Dynamic Service Composition of Holonic Virtual Enterprise Processes in Multi-agent Systems

Fu-Shiung Hsieh[#1], Chih Yi Chiang[*2]

[#] *Department of Computer Science and Information Engineering, Chaoyang University of Technology*
Taiwan, R.O.C.
[1]`fshsieh@cyut.edu.tw`

[*] *Department of Computer Science and Information Engineering, Chaoyang University of Technology*
Taiwan, R.O.C.
[2]`s9627605@cyut.edu.tw`

***Abstract*— Composition of processes for virtual enterprises is a significant issue. Although the flexible architecture provided by holonic manufacturing system (HMS) paves the way for adaptive composition of virtual enterprise processes, it also poses challenges in planning processes for virtual enterprises. The challenges are due, in part, to the loosely coupled structure of holons and also to the complex interactions among holons. Development of new methodologies is required to optimize the holonic processes in HMS to achieve the objectives. In this paper, we concentrate on the development of method for the composition of holonic enterprises processes. We consider the holonic processes composition (HPC) problem to synthesize processes with minimal costs while meeting the timing constraints in HMS. We propose a two-layer contract net protocol to find the minimal cost solution. We implement a problem solving environment to solve HPC. We illustrate how to apply a multi-layer contract net protocol to find the optimal solution by using an example.**

***Keywords*— Contract net, holonic, multi-agent system, Petri net, workflow, virtual enterprise.**

1. INTRODUCTION

The manufacturing sector has been facing major challenges as it undergoes revolutionary changes fuelled by new and sophisticated demands from customers, global competition, distribution of manufacturing and marketing activities and technological advances [1]. Due to the dynamically changing characteristics of manufacturing environments, static control structures are not suitable anymore. Bousbia and Trentesaux reviewed state-of-the-art and future trends of self-organization in distributed manufacturing control [6]. The increasing versatility in product demands calls for a system architecture able to evolve in time [25]. Holonic manufacturing systems (HMS) [2], [27], [5], [7] provide a reconfigurable, flexible and decentralized manufacturing environment to accommodate changes and meet customers' requirements dynamically based on the notion of holon [16], an autonomous, co-operative and intelligent entity able to collaborate with other holons to process the tasks.

In HMS, a system of holons that can autonomously cooperate to achieve a goal forms a holarchy. A challenge is to design a mechanism to guide the holons such that the decisions made by the individual holons as a whole composite a holarchy that achieves the objectives such as meeting customers' demands and due dates. In existing literature, there are many studies on HMS [20], [10], [23], [4], [26], [9], [22], [18], [12], [17], [13]. There are many works on planning and optimization in HMS. Giebels, Kals and Zijm [26], [9] proposed the system architecture of a flexible manufacturing planning and control system. Existing works on how to specify formally the dynamic behaviour of holonic systems appear in [18], [14], [12]. In [15], the authors present an approach based on agent negotiation with an extended contracting protocol for supporting logistics and production planning. Studies on how should the production control structure evolve to adapt to changes have been made by [17]. Existing results on how to achieve global optimization in decentralized systems can be found in [13].

One approach to overcome the limitations of classical scheduling is the use of distributed schemes such as agent or holonic-based control architectures [3]. Babiceanu, Chen & Sturges presents a solution for scheduling using the holonic control approach [3] in which a feasible solution emerges from the combination of individual material handling holons' solutions. Leitão and Restivo [19] presents a holonic approach to manufacturing scheduling, where the scheduling functions are distributed by several entities, combining their calculation power and local optimization capability.

A holonic process is dynamically formed by a set of product holons and resource holons based on a certain task distribution protocol such as contract net protocol (CNP) to execute a task. CNP [24] is a well known protocol for distributing tasks. Application of CNP for task allocation in HMS is found in [22], [10], [23]. Formation of holonic processes in HMS based on CNP has been studied in [13], [12], [11], where Petri net [21] models have been proposed to capture the interactions between resource holons and product holons. These results pave the way for the development of methodology for composing holonic processes that satisfy timing constraints. To achieve the objectives, we first formulate a holonic process composition problem based on Petri nets and propose a method to find an optimal solution.

The remainder of this paper is organized as follows. In section 2, we describe and state the holonic processes composition problem (HPC). In section 3, we propose a two-layer contract net protocol for solving HPC. In section 4, we first detail our design using UML activity diagrams to describe handling of the two layer protocol messages. We then demonstrate our implementation using an example. Section 5 concludes this paper.

2. HOLONIC SYSTEM OPERATION

An HMS consists of three types of holons: resource holons, product holons and order holons [27]. A resource holon consists of production resources with relevant components to control the resources. 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 composite holon called a holarchy.

Figure 1 illustrates a scenario in which a production process is to be formed in HMS with seven product holons $h_1, h_2, ..., h_7$ and ten resource holons $r_1, r_2, ..., r_{10}$ to accomplish a task with timing constraints. Holonic processes are production processes dynamically created based on the collaboration of product holons. Each product holon has an internal process flow. Execution of the internal process of a product holon may rely on the outputs from the internal processes of one or more upstream product holons. For example, product holon h_5 and h_6 depends on either h_1 or h_2 to provide the type-one parts and also depends on either h_3 or h_4 to provide the type-two parts. Product holon h_7 depends on h_5 to provide the type-three parts and also depends on h_6 to provide the type-four parts.

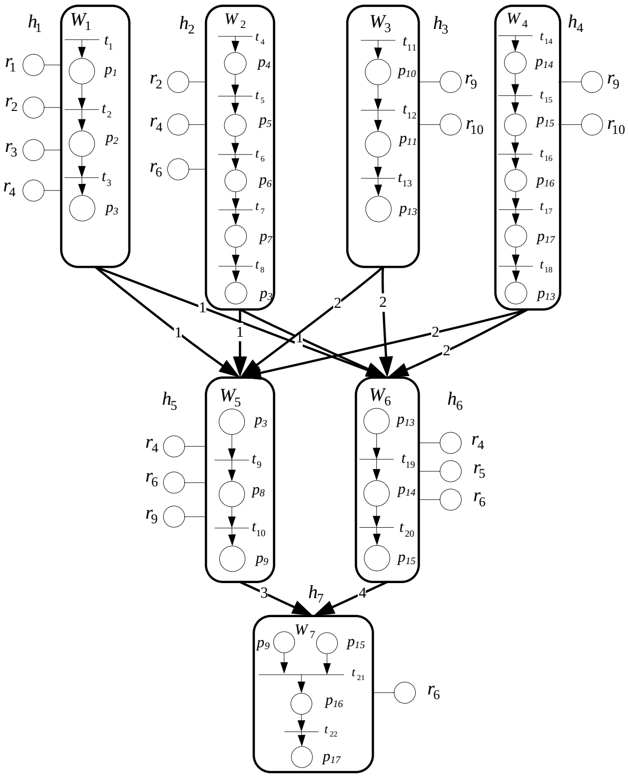


Fig. 1 Holonic process formation

Execution of a task τ requires collaboration of a set of product holons and resource holons. To process a task, the internal processes of a minimal set of product holons need to be connected. One way to form a holarchy is based on the contract net protocol (CNP). Formation of holonic processes in HMS based on CNP has been studied in [13], where CNP is applied to form a holarchy base on establishment of contracts between a set of product holon and a set of resource holons. In this paper, we consider the composition of minimal cost holonic processes that execute a task τ while satisfying timing constraints.

To describe the dependency between product holons in forming a holarchy for executing a task τ, we use a circle to represent a product holon. To describe the inputs or outputs of a product holon, we use a set of circles called interface nodes. The output interface node of a product holon may connect to the input interface node of another product holon. The dependency between product holons is represented by a digraph $D(N, E)$.

Definition 2.1: The dependency between product holons is described by a digraph $D(N, E)$, where $N = H \cup I$ is a set of nodes, E is a set of arcs, a node in H denotes a holon and a node in I denotes an interface between holons. An arc connecting node $h \in H$ to node $i \in I$ means that holon h will provides its outputs to other holons via interface node i. An arc connecting node $i \in I$ to node $h \in H$ means that holon h may accept the outputs from other holons via interface node i.

In a given digraph $D(N, E)$, the number of outgoing arcs of a holon node h is called the out-degree of h. A product holon node h with zero out-degree is called a final product holon. We use $Out(h)$ and $In(h)$ to denote the set of output interface nodes and the set of input interface nodes of product holon h, respectively.

Definition 2.2: A collaborative network for product holon h is described by a digraph $C(N_C, E_C) \subseteq D(N, E)$, where N_C is a finite set of nodes and E_C is a finite set of arcs. $N_C = H_C \cup I_C$, $H_C \subseteq H$, $I_C \subseteq I$, $h \in H_C$ and for each $h' \in H_C$, there exists $h'' \in H_C$ connecting to i for each $i \in In(h)$. If there is exactly one incoming arc and one outgoing arc for each interface node in I_C, $C(N_C, E_C)$ is called a minimal collaborative network for product holon h.

In general, the number of minimal collaborative networks that can perform a given task grows exponentially with the size of $D(N, E)$. The existence of multiple minimal collaborative networks for a task poses an optimization issue. An important problem is to minimize the overall cost while forming a minimal collaborative network to perform a task with a timing constraint. Let ω denote the timing constraints. Let w_C denote the cost of $C(N_C, E_C)$. The holonic processes composition problem can be stated informally as follows:

Holonic Processes Composition (HPC) Problem:

$$\min_{\substack{s.t.\, C(N_C,E_C)\subseteq E(N,E) \\ C(N_C,E_C)\, satifies\ \omega}} w_C$$

A challenge in the development of a solution methodology to HPC problem is due to the distributed computing environment of holonic systems. An effective solution method must be compliant with the holonic architecture. The solution approach proposed in this paper is based on interactions of holons in cooperative, distributed architecture that exhibit the characteristics of HMS. To describe the details of our solution method for the HPC Problem, a mathematical formulation is required. To capture the interactions of holons through the underlying workflows and analyzing the relevant timing requirements, we propose timed Petri net models.

3. TWO-LAYER CONTRACT NET PROTOCOL

One way to form a holarchy is based on the contract net protocol (CNP). Interactions among resource holons, product holons and order holons are through the well-known contract net protocol [24]. In contract net protocol, there are two roles an agent can play: manager or bidder. Four stages are involved to establish a contract between a manager and one or more bidders: (1) Request for tender: The manager announces a task to all potential bidders. The announcement contains the 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 proposals, the manager awards the contract to the best bidder. (4) Establishment of contract: The awarded bidder may either commit itself to carry out the task or refuse to accept the contract by sending messages to the manager. For the latter case, the manager will reevaluate the bids and award the contract(s) to another bidder(s).

Formation of holonic processes in HMS based on CNP has been studied in [13], where CNP is applied to form a holarchy base on establishment of contracts between a set of product holons and a set of resource holons. HMS provides a flexible architecture to configure product holons to fulfill the order requirements. We propose an extended contract net protocol that consists of an upper layer and a lower layer protocols. The upper layer contract net protocol is applied by an order holon and the potential product holons to find the minimal cost product holons that form a holarchy.

After conducting the upper layer protocol, a holarchy is formed by a set of product holons. To acquire the resources required to execute the operations in the holarchy, the lower layer contract net protocol is applied between the holarchy and the potential resource holons to analyze the feasibility and determine the resource holons. Figure 1 illustrates the two-layer contract net protocol.

Resource holons, product holons, and order holons are assigned different roles at different phases in the two-layer protocol. To describe the negotiation processes between an order holon, the product holons and the resource holons in HMS, the following messages are defined.

Definition 2.1: X_{rft} denotes a “Request for tenders” message sent by either an order holon or a product holon. X_{sop} denotes the “Submission of proposals” message sent by either a product holon or a resource holon. X_{aoc} denotes the "Awarding of contract" message sent by an order holon or a product holon. X_{eoc} denotes the "Establishment of contract" message sent by an awarded product holon or an awarded resource holon.

TABLE 1
THE INPUT TYPE(S) AND OUTPUT TYPE(S) FOR PRODUCT HOLONS $h_1, h_2, ..., h_7$

Product holon	Input type	Output type
h_1		Product Type 1
h_2		Product Type 1
h_3		Product Type 2
h_4		Product Type 2
h_5	Product Type 1	Product Type 3
h_6	Product Type 2	Product Type 4
h_7	Product Type3, 4	Product Type 5

Example:Figure 2(a)~(d) illustrate a scenario in which a production process is to be formed in HMS based on the upper layer contract net protocol applied between the potential product holons $h_1, h_2, ..., h_7$ to fulfill the requirements of an order holon h_1 . Holonic processes are production processes dynamically created based on the collaboration of product holons. Each product holon has an internal process flow, the required input types and output types. Table 1 shows the input type(s) and output type(s) for product holons $h_1, h_2, ..., h_7$. Execution of the internal process of a product holon may rely on the outputs from one or more upstream product holons to produce the outputs. For example, product holon h_7 depends on either h_5 or h_6 to provide the required parts.

Furthermore, h_5 also depends on either h_1 or h_2 to provide the required parts whereas h_6 depends on either h_3 or h_4 to provide the required parts. To minimize the costs in the upper layer protocol, each product holon finds the minimal cost upstream product holons by applying the minimal cost holons determination algorithm (MCHDA). MCHDA will be detailed in Section 3. Suppose the cost of h_1 is less than that of h_2 and the cost of h_3 is less than that of h_4 . Figure 2(e) shows the holarchy H formed resulting from the upper layer protocol. The cost of a product holon is obtained based on the costs of the operations.

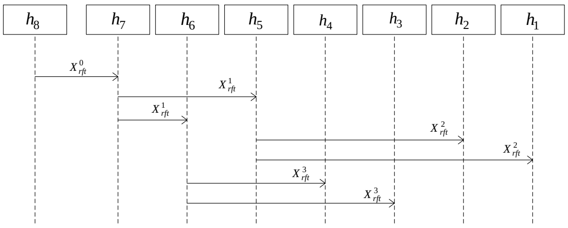


Fig. 2 (a) Request for tender

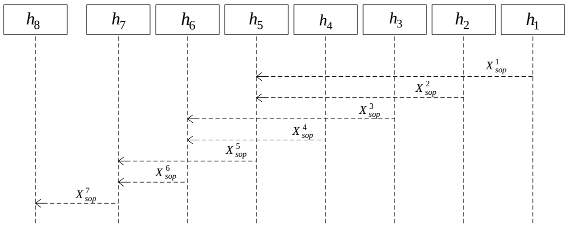


Fig. 2 (b) Submission of proposals

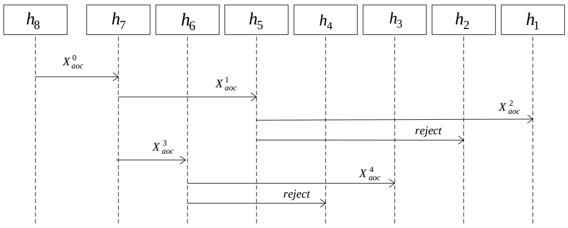


Fig. 2 (c) Awarding of contracts

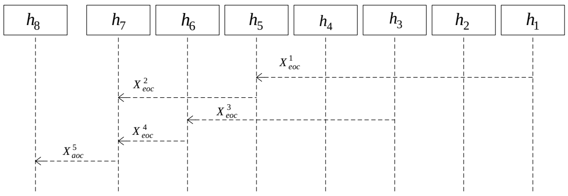


Fig. 2 (d) Establishment of contracts

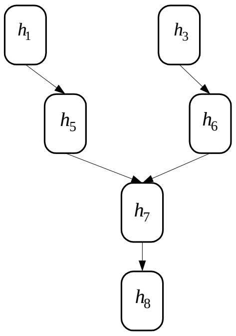


Fig. 2 (e) A holarchy H formed

The upper layer specifies the interactions between an order holon and the potential product holons. The order holons always act as managers to initiate tender processes with the product holons as the bidders. The product holons act as the managers in turn to issue a new request for bids to other product holons for the set of subtasks stemming from the product requirements.

The lower layer specifies the interactions between the holarchy formed by combining the product holons obtained from the upper layer protocol and the potential resource holons to analyze the feasibility of the solution. Figure 3(a)~(d) illustrate a scenario in which the lower layer contract net protocol is applied between the holarchy H and resource holons $r_1, r_2, ..., r_6$.

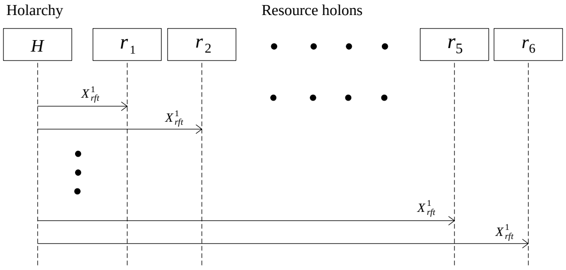


Fig. 3 (a) Request for tender

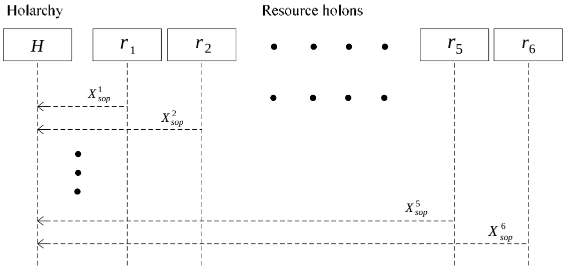


Fig. 3 (b) Submission of proposals

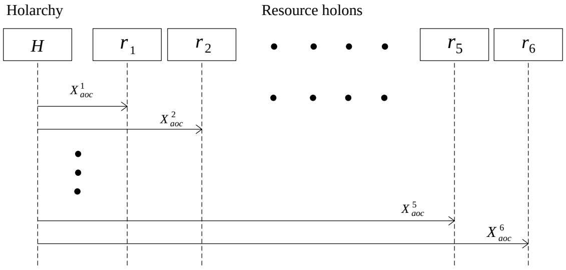


Fig. 3 (c) Awarding of contracts

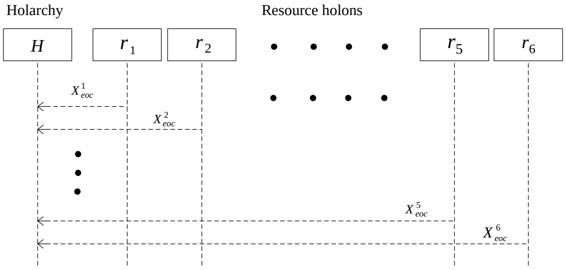


Fig. 3 (d) Establishment of contracts

4. DESIGN AND IMPLEMENTATION

To realize the proposed idea, we implement a holonic system based on the two-layer contract net protocol. In this section, we first present the activity diagrams for handling the two-layer contract net protocol messages and then illustrate the developed system. The activity diagram for handling a "Request for tender" message X_{rft} is shown in Fig. 4(a).

In handling a X_{rft} message, a holon first checks whether the product type is supported. If it is supported, the holon applies to check whether the timing constraint is satisfied. If timing constraint is satisfied, the holon will encode a X_{rft} message, forward the X_{rft} message and wait for the X_{sop} message from the bidders (upstream holons). The holon will wait for all the arriving X_{sop} messages until a timer expires. All X_{sop} messages arriving before expiration of the timer are decoded to determine the bidders with minimal cost. Based on the winners, the holon encodes its proposal and sends its X_{sop} message to the manager.

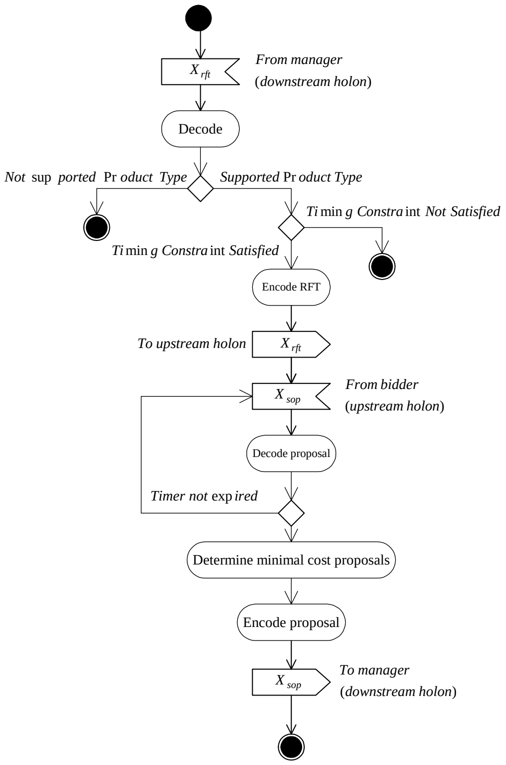


Fig. 4(a) Activity diagram for handling a "Request for tender" message in the upper layer

Fig. 2(b) shows how to handle "Submission of proposal” messages X_{sop} from the bidders. On receiving “Submission of proposal” messages the bidder, a holon must decode the proposals to determine the winners. The holon then encodes the contract awarding message X_{aoc} and responds by sending X_{aoc} to the corresponding winners.

To implement a HMS based on the methodology proposed in this paper, the four types of messages, including “Request for tender” X_{rft}, “Submission of proposal” X_{sop}, “Awarding of contract” X_{aoc} and “Establishment of contract” X_{eoc}, used in the two-layer contract net protocol must be specified. In the upper layer contract net protocol, a “Request for tenders” message X_{rft} sent by an order holon includes product type, due date and also the order holon ID. An example is the message X_{rft}^{0} issued by order holon h_8 in Fig. 2(a). A “Request for tenders” message X_{rft} sent by a product holon to the potential upstream product holons includes product type and also the product holon ID. The messages X_{rft}^{1}, X_{rft}^{2} and X_{rft}^{3} in Fig. 2(a) are examples of “Request for tenders” messages sent by product holons. On receiving a message X_{rft} sent by an order holon or a product holon, a potential product holon sends “Submission of proposals” message X_{sop}. As Petri net models are used to specify the internal process of holons, the proposal submitted by a holon is represented by a Petri net model. Petri Net Markup Language (PNML) [28] is a standard for the representation of Petri nets. Therefore, we adopt PNML to represent the Petri net models of holons. A X_{sop} message issued by a product holon consists of the URL of the workflow Petri net model in PNML, the product holon ID and the product type produced by the holon. For the example in Fig. 2(b), X_{sop}^{1}, X_{sop}^{2}, X_{sop}^{3}, X_{sop}^{4}, X_{sop}^{5}, X_{sop}^{6} and X_{sop}^{7} are examples of “Submission of proposals” messages sent by product holons. On receiving X_{sop} messages sent by the potential product holons, an order holon or product holon must determine the winners based on the optimization algorithm and send the “Awarding of contracts” X_{aoc} messages to the best product holons. In Fig. 2(c), X_{aoc}^{0}, X_{aoc}^{1}, X_{aoc}^{2}, X_{aoc}^{3} and X_{aoc}^{4} are examples of “Awarding of contracts” messages sent by order holon or product holons. On receiving the X_{aoc} message, a winner responds by issuing X_{eoc} message back to the manager. For Fig. 2(d), X_{eoc}^{1}, X_{eoc}^{2}, X_{eoc}^{3}, X_{eoc}^{4} and X_{eoc}^{5} are examples of “Establishment of contracts” messages.

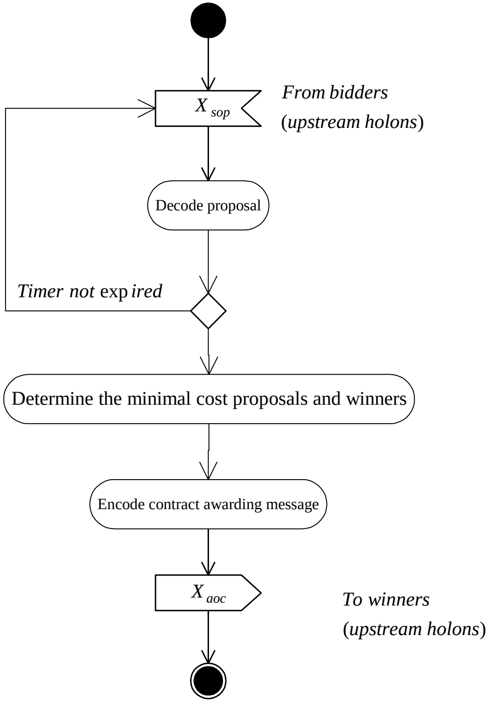


Fig. 4(b) Activity diagram for handling a “Submission of proposal” message in the upper layer

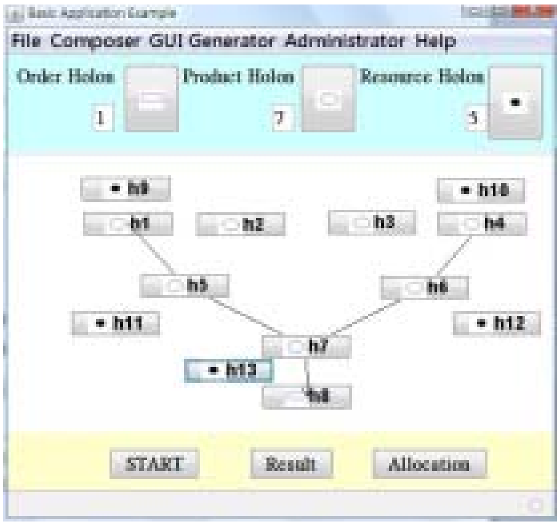
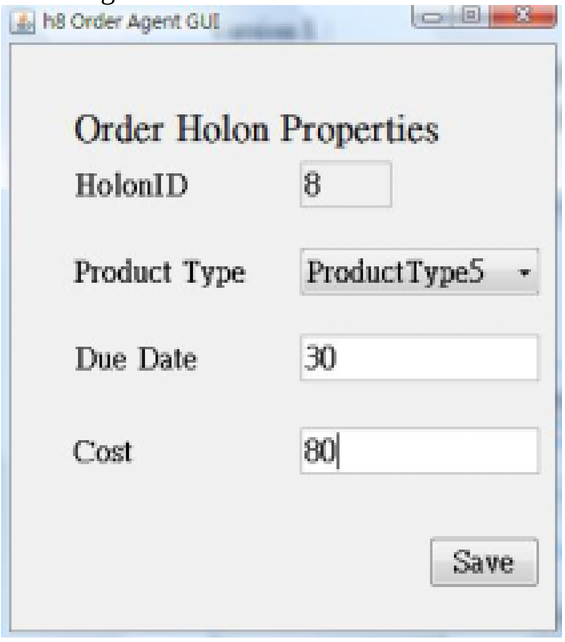


Fig. 5 HMS Editor/Monitor

We develop a HMS Designer based on the two-layer contract net protocol proposed in this paper. We illustrate the function of HMS Designer by applying it to an application scenario. HMS Designer is developed based on the Petri net models of holons that we proposed in Section 3, the services publication/discovery mechanism in

Section 4 and the message handling activity diagrams proposed in Section 5. Fig. 5~ Fig. 8 shows the screen shots of our system, which consists of a HMS Editor/Monitor (Fig. 5) and graphical user interface to set the properties of order holons, product holons and resource holons (Fig. 6~ Fig. 8) and display the established contracts between order holons, product holons and resource holons based on our proposed two-layer contract net protocol. The following example details an application scenario of the HMS Designer.



Order Holon Properties

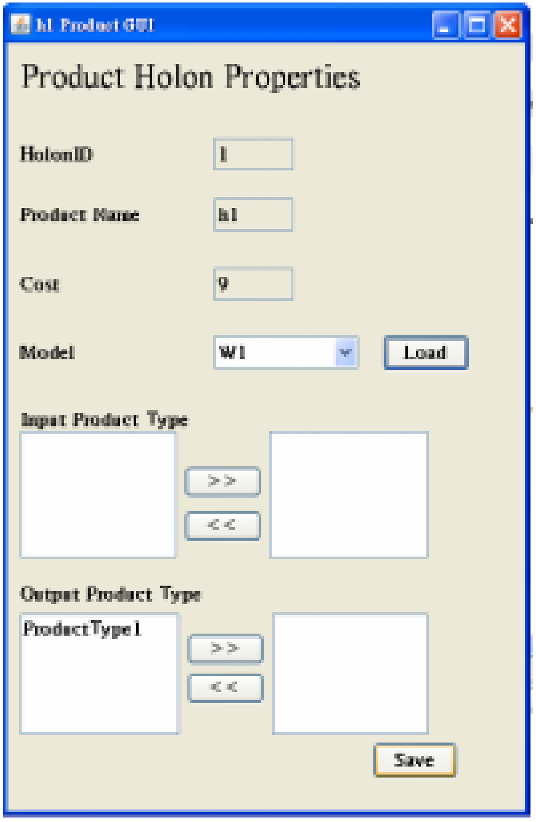
HolonID 8

Product Type ProductType5

Due Date 30

Cost 80

Fig. 6 Properties of an Order Holon



Product Holon Properties

HolonID 1

Product Name h1

Cost 9

Model W1

Input Product Type

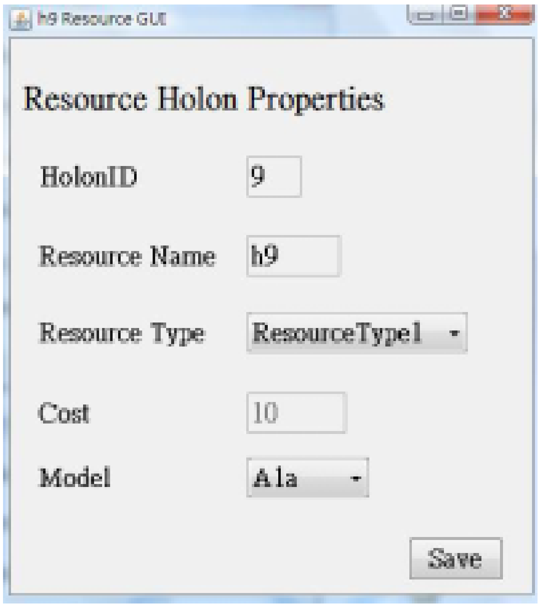
Output Product Type

ProductType1

Fig. 7 Properties of a Product Holon

Example: Consider the requirements of an order with requested product type: Product Type 5, order due date: 30, and cost constraint: 80.

Fig. 5 demonstrates the HMS Editor in which users (order holon managers, product holon managers and resource holon managers) may define the holons in a system. User may drag and drop the order holon, product holon and resource holon icons to define the holons. For this example, there are thirteen holons ($h_1 \sim h_{13}$) that have been defined, including seven product holons and one order holon. For clarity, we only show the product holons $h_1 \sim h_7$ and order holon h_8 while hiding resources holons $h_9 \sim h_{13}$ in Fig. 5. Fig. 6 shows the graphical user interface for order holon managers to set order holon properties. The properties of an order holon include holon id, product type, quantity, due date and cost constraint. In Fig. 6, the product type, due date and cost of order holon h_8 is set to Product Type 5, 5, 30 and 80, respectively. Fig. 7 illustrates the graphical user interface for product holon managers to set product holon properties. The properties of a product holon include holon id, model (represented by Petri net in PNML format), input/output product types and costs, where input product types is the product types required for producing the output product types. Fig. 8 illustrates the graphical user interface for resource holon managers to set resource holon properties. The contracts between product holons are shown in Fig. 5.



Resource Holon Properties

HolonID 9

Resource Name h9

Resource Type ResourceType1

Cost 10

Model A1a

Fig. 8 Properties of a Resource Holon

5. CONCLUSIONS

To meet the changing customers' demands with minimal costs while delivering the products timely, manufacturers rely on a flexible, effective architecture to dynamically collaborate with the partners to fulfill the orders' requirements, optimize the costs and reduce the risk. This paper presents a systematic methodology to synthesize minimal cost production processes that satisfy timing constraints based on holonic system architecture. A holonic process is dynamically formed by a set of product holons and resource holons to execute a task. We consider the problem to synthesize holonic processes with minimal costs while meeting the timing constraints in holonic systems. We formulate this problem based on a hybrid model in which contract net protocol is adopted as the negotiation protocol and Petri net is used to specify and analyze the timing and resource constraints. To specify the costs of operations, we augment the Petri net with a cost function. We formulate an optimization problem to minimize the cost while meeting the timing constraints. Our method for solving the optimization is broken down into two parts: a method to find the collaborative workflows that satisfy timing constraints and a method to find the minimal cost solution. All holons that cannot meet the timing requirements are excluded. We also establish a simple condition to test whether there exists an optimal solution based on the structure of the collaborative workflow. We illustrate how to apply a multi-layer contract net protocol to find the optimal solution by using an example. As our methodology is generic and developed based on holonic system architecture, it can be tailored for specific HMS applications.

ACKNOWLEDGMENT

This paper is supported in part by National Science Council of Taiwan under Grant NSC97-2410-H-324-017-MY3.

REFERENCES

[1] Agility Forum. Next Generation Manufacturing Project: "A Framework for Action." Agility Forum, Leaders for Manufacturing, Technologies Enabling Agile Manufacturing, 1997.

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

[3] Babiceanu, R. F., Chen, F. F. and Sturges, R. H., "Real-time holonic scheduling of material handling operations in a dynamic manufacturing environment," Robotics and Computer-Integrated Manufacturing, Volume 21, Issues 4-5, August-October, Pages 328-337, 2005.

[4] Brennan, R. W. and Norrie, D. H., "Evaluating the performance of reactive control architectures for manufacturing production control." Computers in Industry, 46, 235–245, 2001.

[5] Brussel H.V. , Wyns J., Valckenaers P., Bongaerts L. and Peeters P., "Reference architecture for holonic manufacturing systems : PROSA," *Computers in Industry*, 37, 255-274, 1998.

[6] Bousbia, S. and Trentesaux, D., "Self-organization in distributed manufacturing control: state-of-the-art and future trends," Proceedings of 2002 IEEE International Conference on Systems, Man and Cybernetics, Volume 5, 6-9 Oct. Page(s):6 pp. vol.5, 2002.

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

[8] Duffie, N. A. "Synthesis of heterarchical manufacturing systems." Computers in Industry, 14, 167–174, 1990.

[9] M. M. T. Giebels, H. J. J. Kals and W. H. M. Zijm, "Building holarchies for concurrent manufacturing planning and control in EtoPlan," Computers in Industry, Volume 46, Issue 3, October, Pages 301-314, 2001.

[10] Gou L, Luh, P. B. Kyoya Y., "Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation," *Computers in Industry* 37 pp. 213-231, 1998.

[11] Hsieh, F. S. "Model and Control Holonic Manufacturing Systems Based On Fusion Of Contract Nets And Petri Nets", Automatica, Vol. 40, pp. 51-57, 2004.

[12] Hsieh, F. S. "Analysis of contract net in multi-agent systems," Automatica, vol. 42. no. 5., p. 733-740, 2006.

[13] Hsieh, F. S. "Holarchy Formation and Optimization in Holonic Manufacturing

Systems with Contract net," Automatica, vol. 44. no. 4., p. 959-970, 2008.

[14] Johnson, C. A. "Towards a formalized HMS model," in Agent-Based Manufacturing: Advances in the Holonic Approach, S. M. Deen (ed.), Springer-Verlag, Berlin, 193–206, 2003.

[15] Karageorgos, A., Mehandjiev, N., Weichhart, G. andHämmerle, A., "Agent-based optimisation of logistics and production planning," Engineering Applications of Artificial Intelligence, Volume 16, Issue 4, June, Pages 335-348, 2003.

[16] Koestler A., "The Ghost in the Machine." Hutchinson, London, 1967.

[17] Leitão, P. and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," Computers in Industry, Volume 57, Issue 2, February, Pages 121-130, 2006.

[18] Leitão, P., A.W. Colombo and F. Restivo, "An approach for the formal specification of holonic control systems." In: V. Marík, D. McFarlane and P. Valckenaers, Editors, Holonic and Multi-Agent Systems for Manufacturing, Lecture Notes in Artificial Intelligence Vol. 2744, Springer-Verlag pp. 59–70, 2003.

[19] Leitão, P. and Restivo, F., "A holonic approach to dynamic manufacturing scheduling," Robotics and Computer-Integrated Manufacturing, In Press, Corrected Proof, Available online 3 December, 2007.

[20] McFarlane, D. C. and Bussmann. S. "Developments in holonic production planning and control." International Journal of Production Planning and Control, 11(6), 522–536, 2000.

[21] Murata T., "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE*, vol. 77, No. 4, pp. 541-580, 1989.

[22] Neligwa, T. and Fletcher, M. "An HMS operational model," in Agent-Based Manufacturing: Advances in the Holonic Approach, S. M. Deen (ed.), Springer-Verlag, Berlin, 163–191, 2003.

[23] Ramos, C. "A Holonic Approach for task scheduling in manufacturing systems." Proceedings of the 1996 IEEE International Conference on Robotics and Automation, 2511–2516, 1996.

[24] Smith, R.G. "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Trans. On Computers* Vol. 29, pp. 1104-1113, 1980.

[25] Trentesaux, D. and Tahon, C., "DPACS: a self-adaptive production activity control structure," Proceedings of 1995 INRIA/IEEE Symposium on Emerging Technologies and Factory Automation, 1995. ETFA '95, Volume 2, 10-13 Oct. vol.2, pp.543 – 551, 1995.

[26] G. Wullink, M. M. T. Giebels and H. J. J. Kals, "A system architecture for holonic manufacturing planning and control (EtoPlan)," Robotics and Computer-Integrated Manufacturing, Volume 18, Issues 3-4, June-August, Pages 313-318, 2002.

[27] Wyns J., "*Reference Architecture for Holonic Manufacturing*," PhD Dissertation, PMA Division, Katholieke Universiteit, Leuven, Belgium, 1999.

[28] Michael Weber, and Ekkart Kindler, "The Petri Net Markup Language," http://www2.informatik.hu-berlin.de/top/pnml/download/about/PNML_LNCS.pdf., 2002.

[29] Hsieh, F. S. "Dynamic composition of holonic processes to satisfy timing constraints with minimal costs," To appear on Engineering Applications of Artificial Intelligence, doi:10.1016/j.engappai.2008.12.002, 2009.