFarlane, D.: Designing Communication Protocols for Holonic Control Devices using Elementary Nets. In: Mařík, V., William Brennan, R., Pěchouček, M. (eds.) HoloMAS 2005. LNCS (LNAI), vol. 3593, pp. 76–86. Springer, Heidelberg (2005)
2. Peters, R., Tobben, H.: A Reference Model for Holonic Supply Chain Management. In: Mařík, V., William Brennan, R., Pěchouček, M. (eds.) HoloMAS 2005. LNCS (LNAI), vol. 3593, pp. 221–232. Springer, Heidelberg (2005)
3. Bussman, S., McFarlane, D.C.: Rationales for Holonic Manufacturing Control. In: Proc. of the 2nd Int. Workshop on Intelligent Manufacturing Systems, Leuven, pp. 177–184 (1999)
4. Koestler, A.: Jenseits von Atomismus und Holismus – der Begriff des Holons, Wien (1970)
5. Delfmann, W., Albers, S.: Supply Chain Management in the Global Context, Koln (2000)
6. Lackmann, F., Nayabi, K., Hieber, R.: Supply Chain Management Software, Planungssysteme im Uberblick, Germany (2003)
7. Bacelli, F., Lin, Z.: Compresion properties of stochastic decision free Petri nets. IEEE Trans. Autom. Contr. 37(12), 1905–1920 (1992)
8. Laftit, S., Proth, J.M., Xie, X.L.: Optimization of invariant criteria for event graphs. IEEE Trans. Autom. Contr. 37(12), 547–555 (1992)
9. Proth, J.M., Xie, X.L.: Cycle time of stochastic event graphs: evaluation and marking optimization. IEEE Trans. Autom. Contr. 39(7), 1482–1486 (1994)
10. Campos, J., Chiola, G., Silva, M.: Properties and performance bounds for closed free choice synchronized monoclass quering networks. IEEE Trans. Autom. Contr. 36(12), 1368–1382 (1991)
11. Chauvert, F., Herrmann, J.W., Proth, J.M.: Optimization of Cyclic Production Sustems: A Heuristic Approach. IEEE Trans. On Robot. And Automat. 19(1), 15–154 (2003)
12. Proth, J.M., Xie, X.L.: Petri Nets. A Tool for Design and Management of Manufacturing Systems. Wiley, Chichester (1996)

13. Minis, I., Proth, J.M.: Production management in a Petri net environment. Recherche Operationelle 29(3), 321–352 (1995)
14. Zurawski, R., Zhon, M.C.: Petri nets and industrial applications: a tutorial. IEEE Trans. Ind. Electr. 41(6), 567–583 (1994)
15. Singh, J.: Operations Research. Penguin Books, Midlessex (1979)
16. Ciufudean, C., Popescu., D.: Scheduling Diagnosis of Flexible Manufacturing Systems. In: Proc. Int. Conf. on Autom, Contr. And Syst. Eng., Cairo (2005)
17. Ciufudean, C., Filote, C., Graur, A., Turcu, C.: Diagnosis of Complex Systems Using Ant Decision Petri Nets. In: The 1st Int. Conf. on Avail., Reliab. and Security, Vienna, pp. 473–482 (2006)
18. Ciufudean, C., Filote, C.: Performance Evaluation of Distributed Systems. In: International Conference on Control and Automation, Budapest, pp.21-25 (2005)
19. Yalcin, A., Boucher, T.O.: Deadlock Avoidance in Flexible Manufacturing Systems using Finite Automata. IEEE Trans. on Robot. And Automat. 16(4), 424–429 (2000)
20. Kouikoglou, V.: Optimal Rate Allocation in Unreliable Assembly/Disassembly Production Networks with Blocking. IEEE Trans. on Robot. And Automat. 16(4), 429–434 (2000)
21. Ciufudean, C., Satco, B., Filote, C.: Reliability Markov Chains for Security Data Transmitter Analysis. In: The 2nd Int. Conf. on Avail., Reliab. and Security, Vienna, pp. 886–892 (2007)
22. Recalde, L., Teruel, E., Silva, M.: Modeling and analysis of sequential processes that cooperate through buffers. IEEE Trans. on Rob. and Autom. 14(2), 267–277 (1998)
23. Zuo, M.J., Liu, B., Murthy, D.N.P.: Replacement-repair policy for multi-state deteriorating products under warranty. European Journal of Operational Research (123), 519–530 (2000)
24. Hopkins, M.: Strategies for determining causes of events, Technical Report R-306, UCLA Cognitive Systems Laboratory (2002)
25. Hall, K.H., Staron, R.J., Vrba, P.: Experience with Holonic and Agent-Based Control Systems and Their Adoption by Industry. In: Mařík, et al, pp. 1–10 (2005)