

## MULTI-AGENT CONTROL OF HOLONIC MANUFACTURING SYSTEMS BASED ON PETRI NET

**Fu-Shiung Hsieh**

*Overseas Chinese Institute of Technology  
Taiwan, R.O.C.*

**Abstract:** Holonic manufacturing systems requires a robust and effective mechanism to reorganize available resources to cope with uncertainties to achieve the production goal. This paper combines the contract net protocol and Petri net to adaptively synthesize Petri net to control holonic manufacturing systems. The main results include: (1) a multi-agent model for holonic manufacturing systems based on contract net protocol, (2) a collaboration network formation process to reorganize resources to accomplish a task, (3) a collaborative Petri net model of collaborative networks and (4) characterization of liveness conditions to award contracts in multi-agent systems. *Copyright © 2002 IFAC*

**Keywords:** agents, flexible manufacturing systems, Petri net, production control.

### 1. INTRODUCTION

Modern manufacturing systems have to face constant and dynamic changes, such as customer specification changes, machine break-down, hot orders and other kinds of disturbances. This requires the manufacturing system to be adaptive to cope with changes and uncertainties. The holonic concept has been proposed as an efficient paradigm for developing such an adaptive manufacturing system. Holonic manufacturing systems (HMS) (Wyns, 1999; Brussel, 1998) are the next wave of manufacturing revolution to deal with dynamic changes. The features of HMS pose several challenging design issues, including: a robust mechanism to reorganize the available resources to achieve the production goal by forming a set of resources to accomplish the assigned tasks through collaboration and an effective mechanism to cope with uncertainties.

There are several research works on HMS (Liu, 2000; Mihaela, 2000; Shu, 2000). However, the issue of deadlock (Hsieh, 2001; Reveliotis, 1999; Lawley, 1998a; Lawley, 1998b; Reveliotis, 1997; Cho, 1995; Ezpeleta, 1995; Hsieh, 1994; Wysk, 1991; Banaszak, 1990; Viswanadham, *et al.*, 1990) has not been

addressed in these papers. This paper focuses on the development of a framework to effectively and dynamically organize available resources to robustly achieve the goal of production while avoiding undesirable states such as deadlocks in the context of holonic manufacturing systems.

Multi-agent framework (Ferber, 1999; Nilsson, 1998) provides many desirable characteristics to proactively handle uncertainties and conduct error recovery. We model a HMS as a cooperative multi-agent system. To optimise system performance, the contract net protocol (Smith, 1980) is applied to achieve effective and robust resource allocation for HMS based on multi-agent framework. Petri nets have been demonstrated as effective tools for modeling and control of manufacturing systems (Jeng, 1997; Zhou, 1992; Zhou, 1991; Murata, 1989; Zhou, 1989; Narahari, 1985). How to combine the modeling and analysis capability of Petri nets with the robust contract net protocol to dynamically distribute tasks based on available resources is an interesting problem. Our results include: (1) a multi-agent model for the HMS based on contract net protocol (2) a collaboration network formation process to reorganize resources to accomplish a task

(3) a Petri net model of collaborative networks (4) characterization of feasible and liveness conditions to award contracts in multi-agent holonic manufacturing systems.

The remainder of this paper is organized as follows. Section 2 proposes a multi-agent model for HMS and briefly reviews the collaborative network formation process to reorganize resources. Section 3 introduces Collaborative Petri Net (CPN) model for collaborative networks. Section 4 details the feasible and liveness conditions based on CPN. Section 5 presents the collaborative algorithms to award contracts in multi-agent system. Section 6 concludes this paper.

## 2. A MULTI-AGENT COLLABORATIVE NETWORK MODEL FOR HMS

A HMS is usually modelled as a cooperative multi-agent system (Ferber, 1999; Nilsson, 1998) with three types of basic elements: resource holons, product holons, and order holons according to the PROSA reference architecture developed by Wyns (Wyns, 1999). A resource holon consists of a production resource in the HMS and relevant components that control the resource. A product holon contains the production process and product knowledge to ensure the correct fabrication of products with sufficient quality. An order holon represents a manufacturing order. Let  $A$  denote the set of all agents, including the set of all resource holon agents,  $A_R$ , the set of all product holon agents,  $A_P$  and the set of all order holon agents  $A_O$ . That is,  $A = A_R \cup A_P \cup A_O$ . Such a multi-agent system can be represented by a set of nodes. Interactions among the set of agents are through the well-known contract net protocol. In contract net protocol, there are two roles agents can play: manager or bidder. Four stages are involved for establishment of contracts between a manager and one or more bidders:

- (1) Request for tenders: In this stage, the manager announces a task to all potential bidders. The announcement contains the detailed description of the task.
- (2) Submission of proposals: On receiving the tender announcement, bidders capable of performing the task draw up proposals and submit to the manager.
- (3) Awarding of contract: On receiving and evaluating the submitted proposal, the manager awards the contract to the best bidder.
- (4) Establishment of contract: If the awarded bidder commits itself to carry out the required task, it will send a message to the manager and become a contractor. Otherwise, the awarded bidder might refuse to accept the contract by notifying the manager and trigger a re-evaluation of the bids and awarding of the contract to another bidder.

As original contract net protocol does not take into account the internal process of each agent, direct application of the original contract net protocol to holonic manufacturing systems may not yield satisfactory results. In the following, we will tailor

the original contract net protocol for holonic manufacturing systems by introducing a more detailed production process model based on Petri Net. The new extended contract net protocol is effective for handling resource allocation and task distribution of holonic manufacturing systems.

We propose a distributed collaborative network formation process based on contract net protocol (Smith, 1980). The order holon agents always act as managers to initiate a tender process while the product holon agents act as the bidders. The product holon agents act as the managers in turn to issue a new request for bids to resource holon agents for the set of subtasks stemming from the product requirements. In other words, product holon agents are the contractors while resource holon agents are the subcontractors for the order holon agent. In the remaining of this paper, we consider the case that the product holon agent first asks the resource holon agents to commit themselves to it before committing itself to the order holon agent. We also assume that the resource holon agents reserve the required resources for every proposal submitted. The formation of collaborative network is based on the four-phase contract net protocol by sending different messages in each phase.

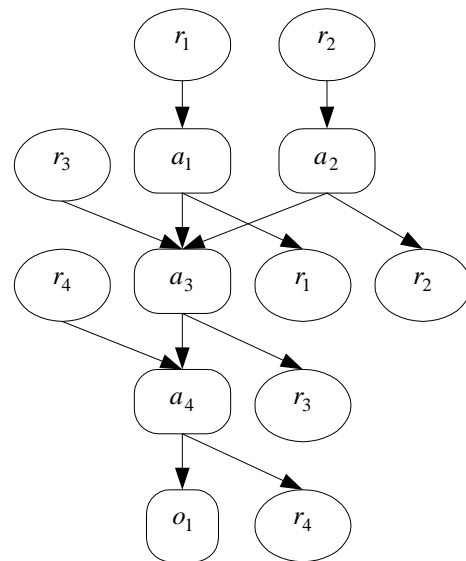


Fig. 1

Assume that each order consists of only one type of products and the classes of holonic manufacturing systems under consideration exhibit the following features.

- A.1: Each order holon agent will establish contract with only one product holon agent called host product holon agent.
- A.2: Each manager product holon agent may establish contracts with one or more bidder product holon agents and one or more bidder resource holon agents, where a bidder product holon agent represents jobs in progress (which may hold some resources) and a bidder resource agent represents machine required for the next operation.
- A.3: For a manager product holon agent that establish contracts with more than one bidder product

holon agents, at most one of the jobs associated with the bidder product holon agents holds resources for the next operation. All the other jobs associated with the remaining bidder product holon agents release all the resources before establishing contracts with the manager product holon agent.

Depending on the available resources and the structure of production process, the resulting collaborative network varies. Figure 1 illustrates a production process consisting of four subprocesses represented by product holon agent  $a_1, a_2, a_3$  and  $a_4$ , respectively. Four resource holon agents  $r_1 \sim r_4$  are involved in the production activities of product holon agent  $a_1, a_2, a_3$  and  $a_4$ . Figure 1 demonstrates four contracts have been established between resource holon agents  $r_1 \sim r_4$  and product holon agent  $a_1 \sim a_4$ , and one contract has been established between product holon agent  $a_4$  and order holon agent  $o_1$ . A directed arc from  $r_i$  to  $a_i$  represents an established contract between resource holon agent  $r_i$  and product holon agent  $a_i$ . For the example of Figure 1, product holon agent  $a_4$  is the host product holon agent.

To represent the resource holding or release by a product holon agent, for a resource holon agent  $r$  that has established and executed contract with a product holon agent  $a$  and will be released after the contract has been executed, add a solid directed arc from product holon agent  $a$  to resource holon agent  $r$ . A solid directed arc from a resource holon agent or product holon agent to a product holon agent represents an established contract. A directed arc from a product holon agent to a resource holon agent means that the resource will be released after the contract between them is executed. The node associated with the resource holon agent is called a release node. The digraph obtained using the above procedure is called a commitment graph. Fig. 1 demonstrates a commitment graph.

### 3. COLLABORATIVE PETRI NET MODELS

The system introduced above exhibits several properties, including: concurrent, synchronous or asynchronous events. To facilitate the analysis, we adopt Petri Net as the modeling tool. This Section first gives a brief introduction to Petri Net notation and then presents a procedure to construct collaborative Petri Net for a given commitment graph.

A Petri Net (PN)  $G$  is a five-tuple  $G = (P, T, I, O, m_0)$ , where  $P$  is a finite set of places with cardinality  $|P|$ ,  $T$  is a finite set of transitions,  $I \subset P \times T$  is a set of transition input arcs,  $O \subset T \times P$  is a set of transition output arcs, and  $m_0 : P \rightarrow Z^{|P|}$  is the initial marking of the PN with  $Z$  as the set of nonnegative integers. The

marking of  $G$  is a vector  $m \in Z^{|P|}$  that indicates the number of tokens in each place and is a state of the system. The readers may refer to (Murata, 1989) for further definitions such as enabled transitions, transition firing rules and the set of reachable markings of the PN  $G$  from an initial marking  $m_0$ , denoted as  $R_\infty(N(m_0))$ .

To make it more efficient to compute the release of resources, we first convert the commitment graph into a Petri net. The conversion is performed by the following procedure.

#### Procedure to construct collaborative Petri Net

Step1: Create a transition for each product holon agent  $a$  in the commitment graph.

Step2: Create a place for each solid directed arc from a resource holon agent to a product holon agent in the commitment graph. Create a directed arc from the created place to the transition. Add a token in the created place.

Step 3: Create a place for each solid directed arc from a product holon agent (bidder) to a product holon agent (manager) in the commitment graph. Create a directed arc from the transition corresponding to the bidder product holon agent to the created place. Create a directed arc from the created place to the transition corresponding to the manager product holon agent. Add a token in the created place.

Step 4: Create a place for each release node associated with each resource holon agent, where a release node is a node that has one or more solid directed arc from product holon agent to it. Create a directed arc from the transition associated with product holon agent to the place.

The Petri net constructed using the above procedure is a deterministic, acyclic Petri net called collaborative Petri Net (CPN) defined as follows.

Definition 3.1: A collaborative Petri Net (CPN) associated with a collaborative network  $j \in J$  is defined as a six tuple

$N_j = (P_j, T_j, I_j, O_j, m_{j0}, u_j)$ , abbreviated as  $N_j(m_{j0}, u_j)$ , where  $u_j$  is a commitment policy defined based on commitment actions of a given CPN as follows.

Definition 3.2: A commitment action  $a$  is a vector in  $Z^{|P_j|}$  that determines how many times that each transition in  $T_j$  may be fired concurrently. We will use  $a(t)$  to denote the number of transition firing of  $t$  allowed under  $a$ . A commitment policy  $u_j$  is a mapping that generates a sequence  $\{a_n\}$  of commitment actions for the CPN  $N_j$  based on its initial marking  $m_{j0}$ . That is,  $u_j : M_0(N_j) \rightarrow$

$(Z^{|P_j|})^\infty$ , where  $M_0(N_j)$  is the set of admissible initial markings of  $N_j$ .

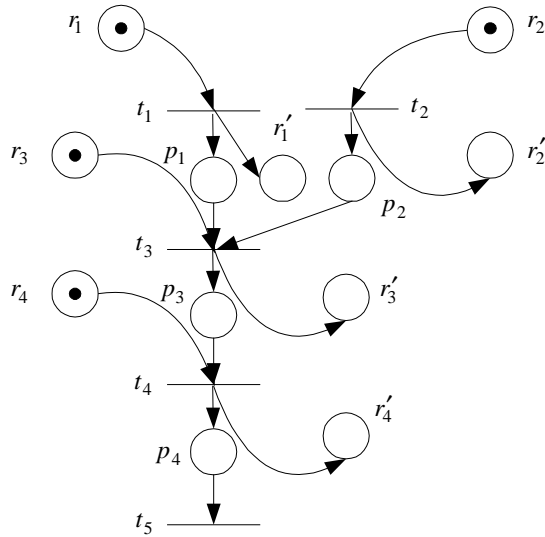


Fig. 2

Example: Fig. 2 illustrates the collaborative Petri net associated with Figure 1, with places  $r_1, r_2, r_3$  and  $r_4$  representing the resource holon agents and transitions  $t_1, t_2, t_3$  and  $t_4$  representing the four product holon agents:  $a_1, a_2, a_3$  and  $a_4$  and transition  $t_5$  representing order holon agent  $o_1$ .

#### 4. LIVENESS CONDITION OF CPN

A precondition for a multi-agent system to be deadlock free is the existence of sufficient resources. For a system with only one CPN, say  $N_j$ , in it, let  $\underline{R}_j$  denote a minimal set of resources required to complete the operations of  $N_j$ . As there is one to one correspondence between a transition of a CPN and an agent of the associated commitment graph,  $\underline{R}_j$  can be obtained based on  $N_j$ . As  $N_j$  is of tree structure, a minimal resource requirement can be computed based on its minimal firing sequence.

Let  $\bullet t$  denote the set of input places of  $t$ , not including its control place. Let  $t^\bullet$  denote the set of output places of  $t$ . Let  $\square t = \{t' \mid t' \bullet \subseteq \bullet t\}$ . Suppose  $\square t = \{t_1, t_2, \dots, t_{K-1}, t_K\}$ .

Let  $R_s$ , a vector in  $Z^{|R|}$ , denote the set of resources required for firing a sequence of transitions,  $s$ . Let  $R_t$ , a vector in  $Z^{|R|}$ , denote the resources required to fire transition  $t$ .

Definition 4.1: A firing sequence  $s$  is said to be a minimal firing sequence for firing a transition  $t$  of  $N_j$  if (1) transition  $t$  occurs only once in  $s$  (2)  $t$  is the last transition in  $s$  and (3) there does not exist any

firing sequence  $s'$  with  $R_{s'} < R_s$ . A minimal firing sequence to fire transition  $t$  is denoted as  $s_t^*$ .

$$R_{s_t^*} = \max(R_{s_{t_1}^*}, R_{s_{t_2}^*}, \dots, R_{s_{t_K}^*}) \dots \dots \dots (1)$$

Let  $s_j^*$  denote a minimal firing sequence of  $N_j$ . A minimal resource requirement of a CPN  $N_j$  can be computed by firing a minimal firing sequence  $s_j^*$ .  $\underline{R}_j$  can be computed using (1).

The above result can be applied to calculate the resource requirement for multi-agent collaborative network under the extended contract net protocol. For each product holon agent  $a \in A_p$ , let  $A_p(a)$  denote the subset of adjacent product holon agents preceding product holon agent  $a$ . Suppose  $K = |A_p(a)|$  and  $A_p(a) = \{a_1, a_2, \dots, a_K\}$ .

Definition 4.2: Let  $Q_a$ , a vector in  $Z^{|R|}$ , denote the set of resources required to complete the subtasks of all product holon agents preceding  $a$  and including  $a$ . Let  $R_a$ , a vector in  $Z^{|R|}$ , denote the resource requirement to complete subtasks of  $a$ .

$$Q_a = \max_{a' \in A_p(a)} Q_{a'} \dots \dots \dots (2)$$

Obviously,  $\underline{R}_j$  can be computed using equation (2).

Let  $a$  denote the host product holon agent associated with  $N_j$ . Then  $\underline{R}_j = Q_a$ .

Theorem 4.1: Given a CPN  $N_j$  with marking  $m_j$ , there exists a commitment policy  $u_j$  such that  $N_j(m_j, u_j)$  is live if and only if there exists a sequence of commitment actions that bring  $m_j$  to a marking  $m'_j$  under which  $R_j(m'_j) \geq \underline{R}_j$ , where  $R_j(m'_j) \in Z^{|R|}$  denotes the set of resources in idle state under  $m'_j$ .

Remark that application of Theorem 4.1 requires computation of  $\underline{R}_j$  and  $R_j$ .

Theorem 4.1 considers systems with only one collaborative network. However, there may be multiple collaborative networks existing in the system to handle different orders. To analyse systems with more than one collaborative networks, consider a system with a set  $\mathcal{N} = \{N_j, j \in \mathcal{J}\}$  of CPNs. Let us denote the system with the set  $\mathcal{N}$  of CPNs as  $N = \cup_{j \in \mathcal{J}} N_j$ , where  $N \equiv (P, T, I, O, m, u)$ , abbreviated as  $N(m, u)$  or  $N$  and  $u$  is jointly defined by  $u_j, j \in \mathcal{J}$ . The following Theorem establishes a liveness condition based on  $N = \cup_{j \in \mathcal{J}} N_j$ .

Theorem 4.2: Given a set of CPNs  $N = \bigcup_{j \in J} N_j$  with marking  $m$ , there exists a commitment policy  $u$  such that  $N(m, u)$  is live if and only if there exists a sequence of commitment actions that bring  $m$  to a marking  $m'$  under which  $R(m') \geq \underline{R}$ , where  $R(m') \in Z^{|\underline{R}|}$  denotes the set of resources in idle state under  $m'$  and  $\underline{R}$  is a minimal set of resources required to complete the tasks of  $N$  with  $\underline{R} \equiv \sum_{j \in J} \underline{R}_j$ .

In a multi-agent environment, an agent usually has only knowledge about itself but has only limited knowledge regarding other agents. In our multi-agent model, we assume that only host product holon agent has complete information of the CPN formed by itself. A host product holon agent has no idea about the CPN of other product holon agents. However, we assume that each host product holon agent is able to obtain information such as status of resource usage from others by sending and receiving messages. To avoid conflicts or undesirable states such as deadlocks, each product holon agent need to check the feasibility before making a commitment with other product holon agents or resource holon agents. In the following we first establish preliminary results to compute  $\underline{R}_j, R_j, \underline{R}$  and  $R$ . In the next section, we will develop algorithm to conduct the feasibility test based on Theorem 4.1 and Theorem 4.2.

Remark that application of Theorem 4.2 requires computation of  $\underline{R}$  and  $R$ . Note that  $R \equiv \sum_{j \in J} R_j$ . To compute  $R_j$  by exploiting the structure of CPNs, we need the following definition.

Definition 4.3: A token flow path  $\pi$  is a directed path consisting of alternating places and transitions. A token flow path starting with a place  $p_1$  without input transitions and ending with a place  $p_2$  without output transitions is denoted as  $\pi(p_1, p_2)$ .  $\Pi_p$  denotes the set of token flow paths ending with place  $p$ . Let  $\pi(m) = \sum_{p \in \pi} m(p)$  denote the total number of tokens of  $\pi$  under a marking  $m$ , where  $p \in \pi$  means that  $p$  is a place of  $\pi$ . The following lemma follows directly from Theorem 8 of (Murata, 1989).

Lemma 4.1: Given a controlled acyclic marked graph (acyclic  $MG$ ) with all of the source transitions in it disabled forever and with all non-source transition enabled forever, the maximum number of tokens that will flow to place  $p$  is  $\min_{\pi \in \Pi_p} \pi(m)$ .

$R_j$  can be efficiently obtained by the following flooding algorithm  $N_j$  to compute  $\min_{\pi \in \Pi_p} \pi(m)$ .

Flooding Algorithm to Compute  $\min_{\pi \in \Pi_p} \pi(m)$

Step 1: For each product holon agent acting as bidder, send a message containing information of the number of jobs (tokens) on hand to the associated manager product holon agent.

Step 2 : On receiving the message sent in Step 1, the manager product holon agent compute  $\min_{\pi \in \Pi_p} \pi(m)$  to

obtain the information that the number of resources that can be returned.

## 5. COLLABORATIVE ALGORITHMS

In this Section we propose a solution algorithm that guarantees all orders can be completed based on collaboration among resource holon agents and product holon agents. Although each product holon agent may act proactively and independently, conflicts may occur due to resource sharing among different product holon agents. Therefore coordination among different product holon agents is needed to resolve the potential conflicts due to inappropriate commitment between agents.

Before establishing contracts with other agents, a host product holon agent first executes contract net protocol to form a collaborative network. Then a collaborative Petri Net model associated with the collaborative network is constructed. Finally, the host product holon agent obtains resource status information from other product holon agents to conduct feasibility test. If it passes the feasibility test, the contracts will be established. Otherwise, it will reduce the number of concurrently established contracts with resource agents. Eventually, the algorithm will end with a feasible solution. An outline of the algorithm to generate feasible commitment as follows.

Algorithm to generate feasible commitment

Step 1: Execute contract net protocol.

Step 2: Construct collaborative Petri Net model  $N_j$ .

Step 3: For each  $j \in J$ , obtain  $R_j$  and  $\underline{R}_j$ .

Step 4: Let  $J' = J$ ,

If  $R \equiv \bigcup_{j \in J'} R_j \geq \underline{R}$  Then

$J'$  is feasible

Else

Reduce one resource allocation

End If

## 6. CONCLUSIONS

This paper focuses on modeling and control of HMS based on multi-agent systems. A multi-agent model is proposed for task allocation of holonic manufacturing systems. A collaborative network formation process based on contract net protocol to reorganize resources and a graph theoretical model to represent the collaborative network are also presented in this paper. To facilitate the analysis, the

collaborative networks is converted to a Petri Net representation called collaborative Petri Nets (CPN). A deadlock free conditions based on liveness analysis of collaborative Petri Nets is established. A collaboration algorithm has been developed to generate feasible commitment based on collaborative Petri Nets.

#### REFERENCES

- Banaszak Z. and B. Krogh, (1990) Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows, *IEEE Trans. Robot. Automat.*, vol. 6, pp. 724-734.
- Brussel H.V., Wyns J., Valckenaers P., Bongaerts L. and P. Peeters. (1998) *Reference architecture for holonic manufacturing systems*, 37:255-274.
- Cho H., T. K. Kumaran and R. A. Wysk, (1995) Graph-theoretic deadlock detection and resolution for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.11, pp. 413-421.
- Ezpeleta J., J.M. Colom, and Martinez, (1995) A Petri Net based deadlock prevention policy for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.11, pp. 173-184.
- Ferber J., (1999) *Multi-Agent Systems, An Introduction to Distributed Artificial Intelligence*, Addison Wesley.
- Hsieh F.S., (2001) Deadlock Free Task Distribution and Resource Allocation for Holonic Manufacturing Systems based on Multi-Agent Framework, *Proceedings of 2000 IEEE International Conference on Systems, Man and Cybernetics*, Tucson, Arizona, pp. 2463-2468, Oct. 7-10, 2001.
- Hsieh F.S. and S. Chang, (1994) Dispatching driven deadlock avoidance controller synthesis for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.10, pp. 196-209.
- Jeng M. D. and F. DiCesare, (1993) A review of synthesis techniques for Petri nets with applications to automated manufacturing systems, *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 1, pp. 301-312.
- Jeng M. D. , (1997) Petri Nets for Modeling Automated Manufacturing Systems with Error Recovery, *IEEE Trans. Robot. Automat.*, vol.13, no. 5, pp. 752-760.
- Lawley M. A., (1998a) Deadlock avoidance for production systems with flexible routing, *IEEE Trans. Robot. Automat.*, vol.15, pp. 497-509.
- Lawley M. A., Spyros A. Reveliotis and Placid M. Ferreira, (1998b) A correct and scalable deadlock avoidance policy for flexible manufacturing systems, *IEEE Trans. Robot. Automat.*, vol.14, pp. 796-809.
- Liu S., William A. Gruver, Dilip Kotak, Shahriar Bardi, (2000) Holonic Manufacturing System For Distributed Control of Automated Guided Vehicles, *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 1727-1732.
- Mihaela U., Stefanoiu D., Douglas N., (2000) Holonic Self-Organization of Multi-Agent Systems by Fuzzy Modelling with Application to Intelligent Manufacturing, *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 1661-1666.
- Murata T., (1989) Petri Nets: Properties, Analysis and Applications, *Proceedings of the IEEE*, vol. 77, No. 4.
- Narahari Y. and N. Viswanadham, (1985) A Petri net approach to the modeling and analysis of flexible manufacturing systems, *Annals Oper. Res.*, vol. 3, pp. 449-472.
- Nilsson N.J., (1998) *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann Publishers, Inc. San Francisco, California.
- Reveliotis S.A., M.A. Lawley and P.M. Ferreira, (1997) Polynomial-Complexity Deadlock Avoidance Policies for Sequential Resource Allocation Systems," *IEEE Trans. Automat. Contr.*, vol.42, pp. 1344-1357.
- Reveliotis S.A., (1999) Accommodating FMS operational contingencies through routing flexibility, *IEEE Trans. Robot. Automat.*, vol.15, pp. 3-19.
- Smith R.G., (1980) The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver, *IEEE Trans. On Computers* Vol. 29, pp. 1104-1113.
- Shu S., Mitch Wilkes, and Kazuhiko Kawamura , (2000) Development of Reusable, Configurable, Extensible Holonic Manufacturing System, *Proceedings of the IEEE International Conference on System, Man and Cybernetics*, pp. 1679-1684.
- Viswanadham N., Y. Narahari, and T.L. Johnson, (1990) Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models, *IEEE Trans. Robot. Automat.* Vol. 6, pp. 713-723.
- Wyns J., (1999) *Reference Architecture for Holonic Manufacturing*, PhD Dissertation, PMA Division, Katholieke Universiteit, Leuven, Belgium.
- Wysk R. A., N.S. Yang, and Joshi, (1991) Detection of deadlocks in flexible manufacturing cells, *IEEE Trans. Robot. Automat.* vol. 7, pp. 853-859.
- Zhou M. C., F. DiCesare and A.A. Desrochers, (1992) A hybrid methodology for synthesis of Petri net models for manufacturing systems, *IEEE Trans. Robot., Automat.*, vol. 8, no. 3, pp. 350-361.
- Zhou M. C. and F. DiCesare, (1991) Parallel and sequential mutual exclusions for Petri net modeling for manufacturing systems with shared resources, *IEEE Trans. Robot., Automat.*, vol. 7, no. 4, pp. 515-527.
- Zhou M.C. and F. DiCesare, (1989) Adaptive design of Petri net controllers for error recovery in automated manufacturing systems, *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 963-973.