The Journal of Korean
Institute of CALS/EC
Vol. 1, No. 1, Aug. 1996

통합생산환경에서의 가상공장 시뮬레이터 개발을 위한 제어모형

박남규, 이현정 *

A Control Model for Prototyping Virtual Factory Simulator in Computer Integrated Manufacturing Environment

Namkyu Park, Hyun Jung Lee

ABSTRACT

Presented in this paper is a control model for developing virtual factory simulator, which is being operated under the distributed environment. The control model consists of production activity plan and information flows. To incorporate elements of the characteristics of the distributed control system, we suggested a collaboration model. This model is working under the client/server architecture, and also designed for cooperative-distributed shop control(CDSC) system in order to exploit several advantages of client/server architecture. Collaboration among each agent(or client) is done through negotiation and task sharing. Based on a contract net model, the CDSC system has three kinds of agents - order agent, resource agent, and communication forwarding agent. Each agent performs shop scheduling and control through negotiation on contract net. No node in CDSC system can have authority over other node. A bidding scheme is employed for negotiation between order agent and resource agents. The CDSC system can support re-negotiation among resource agents and an algorithm for re-negotiation is also developed. Experimental results are shown to advocate the effectiveness of the CDSC system for CIM environments.

---

* 생산기술연구원

1. Introduction

In recent years, there is a new thrust in manufacturing in which the manufacturing philosophy places great emphasis on much-increased consumer's needs and shorter life cycle of products. Under this philosophy, growing attention has been given to the application of computer-based technolbgy under the name of computer integrated manufactúring (CIM). CIM system consists of sensors, equipment, software and communication network. It is not confined to the physical factory, but embraces a total business system of enterprise. In CIM, manufacturing operations and activities are automated by using numerically controlled machines and industrial robots. This results in "islands of automation" which hinders consistent management of manufacturing systems as a whole. Modern manufacturing systems should overcome this isolation by bridging the gap between islands with the aids of computer communication technology. From the perspectives of decision making, CIM is an advent of new concepts spanning from forecasting, through manufacturing, to sales and after-sales services. In this regard, information processing plays a key role to implement CIM effectively. Moreover, shop floor is a place where products are made and all information related with manufacturing occur. Hence, efficient shop floor control is fundamental factor to achieve sound CIM. The shop floor control problem under CIM environments has different characteristics from that of conventional job shops. They are as follows.

- real-time requirement to propagate occurrence(s) of an event of a manufacturing site to another site
- logically integrated, but physically distributed system
- interface problem due to the heterogeneity of system components
- different levels of local intelligence
- requirement of adaptive control ability in decision making

In this research, we addressed a collaboration model for cooperative-distributed shop control at operational level to deal with above facts. We used distributed artificial intelligence concept. In what follows, we briefly review the related research found in the literature, and then describe and discuss the architecture of the system developed.

2. Related Research

YAMS[Parunak, 1986] is a distributed scheduling system adopting a contract net model. It is an object-oriented system with hierarchical structure. A work cell lies in upper level of the hierarchy and a workstation lies in lower level. A work cell is committed to supervise and control factory. A protocol for communication and information exchange among workstation or each work cell was developed using contract net of Davis and Smith[Davis, 1983]. Shaw[Shaw, 1987] presented a distributed-dynamic scheduling system for a cellular manufacturing system equipped with local area network (LAN). He also used a contract net model for network-wide bidding. Shaw modeled network-wide bidding scheme by an augmented Petri-net and carried out simulation. His simulation result shows that distributed scheduling approach works better than traditional scheduling approach. CSS was presented in the literature by Ow et.al.[Ow, 1988]. CSS is also a contract net-based model. It is composed of work order manager and resource broker. Most of research using contract net and negotiation overlooked resource re-negotiation or the problem of information exchange nevertheless. Upton and Barash[Upton, 1991] tried to analyze a distributed system using queueing network model. In a broader sense, it is not far from conventional queueing network because it is also a kind of simulator to predict system performance. Nof et. al.[Nof, 1989] [Nof, 1992] suggested a decision making model to control job that is doing by cooperation and competition when multi-robots sharing a space are working together. This research related to the control of a cell element rather than cell control itself. In Korea, Chung[Chung, 1994] investigated the problem of distributed shop floor control in flexible manufacturing system. He adopted original contract net model as a decision making framework. There is another approach in using shop status information for shop control. MPECS (multi pass expert control system) was suggested by Wu and Wysk[Wu, 1987]. MPECS extends conventional job scheduling approach in terms that it could include shop status information every Δt by partitioning scheduling period into appropriate time interval Δt. Cho and Wysk[Cho, 1993] also suggested a variant of MPECS which improved prediction capability on strategy selection for the next pass using neural networks. However, these approaches can't also cope with the effect of dynamic situation of shop floor because of static and passive nature of the systems in strategy selection. A good evidence for this claim is that determination of Δt affects system performance severely.

Although there are many distributed models, most of them are incomplete in their functionality. This is mainly due to the lack of mechanism to treat shop status information. In this research, we proposed a system which could reflect more information during the course of problem solving.

3. Distributed System Modeling by Contract-Net

Many of currently available scheduling systems for CIM do not reflect properties of distributed systems sufficiently. As mentioned in section 1, as a company imports CIM technology more and more, the local intelligence of each node should become greater likewise. Also as the technology advances, each site of distributed systems has grown to be considered as an independent system as far as information processing capacity and computing power are concerned. Accordingly, the projected trend for envisioned CIM system would be a loosely-coupled distributed system (LCDS). Each node in LCDS has a unique input and output, that is self-regulating input and output. And all nodes can access other nodes and share resources residing on other nodes through factory-wide LAN. Davis and Smith[Davis, 1983] proposed a "*contract net*" model as a new methodology for distributed problem solving. A contract net comprises nodes and messages. Each node communicates with each other through message exchange. They formulate message exchange and control signal flow in contract net by the notion of "negotiation". By negotiation, we mean that a solution to distributed system can be acquired by "cooperation among nodes".

In defining cooperation among each nodes, we must specify two facts, namely how to communicate with each other and what information should be exchanged. Negotiation is a formal specification to above facts. A contract net comprises 3 types of nodes. They are manager nodes, bidder nodes and contractor nodes. A manager node issues a task to be processed. A bidder node is a node, with a certain cost, for processing a task issued by the manager node. And a contract node is a selected node from the bidder nodes. A problem solving for distributed system is isomorphic to the cooperation among 3 types of nodes. In CIM system, a problem solving process for distributed shop floor control is directly analogous to this negotiation scheme. Each local intelligence corresponds to the node of contract net. Work order relates to manager node, machining centers and AS/RS to bidder node, and work-loaded resource to contractor node respectively.

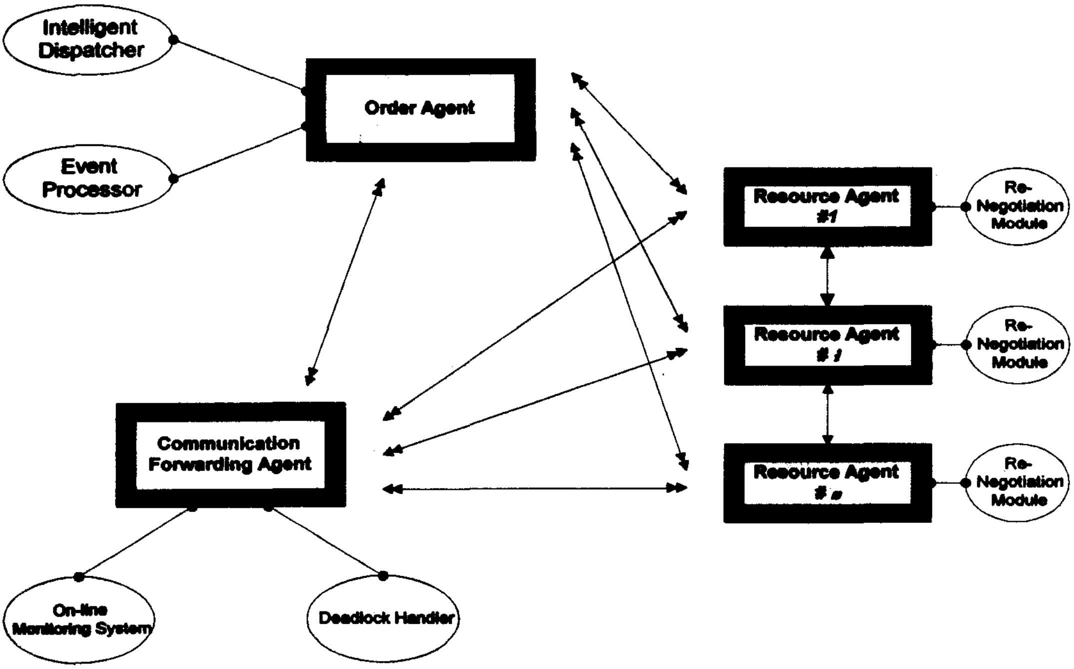

Figure 1. The CDSC system

4. Architecture of the CDSC system

In this section, we propose a distributed shop floor control system with multi-agents. The proposed system called CDSC (cooperative-distributed shop control) consists of 3 kinds of agents - order agent (OA), communication forwarding agent (CFA), and resource agent (RA). Each agent in the CDSC system is directly analogous to node in original contract net model in terms of its function. CDSC system underlies the manufacturing environments comprising several work cells connected through LAN. Under the environments like this, the CDSC system deals with an operational level decision making problem in a distributed, cooperative and dynamic fashion.

Based upon a contract net model by Davis and Smith[Davis, 1983], the CDSC system suggests an architecture for cooperative-distributed shop floor control. In designing the proposed system, we concentrate on the aspects of information flow and relationships for enhancing the efficiency of the whole system. Overall system is shown in figure 1. To escalate CDSC system's efficiency, the focal point is whether it refers to the information residing on other agents or not. The proposed system behaves in a manner as shown in figure 2.

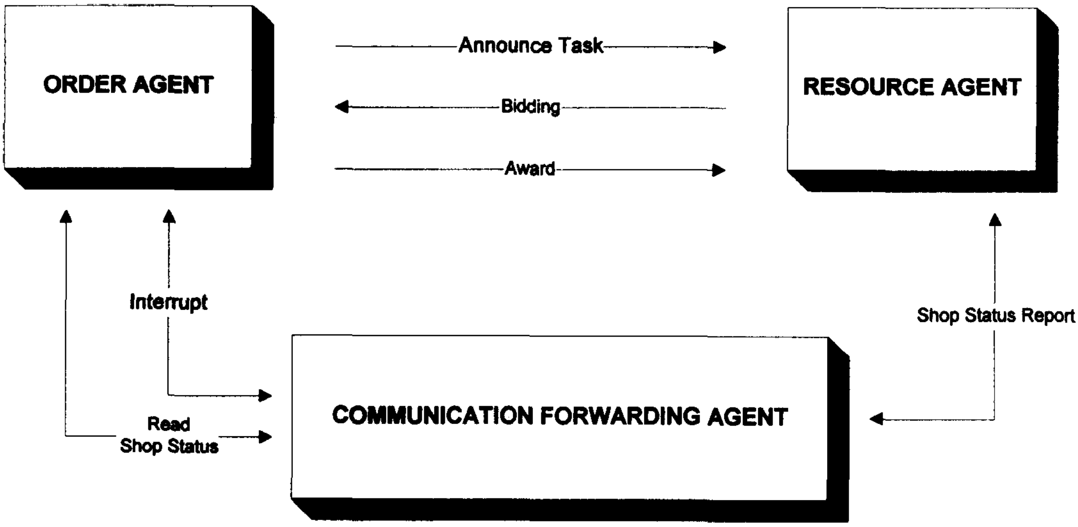

Figure 2. Information flow of the CDSC system

As in figure 2, the OA announces tasks to be carried out. The RA always refers to this announcement and sends CFA a shop status vector (SSV) containing the most current shop status information every time any changes in the system occur. Whenever task announcement reaches to RA, each RA evaluates it by computing expected job finish time according to RA-specific task evaluation procedure. The reports for evaluation are sent to the OA in the form of bids. The OA selects among bids from RA based on the criteria of bid evaluation procedure and sends an award message for selected bid(s) to RA. In this time, the RA also sends CFA an SSV declaring themselves as a flag for assignment. If a task assigned to RA is completed, an SSV is sent to CFA again. At a certain time during problem solving by the CDSC system, if a rush order arrives in the system, the OA releases interrupt message after reviewing the SSV in CFA, and processes the order in accordance with above procedure.

Table 1. Message types used in CDSC system

Acro-nym	Message	Message Content	Sour-ce	Desti-nation
TAV	Task announce vector	message_id, part_id, operation_id, routing information, job order information	OA	RA
AV	Award vector	BV_id, award, current time	OA	RA
IV	Interrupt vector	message_id, Interrupt_class_id, current time	OA	CFA
ABORT	Message abort	message_id, current time	OA	RA
SSV	Shop status vector	resource_id, resource status, loaded part_id, loading time, expected release time, # of jobs in queue	RA	CFA
BV	Bidding vector	message_id, TAV_id, resource_id, earliest start time, expected finish time	RA	OA
ACKV	Acknowledge vector	TAV_id, acknowledge, current time	RA	OA
AVAILV	Node availability vector	message_id, resource_id, current time	RA	CFA
BRDNV	Breakdown message	message_id, resource_id, current time	RA	CFA

4.1 The Order Agent

The OA plays such roles as allocation of jobs to each resource, handling of unexpected circumstances. It is a kind of job dispatcher and gives information on job to RAs. The OA has two main modules such as an intelligent dispatcher and a bid evaluation procedure, and databases for order, routing and so on. An intelligent dispatcher is a knowledge-based expert system for dynamic job scheduling. It contains a comprehensive heuristic rules for a specific setting. The OA has an objective function of maximizing throughput, and in parallel with this, also adopts an objective of minimizing the number of tardy jobs as a supporting strategy. We suggest two methodologies to support this approach. First, the notion of *job criticality* is introduced. A job criticality CR is defined by the sum of current time and remaining processing time of a given job such as below.

$$CR = TNOW + \Sigma(\text{remaining processing time}) + \alpha\ \%$$

where TNOW is a current time and α is a sort of subjectivity measure or a time buffer. A subjectivity measure α is very similar to the concept of dynamic slack used in conventional job shop scheduling theory. We can set this value by using the results of expert system's initial parameter setting. This is closely related with the management problem, so that it can contribute to assess the level of tightness in shop floor control. A job is defined to be *critical* if CR is greater than or equal to due date. If a job is announced to be "critical", the OA changes its dispatching strategy automatically using an intelligent dispatcher.

In figure 3, we described the operation mechanism and a block diagram of the order agent.

Secondly, we establish an intelligent job dispatching strategy. This is mainly due to the reason that human scheduler on shop floor performs the manual construction of job schedules based on his own experience and knowledge without any predefined specification or rules and this practice goes well. For the construction of knowledge-based expert system, we have made knowledge acquisition via simulation using simulation language SIMAN. Simulation model reflects such factors as FMS configuration, routing variability, and shop congestion level. The developed expert system is different from existing researches. Existing researches take a heuristic rule or an algorithm which satisfies given objective function best. This approach maintains a single strategy during the whole period of problem solving. (For multi-pass scheduling or control system, each time interval is a period of problem solving) Thus it would be worse in abruption of a system. In this paper, the OA can actively change its dispatching strategy in accordance with the shop status by adopting intelligent dispatching expert system mimicking human scheduler. We developed an intelligent dispatching system by two phases. A phase one is to extract facts from simulation results, and a second phase is to derive meta-rules using fact-base and to construct knowledge-base. According to our experiments, results of many expert systems reported in the literature so far could not be generalized because of its specific property. Moreover, if we want to change dispatching strategy according to the shop status, such factors as FMS configuration, routing variability, and shop congestion level are very important one which gives an influence on system performance. To meet this requirement, we considered two types of FMS layout by AGV move pattern, three types of routing variability by the number of operations in a route, and three types of shop congestion level by the utilization rate of resource. FMS layout is said to be line type if there is no intersection in AGV path because it consists of line. A loop type layout has a junction in AGV path. The number of routing is generated by a uniform distribution. We also selected 5 performance measures of Maximize throughput, Minimize number of tardy jobs, Minimize tardiness, Minimize flowtime, and Maximize AGV utilization which are proved statistically significant by the pre-experiment to test system performance.

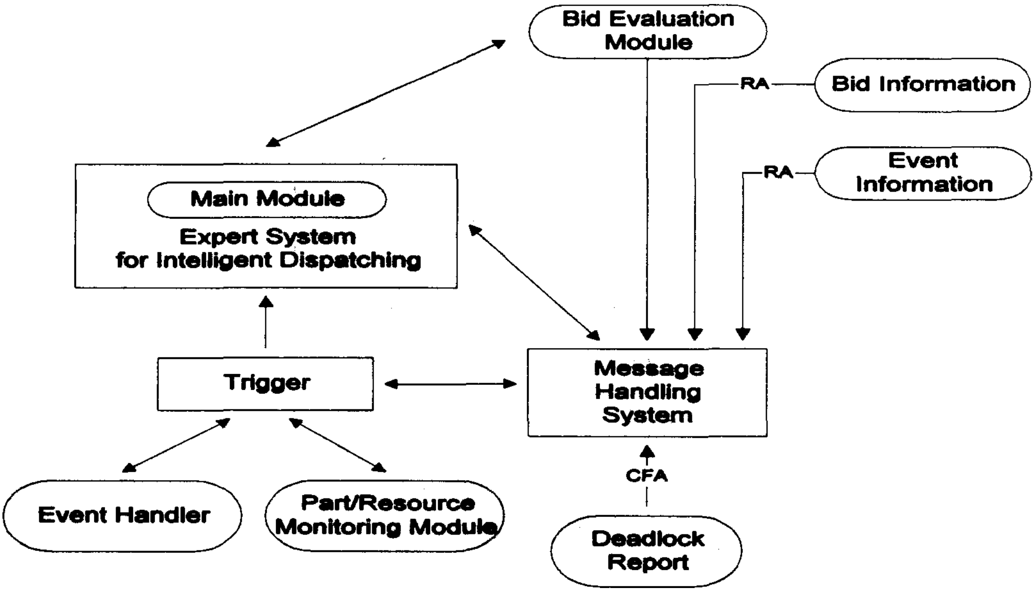

Figure 3. A block diagram for the order agent

4.2 The Resource Agent

The RA is a decision maker for each resource. Each resource has a different characteristics in terms of initial investment, resource life, operational strategy for resource, and response to customer. Consequently it is required for each resource to have an independent objective function for evaluation of orders. In order to achieve this goal, we need to develop a criteria that is capable of measuring all of these characteristics by an absolute value. Each RA may have an independent objective function, but it is assumed in this research that all RAs have same objective function. The RA plays a role of creating bids by evaluating the task announcement vector (TAV). Because a TAV contains alternative routing information, all RAs pertaining to TAV can make bids. Thus there are multiple bids for a TAV. These bids are a kind of coordinated negotiations between OA and RAs. Furthermore, the proposed CDSC system permits re-negotiation between RAs. The RA in CDSC system strongly supports the resolution of shop congestion through re-negotiation. In this research, an algorithm for re-negotiation is presented and tested. Also, we showed a block diagram of the resource agent in figure 4. To support re-negotiation, we introduce a notion of *resource congestion* level for the purpose of determining whether re-negotiation between RAs is issued or not. Resource congestion is used to check the possibility of a job completion within due date under the condition that there is no machine idle time.

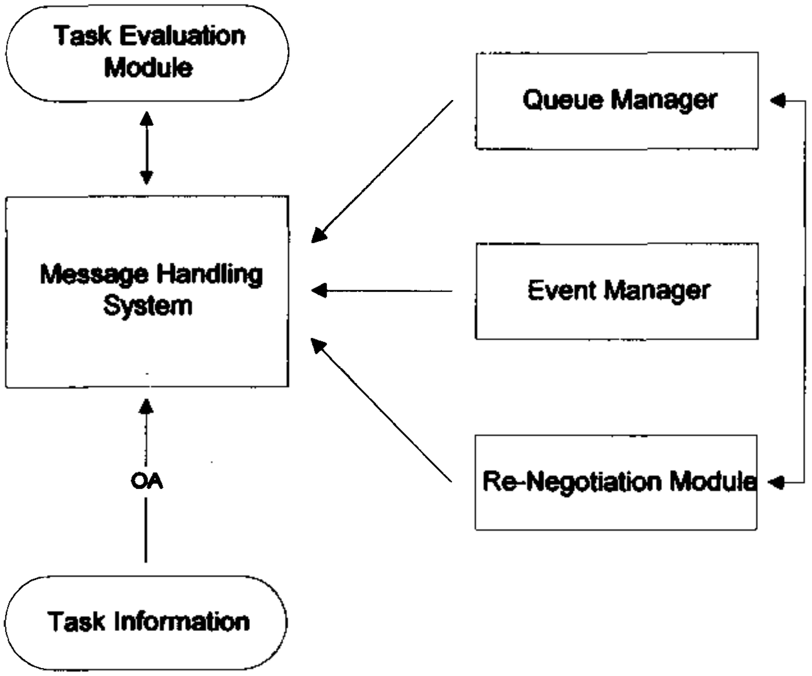

Figure 4. A block diagram for the resource agent

Let's define OQT_i for each resource i to explain resource congestio

$$OQT_i = \sum_{j=1}^{n} t_j$$

where j is a job waiting in queue of resource i. OQT_i is a sum of processing times for all jobs waiting in queue of resource i (Queue does not mean a physical one, but a job list already assigned to resource i). In other words, OQT_i means an optimistic queueing time when all jobs are processed consecutively without waiting. The earliest release time for resource i, ERT_i becomes

$$ERT_i = OQT_i + t_k$$

where k is a job in processing on resource i

Therefore the earliest finishing time for the job announced by TAV from OA to RA can be defined by

$$EFT_i = ERT_i + t_{annc} + \alpha\ \%$$

where t_{annc} is a processing time for the job announced by TAV from OA

If EFT_i is greater than due date of a job, then the resource is called to be in *congestion*. When a TAV is issued to RA to process a job, TAVs are issued to each resource listed in alternative routing except one in congestion. Using the information on congestion, a job order is assigned to other resources in alternative routing by avoiding using resources in congestion. Unfortunately, however, there is a possibility that all resources which can process TAV are in congestion. If Min[EFT_i] for all resources i is greater than or equal to a due date for the job announced by TAV, then none of resources can serve a job within a specified due date. Hence we have to select the resource that can finish the job earliest. In this case, the CDSC system issues the re-negotiation among RAs to remove or minimize tardiness. Through task sharing and resource re-allocation by re-negotiation, we may find resources that could finish the job within its due date with monotone improvements on system performance as a whole. Due to the ability of re-negotiation in CDSC system, the whole system goes through graceful degradation in terms of performance. Figure 5 illustrates a flow chart for such re-negotiation. A procedure for re-negotiation is as follows.

Re-Negotiation Algorithm

Step 1. Initialize. For all RAs that received TAV from OA, if there is no RA such that EFT is less than the due date of announced task, then start re-negotiation

Step 2. For all RAs in step 1, compute $W_i = ERT_i - t_j$ where i = 1,2, ... , r and t_j is a processing time of the task assigned to resource i last.

Step 3. Sort W_i and t_j in ascending order. After then, set X be a minimum of W_i and Y be a minimum of t_j. That is, X = RA[min. W_i] for all resource i, Y = task[min. t_j] for all task j

Step 4. If Y can be processed on X, then go to step 7. Otherwise go to step 5.

Step 5. For all t_j in step 3, let Y' be a task where Y' = task[min. t_j] for all task j and $j \notin Y$.

Step 6. If Y' is not empty, then go to step 4. Otherwise go to step 10.

Step 7. Compute $Z = W_i(X) + t_y + t_{mh} + t_{annc}$.

where $W_i(X)$ is a value of W_i in X, t_y is a processing time of Y, t_{mh} is a transition time required to material handling, and t_{annc} is a processing time of the task announced to RA. If Z is greater than due date of a task announced to RA, then go to step 10. Otherwise go to step 8.

Step 8. Evaluate criticality of X_j and Y after exchange them where X_j is a task that had been assigned to X and was excluded to be exchanged for Y. If none of them are proved critical, then admit this exchange and finish re-negotiation. Otherwise, go to step 9.

Step 9. If the sum of criticality of X_j and Y is greater than {min[EFT_i] - due date} for all resources i, then go to step 10. Otherwise admit this exchange and finish re-negotiation.

Step 10. Select an RA such as min[EFT_i] for all resources i, and finish re-negotiation.

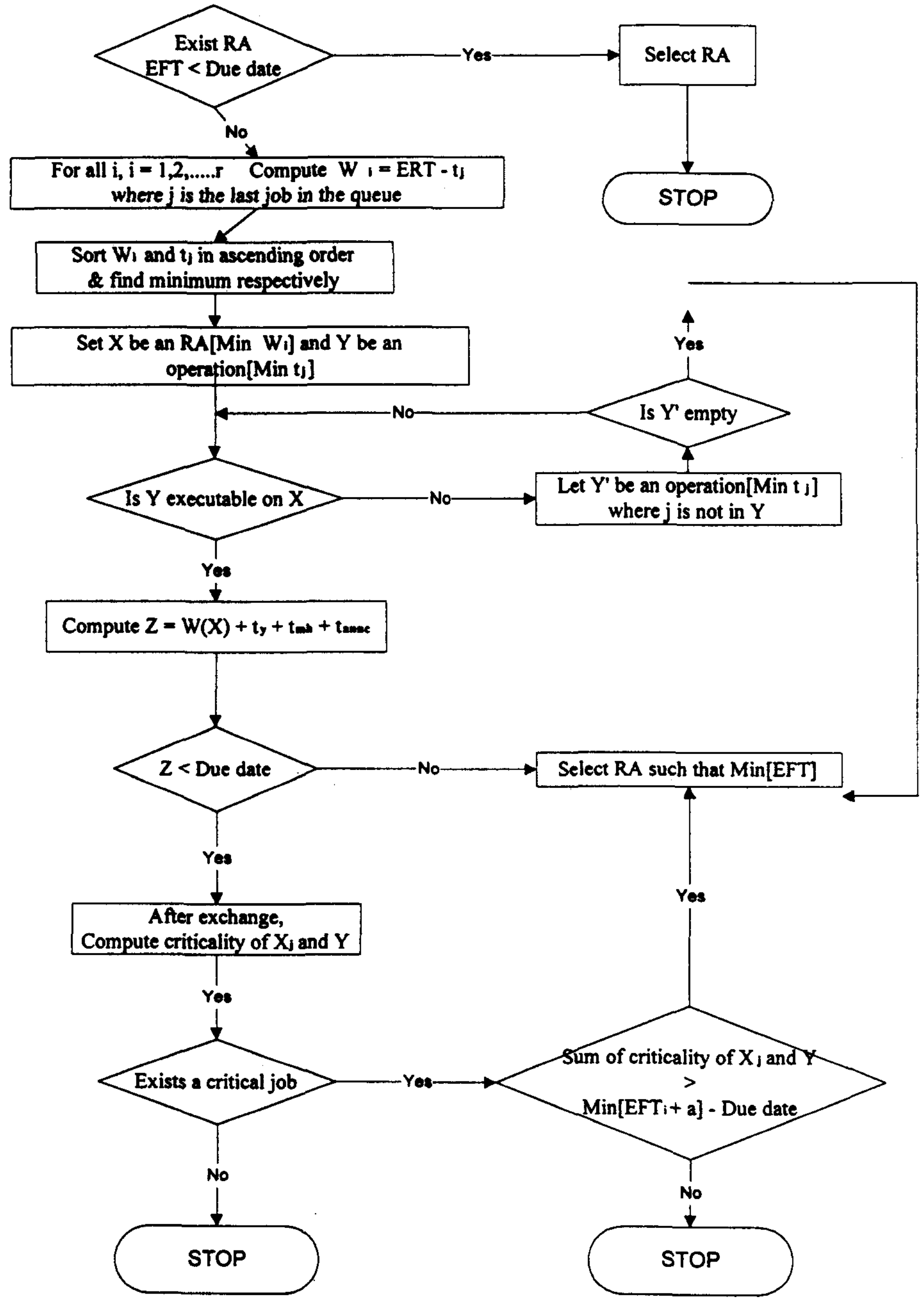

Figure 5. A flowchart for re-negotiation algorithm

In this re-negotiation algorithm, step 4 is on technological constraints. Step 7 checks due date satisfiability, step 8 and 9 are to guarantee monotone improvement in tardiness after exchange. The CDSC system views the shop control problem from the perspectives of both job and resource. During re-negotiation, an exchange of assigned jobs may increase criticality of other jobs or congestion level of other machines. A re-negotiation algorithm proposed here avoids such problem. Therefore whenever a job exchange occurs, machine congestion level can invariably be decreased by monotone improvement. It shows a very good performance compared to existing distributed scheduling approaches without re-negotiation capability.

4.3 The Communication Forwarding Agent

The original contract net model by Davis and Smith[Davis, 1983] consists of manager nodes and contractor nodes. According to Nof et. al.[Nof, 1989b], communication overheads are not a negligible factor. And also, in CIM environment, there are many kinds of transient data across the network over time. Such data need not to be stored permanently. Some kinds of information are updated in a very short time interval. If all of these data are transmitted to and stored in OA, we can easily expect a serious queuing problem. The problem results in information sender wait, violation of real-time requirements by deferred response, and inconsistent information by elapsed time and so on. As an alternative for it, we proposed a CFA as an information pool like a message buffer. Moreover, CFA has a 2-layer architecture. By 2-layer architecture, we mean decision making layer and information processing layer. We can separate decision making from information processing by using 2-layer architecture. All communications between OA and RA are forwarded through CFA. Only a message identifier is sent to the communication counterpart and the content of messages is sent to CFA. Each agent receiving the message identifier refers to CFA. For example, let's suppose OA sends a message #3 to RA. OA sends the body of message #3 to CFA, and sends an identifier containing index such as the time and message number to RA. It is very similar to the blackboard concepts in artificial intelligence. CFA always maintains most recent information on shop floor status because RA sends an SSV to CFA every time

an event occurs in system. Therefore we can easily show the shop load profile and the progress of part processing from the CFA. This can be effectively used to control shop floor in computer integrated manufacturing system. We also studied a manufacturing system deadlock control problems by using the information in CFA. But the focus of these subjects seems to be far from this paper, so that we will not consider here.

In designing CFA, we used the concept of *process* in UNIX operating system because each agent could be represented by an independent process. For this case, the communication between agents is equivalent to inter-process communication (IPC) in UNIX. We adopt a message queue in IPC which distinguishes structured messages by key. In message queue, messages are classified by their type, stored into CFA as a target process. Therefore, it has a very simple structure of interface because all processes involved in IPC are required only for sending and reading a message. All information residing in CFA can be classified into 2 categories. One relates to a system as a whole. The other relates to each resource. A system messages are such as shop status report, machine breakdown, beginning of production, end of production, and so on. A resource messages are such as machine status, length of queue, change in the length of queue, and machine buffer status. In CDSC, transient data are stored into CFA temporarily, and permanent data and knowledge are stored into OA. Thus we can reduce loads incurred by information recording, search, deletion, updating and retrieval. All information are passed through CFA, however, it may be occurred to have a sender wait phenomena. To avoid this possibility, we used a message identifier whenever information is sent. We also permit direct access to communication counterpart if there is a rush order or an emergency event. We can avoid deep queueing problem by admitting direct access. It seems that all those mechanisms are very complicate. But it is very simple for decision maker to use those mechanism because of 2-layer architecture. Separation of decision making from information processing guarantees that decision maker concentrates on decision making itself rather than underlying information processing.

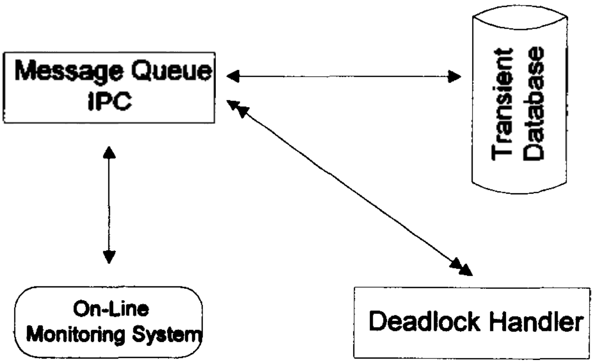

Figure 6. A block diagram for the communication forwarding agent

5. Experimental investigation

To investigate the performance of the proposed architecture, we carried out an experimentation on a pilot plant SNU-CIM center which is installed in ASRI (Automation and Systems Research Institute) at Seoul National University (SNU) in Korea. The SNU-CIM center consists of 3 manufacturing cells and loading/unloading station, material handling system, AS/RS (Automated Storage and Retrieval System), and central control room. All the equipment in the factory are interconnected by way of communication network and a protocol for communication is an Ethernet/TCP/IP.

A production plan generated by a master production schedule is based on a weekly one. Those plans are influenced by the number of product types. For shop floor control, however, routing complexity, processing time, or due date assignment are more influential rather than number of product types. Under the underlying assumptions that a part is being produced based on a given master production schedule, we already knew production quantity and due date, so that we have to concentrate on satisfying due date and finishing production as soon as possible. In these respects, we selected type of objective function, routing complexity, and due date as factors that can influence on system performance. The number of tardy jobs, total tardiness, and makespan time are used to measure system performance. Routing complexity is introduced for the purpose of examining whether re-negotiation and intelligent dispatching with multi-strategy in proposed system is suitable for shop floor control because it directly relates to shop congestion. We compared proposed system with a system that adopts contract net and negotiation without re-negotiation and multi-strategy dispatching. A data sets are generated using a random number generator. A processing time is generated from continuous uniform distribution. The number of operations and sequence of operations are generated from discrete uniform distribution. The number of simulations are determined according to the principle of experimental design with 5% significance level and 5% relative significance. Simulation is done for 100,000 unit times. One unit time is equivalent to 5 seconds in real life situation. Thus 100,000 unit times is approximately equal to 6 days. Experimental results are described graphically in from figure 7 to figure 10.

According to experiments, the proposed cooperative-distributed shop control system outperforms the system without re-negotiation and multi-strategy dispatching. As the number of operations increases, the difference between two systems increases also. This is due to virtual increases of resource capacity by re-negotiation among resources, capability of information handling that has a different priority, and adaptive decision making using shop load profile in CFA. In special, the fact that proposed system behaves better than existing system when shop is in congestion implicitly says efficiency of the re-negotiation. When due date of a product is very loose or tight, our system can not guarantee dominant performance over existing system. If due date is too tight, then the proposed system is rather worse than existing system. Because due date is too tight, many jobs are hard to meet their due. Under this situation, the proposed system issues frequent orders for job exchange by re-negotiation. But, it is unavoidable to have many jobs violating their due date in this circumstances. Therefore frequent job exchanges would rather increase shop congestion than letting the system be intact. Conversely if due date becomes too loose, then the proposed system does not always give an outperforming result.

Because due date is too loose, there is much slack time to finish production. Thus re-negotiation or multi-strategy dispatching will be hardly issued, so that they will not give much benefits. However, such a extreme case does not matter in real situation. In such a situation, we need not take elaborate care of shop floor control. For the normal production condition excluding these two extreme cases, the proposed system always outperforms existing system. If there is an extreme situation in real production, we can combine the proposed system with an existing one to deal with such situation.

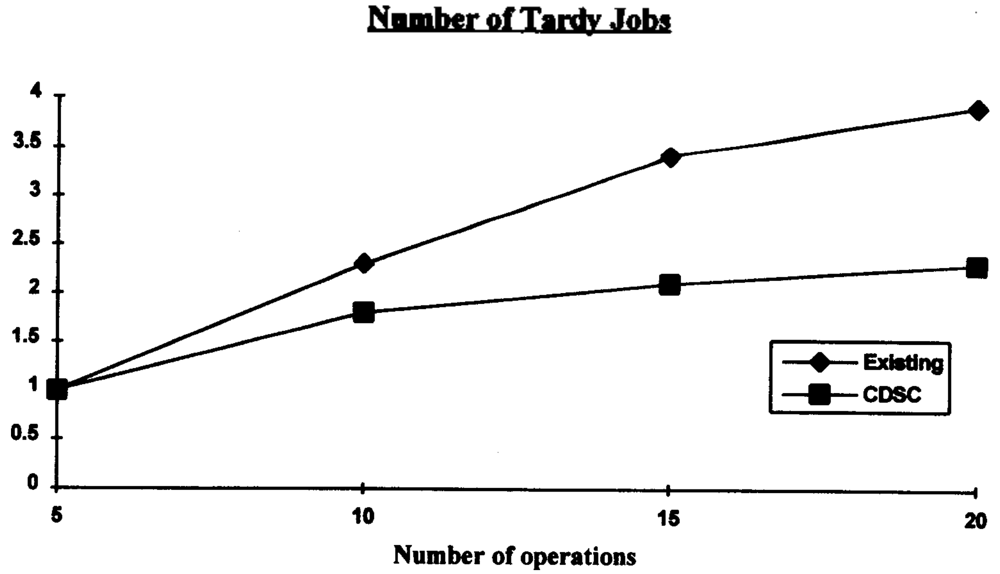

Figure 7. Number of tardy jobs

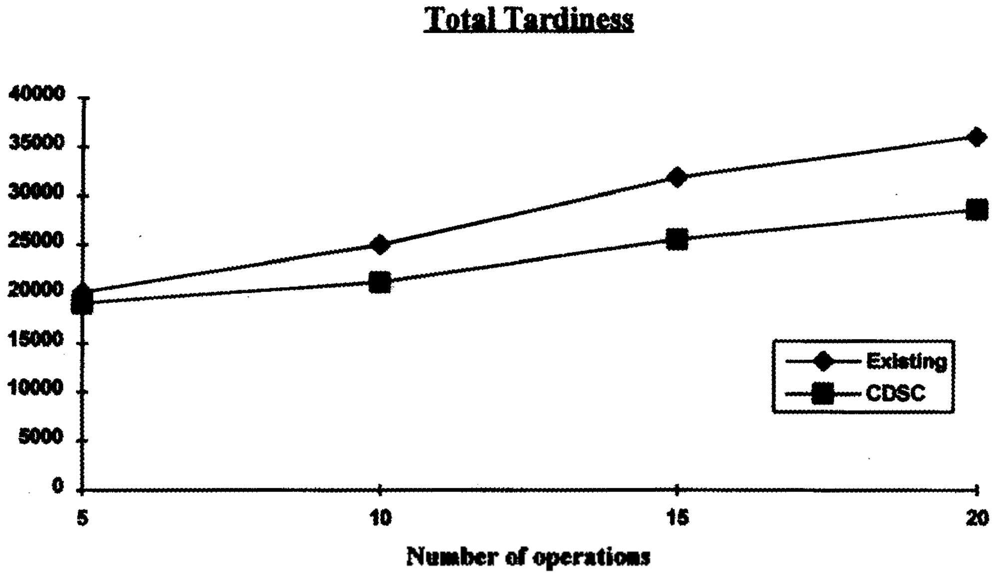

Figure 8. Total tardiness

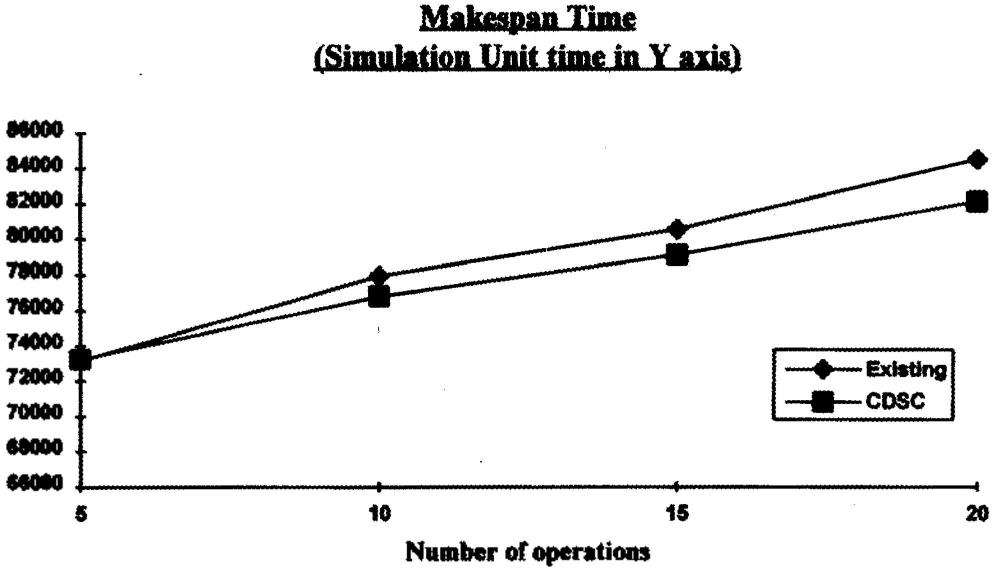

Figure 9. Makespan time

By re-negotiation, we can utilize other resource's capacity, and increase available capacity of resource in a unit time. There is also an interesting phenomena.

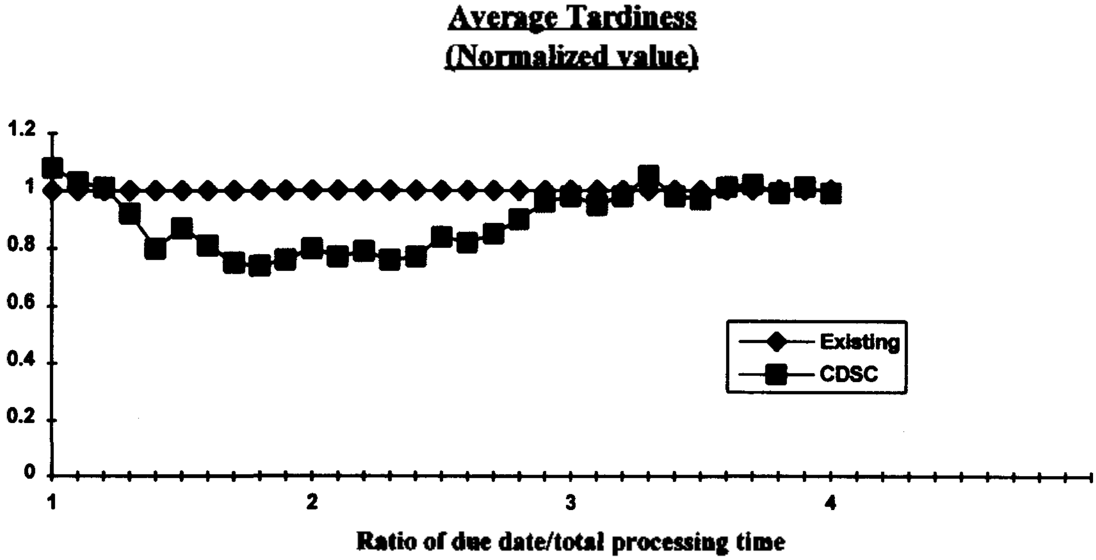

Figure 10. Average tardiness according to due date assignment

6. Conclusion and Suggestions for Further Research

We addressed a model-theoretic framework of shop floor control for developing virtual factory simulator in this research. And also, we designed and suggested a protocol for cooperative-distributed shop floor control. Experimental investigation using a prototype shows best performance of the proposed CDSC system. The CDSC system fits well into real-life situation of current CIM environments and is expected to give better quality in decision making because it uses more information. Moreover, suggested re-negotiation algorithm works well and shows a contribution to system performance. In current, all RAs in the CDSC system assume the same objective function. But, in later development, it should be considered that all RAs have different objective function because no node can have an authority over another one in CDSC system. In other words, each node is completely autonomous. And, full potential of autonomous node should be exploited. In CFA, we proposed 2-layer architecture, so that we can reduce communication burdens and increase decision making quality. To implement the system, a fine tuning of the whole system is required and refinement of re-negotiation is also required. If a machine learning is introduced to order agent, the system performance is expected to be enhanced. Knowledge acquisition for intelligent dispatching needs more intensive study including more detailed simulation and face-to-face interviews with domain experts. Determination of the due date assignment in two extreme cases is also required in order to design integrated shop floor controller for all conditions.

참고문헌

[Burbidge, 1990] Burbidge, John L. and Peter Falster, Ed., *Databases for Production Management*, North-Holland, 1990

[Cho, 1993] Cho,H. and R.A.Wysk, A robust adaptive scheduler for an intelligent workstation controller, to appear in the *Int.J.Prod.Res.*, Vol.31, No.4, pp771-789,1993

[Chang, 1988] Chang,K.H., K.Nyamekye and J.T.Black, A Distributed Expert System for Manufacturing Cell Control, *Recent Developments in Production Research* edited by A.Mital, pp861-867, 1988

[Chung, 1994] Chung, D.Y., A study on distributed shop floor control of flexible manufacturing system, *Master Thesis*, Department of Industrial Engineering, Seoul National University, 1994

[Davis, 1983] Davis,R. and R.G. Smith, Negotiation as a Metaphor for Distributed Problem Solving, *Artificial Intelligence*, Vol.20, No.1, pp63-109, 1983

[Fox, 1991] Fox, Mark S., Resource Allocation in Distributed Factory Scheduling, *IEEE Expert*, Vol.6, No.1, pp29-40, Feb.1991

[Lesser, 1981] Lesser,V.R. and D.D.Corkill, Functionally Accurate, Cooperative Distributed Systems, *IEEE Trans. on Syst. Man Cybern.*, Vol.SMC-11, No.1, pp81-96, 1981

[Lin, 1992] Lin, Grace Y.-J. and James J. Solberg, Integrated shop floor control using autonomous agents, *IIE Transactions*, Vol.24, No.3, pp57-71, 1992

[Mize, 1992] Mize,Joe H. and et. al., Modeling of integrated manufaacturing systems using an object-oriented apprpach, *IIE Transactions*, Vol.24, No.3, pp14-26, 1992

[Nof, 1989a] Nof, Shimon Y. and Colin L. Moodie, Ed., *Advanced Information Technologies for Industrial Material Flow Systems*, Springler-Verlag, 1989

[Nof, 1989b] Nof, Shimon Y. and D. Hanna, Operational characteristics of multi-robot systems with cooperation, *Int.J.Prod.Res.*, Vol.27, No.3, pp477-492, 1989

[Nof, 1992] Nof, Shimon Y., Collaborative Coordination Control of Distributed Multimachine Manufacturing, *Annals of the CIRP*, Vol.41, No.1, pp441-445, 1992

[Oliff, 1988] Oliff, Michael D., *Expert Systems and Intelligent Manufacturing*, North-Holland, 1988

[Ow, 1988] Ow, P.S., S.F.Smith and R.Howie, A Cooperative Scheduling system, *Expert Systems and Intelligent Manufacturing*, North-Holland, pp43-56, 1988

[Park, 1993] Park,N. and J.W.Park, A cooperative-distributed shop floor control for computer integrated manufacturing through negotiation and task sharing, *Proceedings of the 2nd international conference on Decision Science*, pp807-810, Seoul, Korea, 1993

[Park, 1994a] Park,N., A study on design of a cooperative-distributed shop floor control system for computer integrated manufacturing, *Ph.D. Dissertation*, Department of industrial engineering, Seoul National University, Seoul, Korea, 1994

[Park, 1994b] Park,N., S.Y.Jang and J.W.Park, Architectural design for distributed shop floor control systems under CIM environments, *Proceedings of the ORSA/TIMS joint national meeting*, p2, Detroit, 1994

[Parunak, 1986] Parunak,H.V.D. and J.Kindrick, An Architecture for Heuristic Factory Control, *Proceedings of the American Control Conference*, pp548-558, Seattle, 1986

[Parunak, 1987] Parunak,H.V.D., Manufacturing Experience with the Contract Net, *Distributed Artificial Intelligence*, Michael M.Huhns, Pitman, London, pp285-310, 1987.

[Rajan, 1990] Rajan,V.N. and S.Y.Nof, A game-theoretic approach for cooperation control in multi-machine workstations, *Int.J.CIM.*, Vol.3, No.1, pp47-59, 1990

[Shaw, 1987] Shaw, Michael J.P., A Distributed Scheduling Methods for Computer Integrated Manufacturing the use of local area networks in cellular systems, *Int.J.Prod.Res.*, Vol.25, No.9, pp1285-1303, 1987

[Shaw, 1989] Shaw, Michael J.P. and Whinston, Andrew B., A Distributed Knowledge-based Approach to Flexible Automation: The Conract Net Framework, *Int.J.Flexible Manufacturing Systems*, Vol.1, No.1, 1989

[Upton, 1991] Upton, D.M., M.M.Barash and A.M.Matheson, Architectures and auctions in manufacturing, *Int.J.CIM.*, Vol.4, No.1, pp23-33, 1991

[Valenzano, 1992] Valenzano,A., C.Demartini, and L.Ciminiera, MAP and TOP Communications, Standarads and Applications, Addison-Wesley Publishing, 1992

[Wu, 1987] Wu, Szu-Yung David, An Expert System Approach for the Control and Scheduling of Flexible Manufacturing Cells, *Ph.D. Dissertation*, The Pennsylvania State University, 1987

저자소개

박남규

서울대학교 공과대학 산업공학과에서 학사, 석사, 박사학위를 취득하였고, 미국 퍼듀대학교에서 연구원으로 있었다. 동신대학교 산업공학과에서 전임강사로 재직하였고, 현재는 생산기술연구원에 재직 중이다. 연구 관심 분야는 통합생산시스템, 정보시스템, 개발, 시뮬레이션, 대형 소프트웨어 개발 등이다. 대한산업공학회, 한국경영과학회, 한국시뮬레이션학회, 한국CALS/EC학회의 회원으로 활동 중이다.

이현정

숙명여자대학교 경영학과에서 학사, 석사학위를 취득하였다. 현재는 생산기술연구원에 재직 중이다. 연구관심 분야는 통합합관리시스템, 생산관리, 정보시스템분석 및 설계, 시뮬레이션 등이다.