Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Reconfiguration Mechanism for Holonic Manufacturing Systems**

Fu-Shiung Hsieh*

** Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taiwan (e-mail:fshsieh@cyut.edu.tw)*
*** This paper is supported in part by National Science Council of Taiwan under grant NSC95-2416-H-324-009-MY2*

Abstract: Although holonic manufacturing systems (HMS) have been recognized as a paradigm to cope with changes in manufacturing environment based on a flexible architecture, several characteristics of HMS such as re-configurability have not been characterized quantitatively. To realize these advantages, more concrete mechanisms or methodologies need to be developed. The objectives of this paper are to probe into the characteristics of reconfigurability for HMS and propose design methodologies to realize these characteristics. We formulate and study a holarchy reconfiguration problem to deal with dynamic removal/addition of holons in HMS due to failure/recovery of resources. Finding a solution from the scratch to deal with resource failures is not an appropriate approach as it may lead to chaos at the shop floor. To achieve effective reconfiguration in HMS, a viable solution must be based on the nominal configuration and cooperation of holons. We define an impact function to characterize the impact of changes due to different holons in a holarchy. A reconfiguration mechanism based on the impact function is proposed to effectively reconfigure the systems to achieve minimal cost solutions.

1. INTRODUCTION

Since its invention, holonic manufacturing systems (HMS) (Balasubramanian 2001; Wyns, 1999; Van Brussel *et al.* 1998; Christensen 1994) have been recognized as a paradigm to accommodate changes and meet customers' requirements dynamically. HMS is based on the notion of holon (Koestler, 1967) to provide a reconfigurable, flexible and decentralized manufacturing architecture. Koestler proposed the word "holon" to describe a basic unit of organization in biological and social systems. A holon is an autonomous, co-operative and intelligent entity able to collaborate with other holon to process the tasks. In HMS, a holon can be part of another holon and a set of holons that autonomously cooperate to achieve a goal forms a holarchy. The loosely coupled structure of holons makes it possible to attain for manufacturing systems the benefits of stability in the face of disturbances, adaptability and flexibility in the face of change, and efficient use of available resources.

In spite of the promising perspective and the research developed by the holonic community (Brennan, Fletcher and Norrie, 2002;Deen et al., 2003), HMS leave some important questions open (Marik, 2004), namely how to achieve global optimisation in decentralised systems? how should the production control structure evolve to adapt to change? how to specify formally the dynamic behaviour of holonic systems? how to introduce self-organisation capabilities? and how to develop holonic-based control applications? In existing literature, works on how to specify formally the dynamic behaviour of holonic systems appear in (Leitão, Colombo and Restivo, 2003; Hsieh, 2006). Existing results on how to achieve global optimisation in decentralised systems can be found in (Hsieh 2008). Studies on how should the production control structure evolve to adapt to changes have been made by Leitão *et al.* (Leitão & Restivo, 2006). Leitão *et al.* present an agile and adaptive manufacturing control architecture that addresses the need for the fast reaction to disturbances at the shop floor level, increasing the agility and flexibility of the enterprise. Despite the existing research results, there is still a lack of study on how to design an effective reconfiguration mechanism taking advantage of the holonic system architecture. The goal of this research aims at proposing a systematic design methodology to develop reconfiguration mechanism for HMS to deal with changes in holons, including dynamic removal/addition of holons in HMS. While the process to handle removing of holons involves identifying of the parts of holarchy influenced and replacing them by alternative holons, the process to handle addition of holons only involves searching for alternative holons. Therefore, we will focus on the mechanism to handle removal of holons in this paper.

Failure and recovery of resources are two common events leading to removal/addition of holons in HMS. Resources may fail at any point in time, reduce the set of available resources, take place in any holarchy and change its characteristics. For some types of tolerable resource failures, the operation of a holarchy may still be maintained. In this case, reconfiguration may not be required. An interesting question is how to quantitatively characterize the condition under which reconfiguration is not required. The effects of resource failures may propagate from one holon to another. Development of an effective method to analyze the impact of resource failures on the operation of an HMS is a significant issue. Another question is to establish the condition under which an HMS can be dynamically reconfigured to accommodate changes due to resource failures. A challenge in this research direction is due to the distributed architecture of HMS, which limits the availability of information that can be used to reconfigure the system. How to design a reconfiguration mechanism based on the architecture of HMS is a significant issue. The key question of interest is: how to reconfigure HMS to cope with resources failures cost-effectively without causing chaos at the shop floor. Answers to the above questions are crucial to the development of an effective reconfiguration mechanism.

Finding a solution from the scratch from the point resource failures occur is not an appropriate approach as it may lead to chaos at the shop floor. An effective approach to reconfiguration must be based on the nominal solution so that the impact on the operation of the other existing active holarchies can be significantly reduced. To study the reconfiguration mechanism, we define a holarchy reconfiguration problem (HRP), which is based on the holarchy optimization problem (HOP) in (Hsieh, 2008). We study the condition for the existence of a solution to HRP. To propose a solution methodology, a hybrid model that combines contract net protocol (CNP) (Smith, 1980) with Petri nets (Murata, 1989) for HOP has been proposed (Hsieh, 2008). Based on a similar modeling technique, we formulate and study the HRP problem. We propose Petri net models for the holarchies formed in HMS and study the impact of resource failures on the system. We define an impact function to characterize the impact of resource failures on a holarchy. We study the tolerable failure conditions for HMS to survive without reconfiguration. In case the resource failures do not satisfy the condition, the reconfiguration algorithm based on the impact function is initiated to reconfigure the systems to achieve minimal cost solutions while minimizing the impact on the existing tasks at the shop floor.

To achieve effective reconfiguration in HMS, a viable solution must be based on cooperation of holons. The reconfiguration algorithm relies on the interactions and cooperation between four types of holons: detector, initiator, standby and optimizer. A detector is a resource holon that detects resource failures. There are two types of initiators: Type I initiator and Type II initiator. A Type I initiator is a holon to be removed from the system and it usually contains a detector and the operation will be blocked eventually due to resource failures. A Type 2 initiator is a holon to be dynamically added to the system. While the reconfiguration process triggered by Type I initiators to remove holons from a system and is broken down into two stages: (1) initiator identification & standby notification and (2) process optimization, the reconfiguration process triggered by Type II initiators is simpler and is broken down into two stages: (1) initiator identification and (2) process optimization. For this reason, we will focus on the development of reconfiguration mechanism for Type I initiator in this paper to save space.

A Type I initiator will forward standby request to the upstream composite holons which are called standby holons. A standby holon will stay in pending state and may attempt to submit a proposal and establish a contract with another downstream composite holon within a time period again. If no new contract is established by the time period, the standby holon will restore their resources back to the idle states. An optimizer is the downstream holon of the manager of the initiator responsible for finding the optimal configuration based on the remaining resources. We establish a condition for the existence of an optimal solution to HRP.

This paper is differentiated from the works of Leitão *et al.* (Leitão, Colombo and Restivo, 2003; Leitão and Restivo, 2006) in several aspects. Although all these works intend to contribute to the development of a dynamic and adaptive control approach that improves the agility and reaction to unexpected disturbances by taking advantage of the flexibility offered by HMS without compromising the global optimisation, the approaches are different. The self-organization adaptation mechanism proposed by Leitão *et al.* introduces the concept of autonomy factor and a pheromone-like spreading mechanism to propagate emergence and reorganize the system. Our approach first checks whether resource failures are tolerable. If the resource failures are tolerable, no reconfiguration is required. Otherwise, only the composite holons influenced by the resource failures will be notified by a standby request. In addition, as our strategy achieves reconfiguration cost-effectively based on the nominal solution, it minimizes the impact on existing tasks and avoids chaos in the shop floor.

The remainder of this paper is organized as follows. In Section 2, we introduce the reconfiguration problem in HMS. In Section 3, we review Petri net models for holarchies. In Section 4, we study the impact of resource failures. In Section 5, we propose our reconfiguration algorithm. In Section 6, we conclude this paper.

2. HOLARCHY RECONFIGURATION PROBLEM

The dynamics and behaviours of HMS are determined by the interactions of three types of holons: resource holons, product holons, and order holons. A resource holon consists of a production resource with relevant components to control the resource. A product holon contains the production process information to manufacture products. An order holon represents an order. Individual product holons or resource holons cannot process a complex task alone. To process a task, a set of resource holons and product holons form a holarchy. In HMS, a holon can be part of another holon and a set of holons that can autonomously cooperate to achieve a goal forms a holarchy. Holarchy formation problem has been studied in (Hsieh; 2008) to propose models and develop collaborative algorithms to guide the holons to form a holarchy that coherently moves toward the desired goal state ultimately.

A holarchy consists of a set of composite holons, where a composite holon consists of a product holon and a set of resource holons. Fig. 1(a) is a diagraph that shows the dependency between holons, where each node denotes a composite holon. For a node, an outgoing arc labelled with a number i means the corresponding holon may provide type-i tasks to other holons. An incoming arc labelled with a number i means the holon may accept type-i tasks from other holons. A composite holon is denoted as $c_n(U_n, R_n)$ may depend on a set of upstream composite holons $U_n \subseteq \boldsymbol{U}_n$ that provide parts to it and a set of resource holons $R_n \subseteq \boldsymbol{R}_n$, where we use $\boldsymbol{U}_n$ to denote the set of potential upstream composite holons of c_n and $\boldsymbol{R}_n$ to denote the set of potential resources holons of c_n, respectively.

In Fig. 1(a), $\boldsymbol{U}_7 = \{1,2\}$, $\boldsymbol{U}_9 = \{3,4,5,6\}$. We use $\boldsymbol{U}_n(i)$ to denote the set of potential upstream composite holons that provide type-i parts to c_n. $c_n(U_n, R_n)$. A potential composite holon $c_{n'}$, where $n' \in \boldsymbol{U}_n$, plays the role of a bidder and submits a proposal $\pi_{n'}$ to product holon h_n to indicate the part types provided and the related timing information. A resource holon h_r $r \in \boldsymbol{R}_n$, may submit a proposal π_r to product holon h_n to indicate its ability, the number of resources to be allocated, the timing and costs to execute the required operations by h_r. Let $\Pi_n(S)$ denote the set of all proposals submitted to the manager h_n under state S. Let w_n denote the cost function that specifies the cost of $c_n(U_n, R_n)$. A composite holon is feasible if there exists an execution sequence to accomplish the desired task. The holarchy optimization problem can be stated informally as follows:

Holarchy optimization Problem (HOP):

$$\min_{R_n, U_n} w_n = \min_{R_n, U_n} \left(\sum_{r \in R_n} w_r + \sum_{n' \in U_n} w_{n'} \right)$$

s.t. $c_n(U_n, R_n)$ is feasible, $R_n \subseteq \boldsymbol{R}_n$ and $U_n \subseteq \boldsymbol{U}_n$.

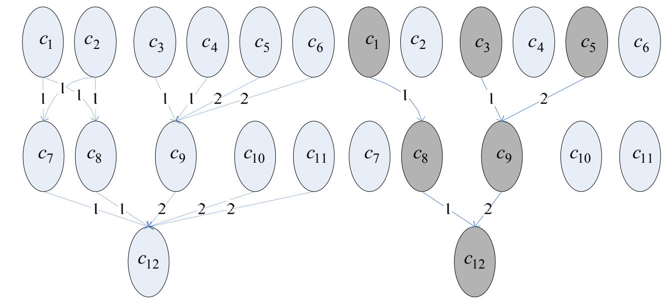
Fig.1 (a) Potential dependency D (b)Holarchy H

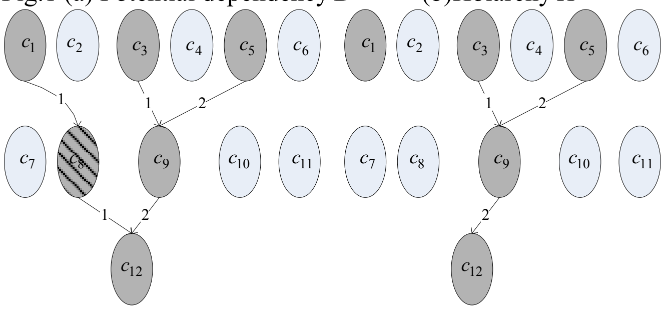
Fig. 1(c)Reconfiguration Initiator (d)Contract Cancellation

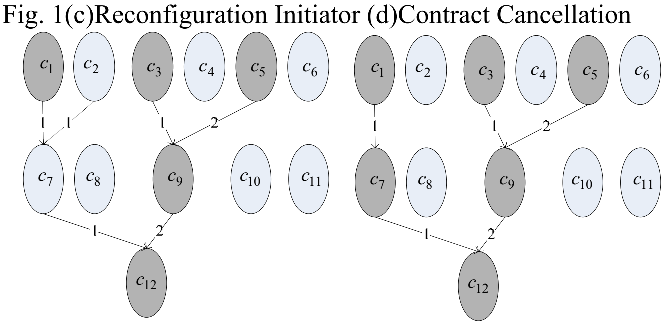
Fig. 1(e)Reconfiguration Process (f)Reconfigured Holarchy

A holarchy may be formed by solving HOP. Fig. 1(b) shows a holarchy. If no resource fails in the course of executing the operations, the task will be terminated normally. In case resource failures occur, the effects may propagate from one holon to another. An effective reconfiguration mechanism must be developed to deal with resource failures. Otherwise, the operation of HMS may be crippled and the performance of HMS will degrade. Finding a solution from the scratch from the point resource failure occurs is not an appropriate approach as it may lead to chaos in the shop floor. An effective approach to HRP must be based on the nominal solution $c_n(U_n^*, R_n^*)$ of HOP so that the impact on the operation of the other existing holarchies can be significantly reduced. Fig. 1(c) shows a scenario in which resource failed in c_8 and makes c_8 unable to terminate the tasks. An effective reconfiguration strategy is to replace the parts of the holarchy in the upstream of c_8 with other alternative holons while retaining the rest of the holarchy. In the reconfiguration process, we classify the composite holons into three types: initiator, standby and optimizer. In Fig. 1(c), c_8 is an initiator, which cancels the contracts with its upstream composite holons as shown in Fig. 1 (d) and each composite holon in the upstream of the initiator c_8 (e.g. c_1) become a standby holon. The initiator c_8 notifies its manager (called the primary optimizer) c_{12} to reconfigure the holarchy relevant to c_7. Each composite holon that acts as a manager is an optimizer, e.g. c_{12} and c_7. In Fig. 1 (e), c_7 is an optimizer that negotiates with c_1 and c_2. Fig. 1(f) shows a possible reconfigured holarchy.

We formulate a holarchy reconfiguration problem (HRP) to cope with the resource failures. Given a nominal solution $c_n(U_n^*, R_n^*)$ to HOP under a given state S, HRP aims at finding a solution based on the proposal $\Pi_n(S')$ under state S' that reflects the resource failures.

Holarchy Reconfiguration Problem (HRP)

$$\min_{R_n', U_n'} w_n = \min_{R_n', U_n'} \left(\sum_{r \in R_n'} w_r + \sum_{n' \in U_n'} w_{n'} \right)$$

s.t. $c_n(U_n', R_n')$ is feasible, $R_n' \subseteq \boldsymbol{R}_n'$ and $U_n' \subseteq \boldsymbol{U}_n'$, where $\boldsymbol{R}_n' = \{r \mid \pi_r \in \Pi_n'(S'), h_r$ is a resource holon $\}$ and $\boldsymbol{U}_n' = \{n \mid \pi_n \in \Pi_n'(S'), c_n$ is a composite holon $\}$.

3. MODELING HOLARCHIES

To formulate the reconfiguration problem, a mathematical model for holarchies formed in HMS is required. To capture the synchronous/asynchronous and concurrent activities in HMS, we adopt Petri nets as our modelling tool. A brief introduction to Petri net is given first. A time Petri Net (PN) (Murata, 1989) G is a six-tuple $G = (P, T, F, m_0, SI)$, where P is a finite set of places, T is a finite set of transitions, $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, $m_0 : P \to Z^{|P|}$ is the initial marking of the PN with Z as the set of nonnegative integers and $SI : T \to R^+ \times R^+$ is a mapping called the static interval that specifies the static earliest firing time and the static latest firing time for each transition. A marking of G is a vector $m \in Z^{|P|}$ that indicates the number of tokens in each place under a state. ${}^{\bullet}t$ denotes the set of input places of transition t. A transition t is enabled and can be fired under m iff $m(p) \geq F(p,t)\ \forall p \in {}^{\bullet}t$. Firing a transition removes one token from each of its input places and adds one token to each of its output places. A marking m' is reachable from m if there exists a firing sequence s bringing m to m'. The reachability set $R(m_0)$ denotes all the markings reachable from m_0. A Petri net G with marking m_0 is live if, no matter what marking has been reached from m_0, it is possible to ultimately fire any transition of G by progressing through some further firing sequence.

Definition 3.1: The workflow of product holon h_n is an acyclic marked graph $H_n = (P_n, T_n, F_n, m_{n0}, SI_n)$. As each transition represents a distinct operation in a task, $T_j \cap T_k = \Phi$ for $j \neq k$.

An activity is a sequence of operations to be performed by a certain type of resources. Γ_n denotes the set of resource types required to perform the operations of h_n. Let γ denote the idle state place of type-γ resources. The $k-th$ activity is described by either a circuit place that starts and ends with an idle state place γ or a directed path that starts with an idle state and end with a non-idle state place in Petri net. A circuit indicates that the resource activity includes resource allocation and de-allocation. A directed path indicates that the resource activity includes either resource allocation or de-allocation but not both. The Petri net model for the $k-th$ activity for a type-γ resources, where $\gamma \in \Gamma_n$, is described by a Petri net H_γ^k defined as follows.

Definition 3.2:Petri net $H_\gamma^k = (P_\gamma^k, T_\gamma^k, F_\gamma^k, m_\gamma^k, SI_\gamma^k)$ denotes the $k-th$ activity for a type-γ resources, where $\gamma \in \Gamma_n$. Remark that $T_\gamma^k \cap T_\gamma^{k'} = \Phi$ for $k \neq k'$.

Let K_γ be the number of activities of a type-γ resources. Let $\Omega_n^\gamma \subseteq \{1,2,3,..., K_\gamma\}$ denote the set of type-γ activity IDs in H_n . The initial marking m_γ^k is determined based on the set Π_n of proposals submitted to product holon h_n . More specifically, $m_\gamma^k(\gamma) = \sum_{\pi_r \in \Pi_n, r \in R_n, \alpha(r)=\gamma} \pi_r^q$, where γ is the type-γ resource idle state place.

To capture the interactions between a product holon h_n and resource holons, we define operator "$\|$" to combine two Petri nets into a new one by merging common places, transitions, or arcs.

Definition 3.3: Composite holon $c_n(U_n, R_n)$ that depends on the set U_n of composite holons is modeled by $C_n = \|_{\gamma \in \Gamma_n} H_\gamma \| H_n$, where $H_\gamma = \|_{k \in \Omega_n^\gamma} H_\gamma^k$. We use $C_n(m_{n0}, u_n)$ to denote the Petri net model of $c_n(U_n, R_n)$. where $m_{n0}(\gamma) = \sum_{k \in \Omega_n^\gamma} m_\gamma^k(\gamma)$ and u_n is a control policy of C_n .

Fig. 2 shows examples of C_n .

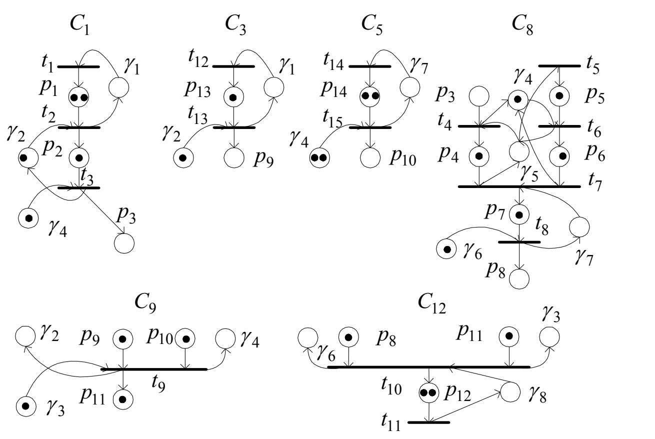

Fig. 2 Petri net Models of C_n

Let N_A denote the set of composite holons in a holarchy A .

Definition 3.4: The overall Petri net model for A is constructed based on composite holon C_n $\forall n \in N_A$ by $C_A = \|_{n \in N_A} C_n$ =(P_A , T_A , F_A , m_{A0} , SI_A , u_A), where u_A is a control policy of C_A .

A holarchy A is feasible if there exists a control policy u_A under which $C_A(m_{A0}, u_A)$ is live. The following property provides a condition to check whether holarchy A is feasible.

Property 3.1:There exists a control policy u_A under which $C_A(m_A, u_A)$ is live if and only if there exists a marking m'_A that can be reached from m_A and $m'_A \geq m_A^*$.

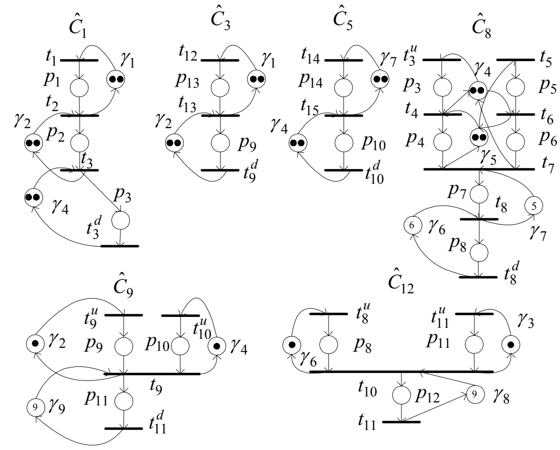

Fig. 3 Petri net Models of $\hat{C}_n$

Although Property 3.1 provides a condition to test the feasibility of A , it is difficult to construct C_A in distributed environment. Instead of checking the liveness of C_A directly, we study the feasibility relation between a holarchy and the composite holons that constitute it by establishing the relation between m_A^* and $\hat{m}_n^*$.

Definition 3.5: Let $\hat{C}_n(\hat{m}_{n0}, u_n)$ =($\hat{P}_n$, $\hat{T}_n$, $\hat{F}_n$, $\hat{m}_{n0}$, $\hat{SI}_n$, u_n) be the Petri net that augments C_n by adding an output transition t_n^d to the terminal output place in $h_n^\bullet$ and adding an arc from to each resource place without input transition, where u_n is a control policy of C_n .

Fig. 3 shows examples of $\hat{C}_n$. Liveness condition of $\hat{C}_n$ can be characterized as follows.

Definition 3.6: $\hat{M}_{n0}^*$,denotes the set of minimal markings under which there exists a control policy u_n under which $\hat{C}_n(\hat{m}_n^*, u_n)$ is live.

Property 3.2: There exists a control policy u_n under which $\hat{C}_n(\hat{m}_{n0}, u_n)$ is live if and only if the initial marking $\hat{m}_{n0} \geq \hat{m}_n^*$, where $\hat{m}_n^* \in \hat{M}_{n0}^*$.

In addition to the above properties, the following property establishes a relation between m_A^* and $\hat{m}_n^*$.

Property 3.3: An upper bound of is $m_A^*(\gamma) \leq \sum_{n \in N_A} \hat{m}_n^*(\gamma) \forall \gamma \in \Gamma_A$.

4. IMPACT OF RESOURCE FAILURES

To realize our proposed scheme for reconfiguration, we develop a method to identify whether a holarchy is a reconfiguration initiator. By definition, a reconfiguration initiator is a holarchy in which resource failures occur and influence the operation. Resource failures in holarchy c_n can be represented by a perturbation δ_n in marking m_n of C_n . To characterize the impact of resource failures, we need the following definition.

Definition 4.1: Let N_A denote the set of composite holons in the system. Suppose there exist control policies u_n $\forall n \in N_A$ such that C_n can be kept live under u_n from marking m_n $\forall n \in N_A$. $C_{n'}$ is influenced by δ_n if there does not exist control policies u'_n $\forall n' \in N_A$ such that $C_{n'}$ can be kept live under $u'_{n'}$ from marking $m_{n'}$ after removing δ_n from m_n .

We use the impact function $I(\delta_n, C_{n'})$ to denote whether $C_{n'}$ is influenced by δ_n .

Definition 4.2: The impact function is defined by $I(\delta_n, C_{n'}) = \begin{cases} 1 \; if \; C_{n'} \; is \; \inf luenced \; by \; \delta_n \\ 0 \; otherwise \end{cases}$.

Property 4.1: C_n is an initiator for a given failure δ_n if $I(\delta_n, C_n)$ =1.

Occurrence of δ_n may block the operation of C_n and the effects may ripple backward to the upstream of C_n . The following property states the ripple effects of δ_n .

Property 4.2: If $I(\delta_n, C_n) = 1$, $I(\delta_n, C_{n'}) = 1 \; \forall n' \in U_n$.

Example: For Fig. 7, suppose $I(\delta_8, C_8) = 1$. As $U_8 = \{1\}$, $I(\delta_8, C_1) = 1$.

The set $\{n' | n' \in U_n, I(\delta_n, C_{n'}) = 1, n' \in U_n, I(\delta_n, C_n) = 1\}$ denotes the standby holons with respect to δ_n .

Property 4.3: If $\delta_n = \underline{0}$, $I(\underline{0}, C_n) = 0 \; \forall n \in N_A$.

Although $I(\delta_n, C_n)$ provides a way to determine whether a holarchy C_n is an initiator with respect to δ_n , due to computational complexity, an approximate impact function $\bar{I}(\delta_n, C_{n'})$ which satisfies $\bar{I}(\delta_n, C_{n'}) \geq I(\delta_n, C_{n'})$ $\forall n, n' \in N_A$ is defined based on the potential resource release function $PR_n(\bullet)$ below. The function $PR_n(\bullet)$ is defined based on a mathematical model AC_n that combines the model of product holons with the resource holons involved to compute the number of resources that will return to idle state.

Definition 4.3: Let $\tilde{H}_\gamma^k = (\tilde{P}_\gamma^k, \tilde{T}_\gamma^k, \tilde{F}_\gamma^k, \tilde{m}_\gamma^k, \tilde{S}I_\gamma^k)$ denotes the acyclic Petri net model for the $k-th$ activity of a type-γ resources, where $\gamma \in \Gamma_n$. Remark that $T_\gamma^k \cap T_\gamma^{k'} = \Phi$ for $k \neq k'$.

$\tilde{H}_\gamma^k$ is constructed by splitting each resource idle place in H_γ^k into two places: one resource allocation place γ_i^k and the other resource de-allocation place γ_o^k . $AC_n = \|_{\gamma \in \Gamma_n} \tilde{H}_\gamma \| H_n$, where $\tilde{H}_\gamma = \|_{k \in \Omega_n^\gamma} \tilde{H}_\gamma^k$.

Fig. 4 shows examples of AC_n corresponding to C_n in Fig. 2.

To compute the number of tokens that will return to idle state place in AC_n , we define a token flow path as follows.

Definition 4.4: A token flow path λ consists of the sequence of places in a directed path of AC_n . $L_n(p)$ denotes the set of token flow paths in AC_n ending with place p . The total number of tokens in λ of AC_n under submarking $\tilde{m}_n$ is denoted as $\lambda(\tilde{m}_n) = \sum_{p \in \lambda} \tilde{m}_n(p)$, where $p \in \lambda$ means that p is a place in λ

The number of type-γ resources that will return to the set of places $\{\gamma_o^k\}$ by firing all the transitions in AC_n as many times as possible under $\tilde{m}_n$ is $\sum_{k \in \Omega_n^\gamma} \min_{\lambda \in L_n(\gamma_o^k)} \lambda(\tilde{m}_n)$.

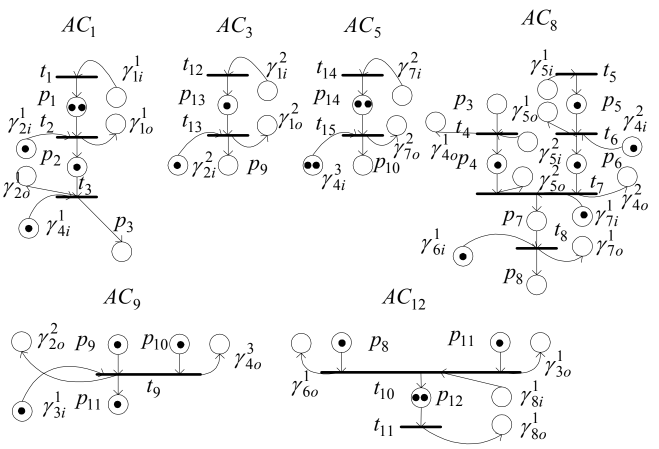

Fig. 4 Petri net Models of AC_n

Definition 4.5: The potentially released resources from AC_n is represented by a vector PR_n , where $PR_n \in Z^{|\Gamma_n|}$ and $PR_n(\gamma, \tilde{m}_n)$ denotes the number of type-γ resources potentially released from $\tilde{m}_n$.

A lower bound on $PR_n(\gamma, \tilde{m}_n)$ is $PR_n(\gamma, \tilde{m}_n)$ = $\sum_{k \in \Omega_n^\gamma} \min_{\lambda \in L_n(\gamma_o^k)} \lambda(\tilde{m}_n)$.

For Fig. 4, $PR_1(\gamma_1) = PR_1(\gamma_2) = 1$, $PR_3(\gamma_1) = 1$, $PR_5(\gamma_7) = 2$, $PR_8(\gamma_4) = 1$, $PR_8(\gamma_5) = 2$, $PR_8(\gamma_7) = 1$, $PR_9(\gamma_2) = PR_9(\gamma_4) = 1$, $PR_{12}(\gamma_3) = 1$, $PR_{12}(\gamma_6) = 1$, $PR_{12}(\gamma_8) = 1$.

Property 4.4 states a condition for a set of holarchies $N_A = \{1,2,3,..., N\}$ to maintain feasibility collaboratively.

Property 4.4: Suppose $N_A = \{1,2,3,..., N\}$. If $PR(\gamma) = \sum_{n \in N_A} PR_n(\gamma, \tilde{m}_n) \geq m_A^*(\gamma) \forall \gamma \in \Gamma_A$, there exists a control policy under which C_n is feasible for each $n \in N_A$.

For Fig. 4 and Fig. 5, $N_A = \{1,3,5,8,9,12\}$.

$PR(\gamma_1) = 2, PR(\gamma_2) = 2, PR(\gamma_3) = 1, PR(\gamma_4) = 2, PR(\gamma_5) = 2,$
$PR(\gamma_6) = 1, PR(\gamma_7) = 3, PR(\gamma_8) = 1$.

Obviously, $PR(\gamma) = \sum_{n \in N_A} PR_n(\gamma, \tilde{m}_n) \geq m_A^*(\gamma) \forall \gamma \in \Gamma_A$.

There exists a control policy under which C_n is feasible for each $n \in N_A$.

We first characterize the condition under which $\bar{I}(\delta_n, C_{n'}) = 0$ and then establish a condition for $\bar{I}(\delta_n, C_{n'}) = 1$. $\bar{I}(\delta_n, C_{n'}) = 0$ corresponds to the conditions under which δ_n is tolerable. Let $\Delta_n(m_n)$ denote the set of tolerable resource failures for C_n under marking m_n.

$\Delta_n(m_n) = \{\delta_n \big| \sum_{n \in N_A} PR_n(\gamma, \tilde{m}_n - \delta_n) \geq \hat{m}_A^*(\gamma) \forall \gamma \in \Gamma_A\}$.

We have the following property.

Property 4.5: If $\delta_n \in \Delta_n(m_n)$, $\bar{I}(\delta_n, C_{n'}) = 0 \; \forall n' \in N_A$.

Corollary 4.1:If $\delta_n \in \Delta_n(m_n)$, $\bar{I}(\delta_n, C_{n'}) = 0 \; \forall n' \in N_A$. All the holarchies can maintain their operations and optimality without reconfiguration.

Theorem 4.1:If $\bar{I}(\delta_n, C_{n'}) = 0 \forall n' \in N_A$, all the holarchies can maintain their operations and optimality without reconfiguration.

To characterize the condition for $\bar{I}(\delta_n, C_{n'}) = 1$, we define the set of initiators X as follows.

Definition 4.6: Let $X = \{n' \big| \delta_{n'} > 0, n' \in N_A\} \setminus \{n' \big| \sum_{n \in N_A} PR_n(\gamma, \tilde{m}_n - \delta_n) \geq \hat{m}_A^*(\gamma) \forall \gamma \in \Gamma_A\}$.

Example: For Fig. 4, suppose $\delta_8(p_5) = 1$, $\delta_8(p) = 0 \forall p \neq p_5$ and $\delta_n(p) = 0 \forall p \in P_n, \forall n \neq 8$. In this case, $X = \{8\}$.

The approximate impact function $\bar{I}(\delta_n, C_{n'})$ is defined based on the set of initiators X by setting $\bar{I}(\delta_n, C_n) = 1$ for each $n \in X$.

Occurrence of δ_n may block the operation of C_n and the effects of blocking may ripple backward to the upstream of C_n. The following property states the ripple effects of δ_n that is valid for $\bar{I}(\delta_n, C_n)$.

Property 4.6: If $\bar{I}(\delta_n, C_n) = 1$, $\bar{I}(\delta_n, C_{n'}) = 1 \; \forall n' \in U_n$.

Based on the set of initiators X, it follows from Property 4.6 that Property 4.7 holds for $\bar{I}(\delta_n, C_n)$.

Property 4.7: $\bar{I}(\delta_n, C_{n'}) = \begin{cases} 1 \; if \; n' \in B \cup X \\ 0 \; otherwise \end{cases}$, where B satisfies the following properties.

(i) For each $n \in X$, $U_n \subseteq B$.

(ii) For each $n \in B$, $U_n \subseteq B$.

Example: For Fig. 4, suppose $\delta_8(p_5) = 1$, $\delta_8(p) = 0 \forall p \neq p_5$ and $\delta_n(p) = 0 \forall p \in P_n, \forall n \neq 8$. In this case, $X = \{8\}$ and $B = \{1\}$.

Example: Suppose $\delta_8(p_5) = 1$, $\delta_8(p) = 0 \forall p \neq p_5$ and $\delta_n(p) = 0 \forall p \in P_n, \forall n \neq 8$. $\bar{I}(\delta_8, C_{n'}) = 0 \forall n' \in N_A - \{1,8\}$. $\bar{I}(\delta_8, C_8) = 1$ and $\bar{I}(\delta_8, C_1) = 1$.

5. RECONFIGURATION ALGORITHM

The reconfiguration algorithm is based on the impact function $\bar{I}$. The impact function $\bar{I}(\delta_n, C_{n'})$ characterizes whether a composite holon $C_{n'}$ is influenced by δ_n . If $\bar{I}(\delta_n, C_{n'}) = 0$, δ_n has no influence on the operation of composite holon $C_{n'}$. In this case, it is not required to reconfigure $C_{n'}$. Each composite holon $C_{n'}$ with $\bar{I}(\delta_n, C_{n'}) = 1$ must be reconfigured properly. Otherwise the composite holon may eventually be blocked. We classify the set of holons involved in the reconfiguration process into three types: initiator, standby and optimizer. A composite holon C_n is an initiator with respect to δ_n if $\bar{I}(\delta_n, C_n) = 1$. Each composite holon in X is an initiator whereas each composite holon in B is a standby. An optimizer is a composite holon responsible for optimization of the reconfigured process. The reconfiguration processes depend on the collaboration of the initiator, standbys and the optimizers. The reconfiguration process is triggered by a resource failure δ_n and is broken down into two stages: (1) initiator identification & standby notification and (2) process optimization. To determine whether $n' \in X$, we define a resource monitor holon to maintain the up-to-date information of $PR_n(\gamma, \tilde{m}_n - \delta_n)$ for each $n \in N_A$. A composite holon that has detected resource failures δ_n must determine whether it is an initiator by sending a request to the resource monitor holon to obtain $PR_n(\gamma, \tilde{m}_n) \forall \gamma, n \in N_A$. The following algorithm is applied to identify initiator and standby holons.

Algorithm to Identify Initiator and Notify Standby Holons

Given $c_n(U_n, R_n)$, δ_n, N_A, $\tilde{m}_n$

Step 1: Send $PR_n(\gamma, \tilde{m}_n - \delta_n)$ to Resource Monitor

Step 2: Request Resource Monitor to provide $PR(\gamma) = \sum_{n \in N_A} PR_n(\gamma, \tilde{m}_n - \delta_n)$ each for each $\gamma \in \Gamma_A$.

Step 3: Identify Initiator and Notify Standby Holons

```
If PR(γ) ≥ m*_A(γ) ∀γ ∈ Γ_A
    Ī(δ_n, C_n) = 0
    Exit
Else
    Ī(δ_n, C_n) = 1
    For each n' ∈ U_n
        Send Standby Request to C_n'
        Wait for Response from C_n'
    End For
    For each r ∈ R_n
        Restore resources to idle state
    End For
    Send Optimization Request to the primary optimizer
End If
```

An initiator sends a Standby Request to bring active relevant composite holons to standby state and to cancel the previously established contract. After finalizing the initiator identification and notification of standbys, each initiator sends an optimization request to the downstream composite holon which takes the role of an optimizer and proceeds to the process optimization stage.

Example:Suppose $\delta_8(p_7) = 1$, $\delta_8(p) = 0 \forall p \neq p_7$ and $\delta_n(p) = 0 \forall p \in P_n, \forall n \neq 8$. In this case, $\bar{I}(\delta_8, C_8) = 1$. According to Property 4.6, $\bar{I}(\delta_8, C_1) = 1$. In this case, C_8 is an initiator as shown in Fig. 1(c), C_1 is a standby and C_{12} is an optimizer.

In the optimization process, there are two types of optimizers: primary optimizer and secondary optimizer. There is only one primary optimizer, which is the manager of the initiator. All the other optimizers are secondary optimizers. The optimization process is started by the primary optimizer. The optimization process propagates from the primary optimizer to its upstream. The primary optimizer first finds the set of required part types $I_R = \{TYPE(p) \big| p \in {}^\bullet h_n^c\}$, where ${}^\bullet h_n^c \subseteq {}^\bullet h_n$ denotes the set of places with which no contract has been established and $TYPE(\bullet)$ is a function that maps a place to the corresponding part type. The primary optimizer then issues a request for tender message which carries I_R to the set of potential bidders. The primary optimizer waits for the set of potential bidders $\boldsymbol{U}_n(i)$ to submit the set of proposals $\Pi_n(S')$ under state S' , where S' is the resulting state reached after the cancellation of the contracts between the Initiator and the primary optimizer. The optimizer needs to search for alternative composite holons to replace the set of composite holons based on $\Pi_n(S')$. The following theorem states a condition for the existence of an optimal solution to the Holarchy Reconfiguration Problem.

Theorem 5.1: Let h_n be the primary optimizer. There exists an optimal solution to HRP under state S' if and only if $U_n(p) \neq \Phi \ \forall p \in {}^\bullet h_n^c$ and $\hat{m}_{n0} \geq \overline{m}_n^*$.

6. CONCLUSION

Effective reconfiguration mechanisms are essential to successfully taking advantage of the flexible architecture of HMS to effectively deal with dynamic removal/addition of holons due to resource failures while avoiding chaos at the shop floor. We formulate a holarchy reconfiguration problem (HRP). We define an impact function to characterize the impact of resource failures on a holarchy. We study the tolerable failure conditions for HMS to survive without reconfiguration. In case the resource failures do not satisfy the condition, the reconfiguration algorithm based on the impact function is initiated to reconfigure the systems to achieve minimal cost solutions while minimizing the impact on the existing tasks at the shop floor. We propose a reconfiguration mechanism for solving HRP based on the interactions and cooperation between four types of holons: detector, initiator, standby and optimizer. There are two types of initiators: Type I initiator and Type II initiator. A Type I initiator is a holon that initiates the actions to identify and remove holons from HMS whereas a Type 2 initiator is a holon that handles dynamic addition of holons to HMS. The reconfiguration process triggered by Type I initiator is broken down into two stages: (1) initiator identification & standby notification and (2) process optimization. As the reconfiguration process for Type I initiators is applicable to Type II initiators with the exception of standby notification step, we focus on the reconfiguration process for Type I initiators only. A condition for the existence of an optimal solution to HRP is established. Our study in this paper paves way for the development of an effective HMS reconfiguration mechanism with predictable performance.

REFERENCES

Balasubramanian S, Brennan R.W., Norrie D.H. (2001), An architecture for metamorphic control of holonic manufacturing systems. *Computers in Industry*, **46**, 13-31.

Brennan, R., M. Fletcher and D. Norrie, (2002) An agent-based approach to re-configuration of real-time distributed control systems, *IEEE Transactions in Robotics and Automation* **18** (4), pp. 444–451.

Christensen J. (1994), Holonic Manufacturing systems – initial architecture and standards directions, in: *Proceedings of the First European Conference on Holonic Manufacturing Systems*, Hannover, December.

Deen, S.M. Editor, (2003) *Agent-Based Manufacturing: Advances in the Holonic Approach*, Springer-Verlag, Berlin, Heidelberg.

Hsieh, F. S. (2006) Analysis of contract net in multi-agent systems, Automatica, vol. **42**. no. 5, p. 733-740.

Hsieh, F. S. (2008) Formation and Optimization in Holonic Manufacturing Systems with Contract net, *Automatica*, Accepted.

Koestler A., (1967) *The Ghost in the Machine*. Hutchinson, London.

Leitão, P. and F. Restivo, (2006) "ADACOR: A holonic architecture for agile and adaptive manufacturing control," Computers in Industry, Volume 57, Issue 2, p. 121-130.

Leitão, P., A.W. Colombo and F. Restivo, (2003) An approach for the formal specification of holonic control systems. Lecture Notes in Artificial Intelligence Vol. 2744, Springer-Verlag pp. 59–70.

Marik, V., (2004) Industrial applications of the agent-based technology, *Proceedings of the INCOM'04 International Conference* Salvador Bahia, Brazil.

Murata, T. (1989) "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, No. 4, April

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.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. and Peeters, P., (1998) Reference architecture for holonic manufacturing systems : PROSA, *Computers in Industry*, **37**, 255-274.

Wyns J., (1999) *Reference Architecture for Holonic Manufacturing*, PhD Dissertation, PMA Division, Katholieke Universiteit, Leuven, Belgium.