Hybrid Shop Floor Control System for Computer Integrated Manufacturing (CIM)

Kyung-Hyun Choi*†
School of Mechanical Engineering, Cheju National University
Seok-Hee Lee
School of Mechanical Engineering, Pusan National University

A shop floor can be considered as an important level to develop Computer Integrated Manufacturing system (CIMs). However, a shop floor is a dynamic environment where unexpected events continuously occur, and impose changes to the planned activities. To deal with this problem, a shop floor should adopt an appropriate control system that is responsible for the coordination and control of the manufacturing physical flow and information flow. In this paper, a hybrid control system is described with a shop floor activity methodology called Multi-Layered Task Initiation Diagram (MTD). The architecture of the control model contains three levels: i.e., the shop floor controller (SFC), the intelligent agent controller (IAC) and the equipment controller (EC). The methodology behind the development of the control system is an intelligent multi-agent paradigm that enables the shop floor control system to be an independent, an autonomous, and distributed system, and to achieve an adaptability to change of the manufacturing environment.

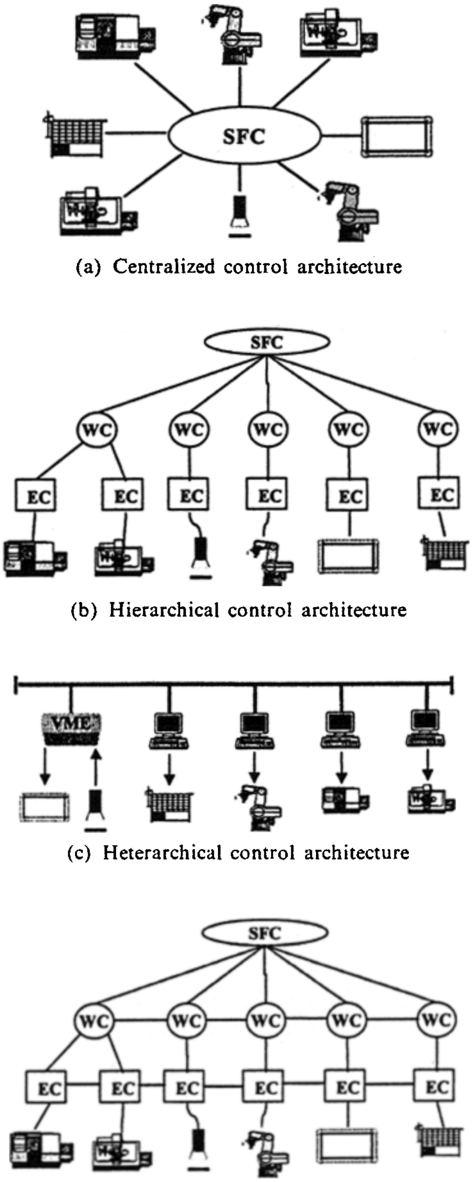
Key Words : Shop Floor, Hybrid Control System, CIM, Object-Oriented Approach, Multi-Layered Task Initiation Diagram

1. Introduction

As product life cycles are reduced, modern manufacturing systems are required to have sufficient responsiveness, and to adapt their behaviors efficiently to a wide range of circumstances. The responses to these demands include progress in the automation of manufacturing systems, such as the use of manufacturing knowledge, shorter programming times, and appropriate control modeling methodology. The efforts to achieve advanced automated factories bring into focus the development of manufacturing systems with high levels of flexibility and intelligence. (Kouiss, 1997)

CIM (Computer Integrated Manufacturing) system has been introduced to complete the advanced manufacturing systems by integrating all the new technologies. With all of its merits, the integration resulted in a rigid and hierarchical control architecture whose structural complexity grew rapidly with the size of the systems and the variety of production. To enhance CIM technology, shop floors in the modern manufacturing structure are provided with ever more versatile production equipments, such as robots, NC machines, AGV, etc. However, the shop floor is a dynamic environment where unexpected events continuously occur and impose changes on planned activities. The shop floor control system plays a very important role in dealing with problems associated with these uncertainties. Its major function is to control and to monitor the automatic equipment, which are the components

† First Author
* Corresponding Author,
E-mail : khchoi@cheju.ac.kr
TEL : +82-64-754-3713; **FAX** : +82-64-756-3886
School of Mechanical Engineering, Cheju National University, 1 Ara-dong, Cheju-city, Cheju-do 690-756, Korea.(Manuscript **Received** March 24, 2000; **Revised** February 7, 2001)



(a) Centralized control architecture

(b) Hierarchical control architecture

(c) Heterarchical control architecture

(d) Hybrid control architecture

Fig. 1 Shop floor control architecture framework

of a shop floor, and to coordinate the operations of different production activities. Therefore, the development of a reliable shop floor control system is necessary to implement shop floors. However, it is a difficult task due to the integration of multi-vendor equipments, the resource changes, and the dynamic reconfiguration in shop floor.

In order to realize shop floor control systems, Kouiss et al. (1997) have employed a multi-agent architecture, and Ming (1996) has suggested an agent-oriented analysis methodology. Liu et al. (1998) have proposed an object-oriented analysis and design method for the modeling of shop floor control systems. These methodologies allow the manufacturing system to be independent, autonomous, and distributed system, and to achieve an adaptability to change of the manufacturing environment as well. However, the control-logic and the networking components are not established for the real-time coordination and synchronization of equipment actions. As a representation methodology for activities of manufacturing systems, Ostroff et al. (1987) have devised Extended State Machine (ESM) to lead to easily verifiable specifications. Petri net or modified Petri nets (Buchholz, 1999; Valette, 1983; Mize, 1992) have successfully applied to the development of different systems.

In this paper, the hybrid shop floor control system will be described, which contains three levels, consisting of the shop floor controller, the intelligent agent controller, and the equipment controller. Each level is modeled by using an object-oriented paradigm to achieve the dynamic reconfiguration. Also new methodology, called Multi-layered Task Initiated Diagram (MTD), will represent various activities, and control logic involved in shop floors will be introduced and tested with the illustration of a simple hypothetical shop floor.

2. Shop Floor Control Framework

The understanding of the control architecture is needed to generate appropriate modules for constructing a shop floor control system. In this sense, the control architecture describes the specification of the decomposition of the control systems functionality as required by shop floor control systems, and the relationships between decomposed modules. As shown in Fig. 1, there are four typical types of control architectures: centralized form, hierarchical form, heterarchical form, and hybrid form. The centralized control architecture employs a centralized computer or controller to manage and maintain the records of all planning and information processing functions (Fig. 1(a)). Machines employed on a shop floor execute the commands released from the centralized controller, and then feed back the results to the centralized controller. Traditionally, a shop floor control has been performed on a centralized computer. This architecture approach is most suited for a completely deterministic environment, (Deshmakh, 1995) however, it has a limited ability for flexible manufacturing systems. (Lin, 1991)

The hierarchical control architecture, as shown in Fig. 1(b), contains a rigid master/slave relationship between two adjacent levels of controllers, that is, SFC (Shop floor controller) and WC (Workstation Controller), WC and EC (Equipment Controller). The exchange of information is not allowed between controllers at the same levels. Within the hierarchy of controllers, a superior sees only its immediate subordinates and not the subordinates of its subordinates. This concept gives each controller a certain control authority within its realm. Due to these benefits, a common structure of a shop floor control system has been defined as using a hierarchical architecture, where central planners of SFC generate a general scheduling and routing plan. This plan is sent to the lower levels, WC, where it is further refined and more details are added. Finally, the schedule is sent to the EC that act on the decision made by higher level. (Duffie, 1994) However, this architecture requires some efforts when a shop floor control system needs to be modified.

Heterarchical control architecture is composed of a set of quasi-independent controllers without a rigid master/slave hierarchy. As shown in Fig. 1(c), one major feature of this architecture is the pursuit of the full local autonomy and the cooperative approach to global decision making. So, this control architecture leads a shop or manufacturing systems to be modular, extensible and self configurable. In addition, since the modern computer technology has been tremendously improved, the control negotiations among controllers are processed at very high speeds and the communication burden is reduced. A benefit of this architecture is that it can be applied in a relatively complicated manufacturing system that consists of loosely coupled highly autonomous entities retaining minimal global information. However, heterarchical systems are internally well ordered but externally permissive, and are organized as a cooperative hierarchy, so that it suffers from a myopic view, isolation of decision-makers and a lack of conflict resolution. (Mcdonnell, 1995)

With the hybrid control feature, it has a loose master/slave relationship between control levels, and a peer to peer communication, such as WC-WC or EC-EC. A superior is responsible for initiating a sequence of activities, whereas the subordinates are able to harmonically complete these activities in sequence. For the completion of tasks each controller interacts with the same level controllers.

3. Object Modeling of Multi-Agent Hybrid Shop Floor Control System

The object-oriented approach is used to describe a method of modeling for the multi-agent hybrid shop floor control system in which the system is organized as a collection of discrete objects. Each modeled object not only contains both the data and the behavior, but also corresponds to the physical object associated with a shop floor or a manufacturing system. A shop floor or a manufacturing system is considered as a composition of two major parts, that is, the set of physical devices which require control (the controlled objects) and the set of controllers (the controlling objects). This object-oriented approach has effects on the development of the interface between the physical devices and the shop floor control system because it provides a vendor independent interface based on equipment type.

3.1 Shop floor resource model

As shown in Fig. 2, an abstraction resource model composed of a shop floor is modeled by using UML (Unified Modeling Language) that is

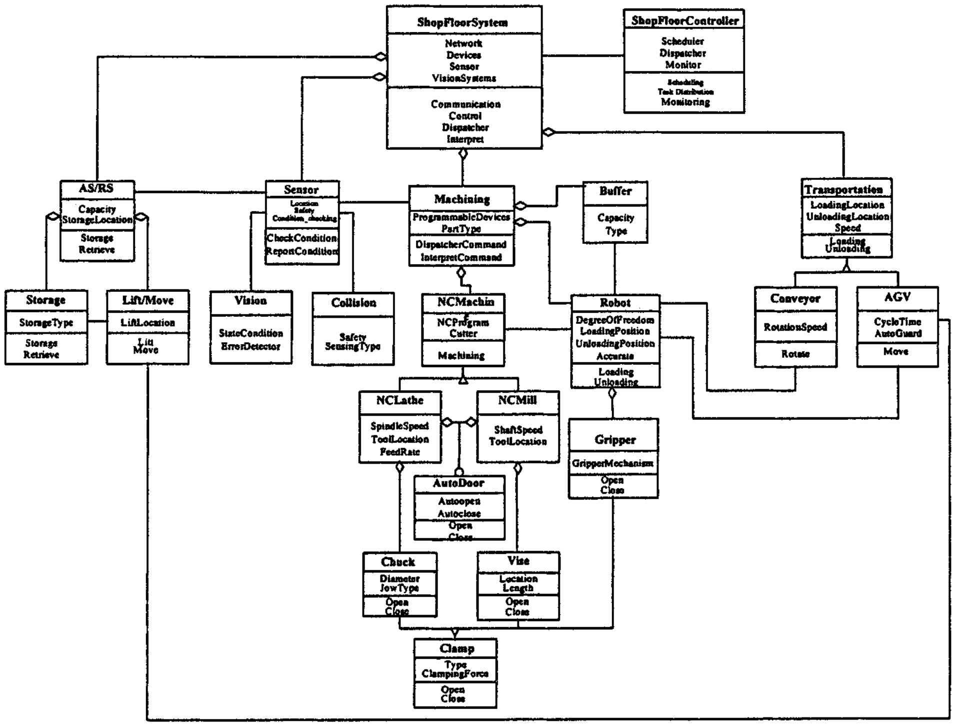


Fig. 2 Shop floor resource object model

one of the object-oriented approach methods. All classes developed in the model are designed to represent the real manufacturing system objects and group of objects. They contain aggregating generic objects or machines (robot, NC, conveyor, etc.) that can easily be specialized and refined. A shop floor can be viewed as a group of work centers. A work center is composed of manufacturing cells, such as a machining cell, an AS/RS cell, an inspection cell, a conveyor and sensors. In turn, a machining cell contains lathes, milling machines, robots, and a cell controller. An inventory is released by AS/RS and transferred to robots through a conveyor. The activities of robots are to load parts into the machines and to unload processed parts from the machines. After material processing, the part inspection is performed by a vision system or a coordinate measurement machine (CMM). And several sensors are embedded in the manufacturing system or shop floor to monitor the functioning of machines, a conveyor, and robots.

3.2 Shop floor control model

The shop floor control system, associated with the shop floor resource, imposes an adaptable hierarchy over other intelligent agent controllers and keeps track of the shop floor status. Such an adaptive hierarchy can help impose product priorities, changes of environment and cope with internal disturbances by reassigning other objects. As shown in Fig. 3, the control architecture implemented here consists of three levels of controllers. : the equipment controller (EC), the intelligent agent controller (IAC), and the shop floor controller (SFC).

3.2.1 Equipment controller (EC)

The Equipment controller (EC) is on the lowest level of the hybrid control architecture. There is one equipment level controller for each piece of equipment in the system. Individual machine pieces of equipment also have machine controllers that provide physical control for the devices. These include robot controllers, NC

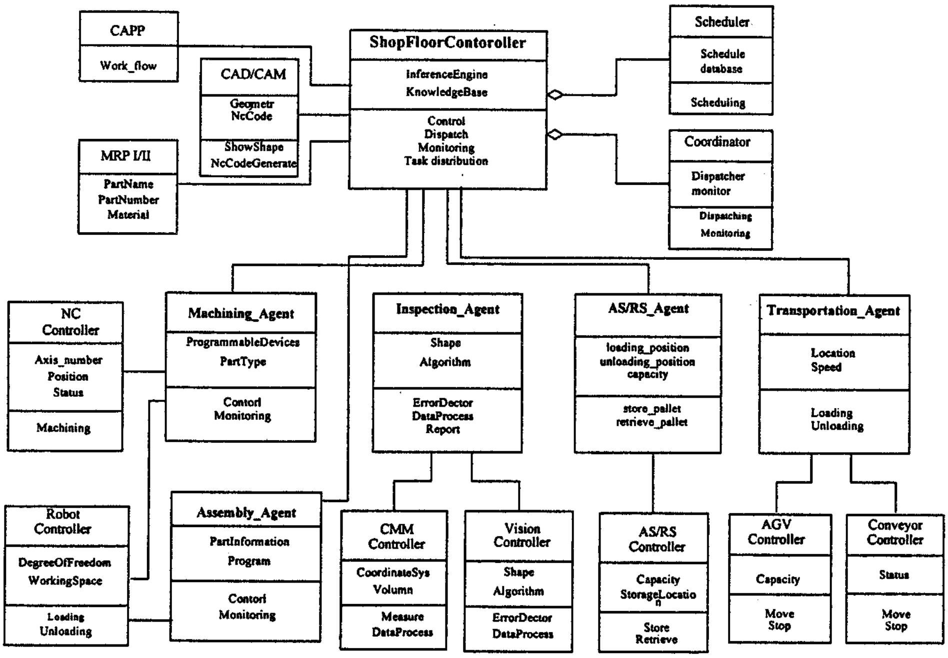


Fig. 3 Object model for hybrid shop floor control system

controllers, PLC, and motion controllers and are usually provided by machine tool vendors. Equipment controllers provide a generic interface, based on equipment type, to other equipment controllers and to a higher level controller, the intelligent agent controller.

An equipment controller converts the processing instruction data into a form directly usable by the specific machine controller and monitors the operation of the machine under its control and it reports the state of a particular machine to an upper level controller, i. e. intelligent agent controller. The behavior of a controller is described by the correspondence in the multi-layered task operation diagram and a resource model.

3.2.2 Intelligent agent controller (IAC)

The intelligent agent controller (IAC) corresponds with a small subset of equipments such as an industrial robot and a NC machine for unloading and loading. The intelligent agent controller deals with commands and information received from the shop floor controller, and is responsible for moving parts between the various pieces of equipment and for specifying processes performed at the equipment. This intelligent agent controller employs a multi-agent paradigm. An agent is defined by the aggregated function classes of shop floor equipment, and encapsulates an entity similar to the real system; such as a machining agent, inspection agent, assembly agent, AS/RS agent, and transportation agent. Each agent controller has a hierarchical structure and the function of the learning behaviors of other agent controllers. Additionally, it can detect an error by the ability of recognition that is acquired through the results of learning, and able to recover the system to a normal state, and communicate with other agents by transferring messages to execute tasks. (Franklin, 1996)

In order to effectively complete control tasks, as shown in Fig. 4, an intelligent agent controller consists of three sub-agents; that is, the make agent, the decision-making agent, and the communication agent. The memory agent manages the entire datum and knowledge that concerns itself and other agents, the shop floor environment and the results of monitoring the system. The communication agent which consists

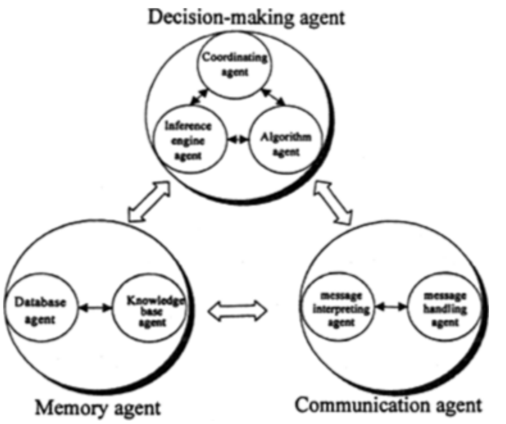


Fig. 4 Intelligent Agent Controller components and relationship

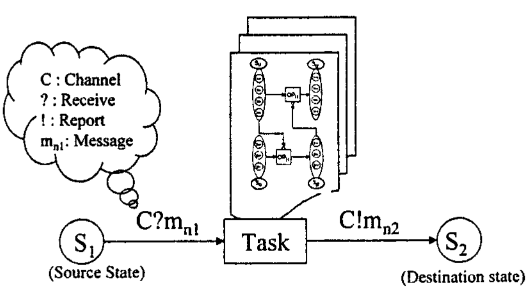


Fig. 5 MTD basic structure

of a message handling agent and a message interpreting agent plays the role as the interface mechanism between agents. A message-handling agent regulates message transfer between connected agents and relays messages to the message-interpreting agent. The message-interpreting agent transforms those messages into relevant information and conveys them to the decision-making agent.

The decision-making agent consists of the decision-making coordinating agent, the inference engine agent, and the decision-making algorithm agent. The decision-making coordinating agent defines those tasks into specific problems to the inference engine agent or the decision-making algorithm agent, or both. The inference engine agent communicates with the memory agent to get accumulated knowledge and available databases to solve their problems. But on the other hand, the decision-making algorithm agent mainly utilizes database. In the solution process, these two agents interact with each other to exchange relevant information. After decisions, they are transferred to the other agents through the decision-making coordination agent and the communication agent.

3.2.3 Shop floor controller (SFC)

The SFC is responsible for all the system-level management, coordination and control. It is also the sole communication port with the external systems, i.e., CAD/CAM, CAPP, and MRP. It consists of two modules, i.e. a scheduler and a coordinator that has two sub-modules: a dispatcher and a monitor. The scheduler determines the optimal tasks, taking into account the finite capacity of the machine tools and what is to be done by the intelligent agent controller. The coordinator manages the set of intelligent agent controllers during production and executes the schedule by dispatching work-orders and constantly monitoring the intelligent agent controllers.

4. Multi-Layered Task Initiation Diagram

The real-time operation of a shop floor is performed by the equipment controller that dispatches a sequence of operation commands to the shop floor equipment. The generation of these operation commands is based on a behavior model of the shop floor control system, with which mapping between the physical and the logical system is established.

In this paper, as a formal representation of the operational behavior of a shop floor control system, a new methodology, which is depicted by a set of diagrams called MTD(Multi-layered Task initiation Diagram), has been developed with their accompanying rules. This MTD regards the tasks to be performed by the shop floor or any of its constituent machines as being primal. Sensor signals indicating the change of state of machines are used to trigger or to initiate tasks. A task may be simple and require a relatively short time to execute, or may be complex and lengthy.

As shown in Fig. 5, the MTD is multi-layered, so that the upper layer shows the dynamic behavior of the task level which is treated at the intelligent agent controller level, while the lower layer illustrates that of the operation-level which is associated with the equipment controller level. MTD is composed of two basic components: a set of tasks and a set of states. MTD is defined as the five tuple, MTD$\{S, E, C, T, A\}$, where S represents a set of states between operations, E represents a set of event labels, C represents a set of communication channel, T represents a set of tasks, and A represents a set of basic-actions.

Table 1 Information and order of parts to be fabricated

variable / Part name	Part drawing	Process	Quantity
P_R1		Milling	20
P_R2		Turning+ Milling	10
P_R3		Turning	15

Each basic-action is given by $\{Ss,$ *Guard, Message, Operation , Confirmation* , $Sd\}$, where Ss is a source state, Guard is a boolean-valued expression which represents the success of the communication, Message represents an event transported through a channel, Operation is a set of tasks between states, Confirmation notifies the success of an operation, and Sd is a destination state or task.

In order to transfer state Ss into Sd, the *Operation*, that is task or operation and is depicted as a box in the diagram, should be executed and completed. *Operation* is classified into two operations or tasks: composite task and single task. A composite task consists of a concerted group of subtasks or operations involving more than one constituent of the shop floor, and is depicted by the framed boxes in the diagram. Whereas tasks involving a single machine are called simple tasks and shown by a box. The task or operation is permitted to begin after the messages arrive through the communication channel. As soon as the task is completed, a confirmation message is sent to the related controllers, and the state transfers into the state

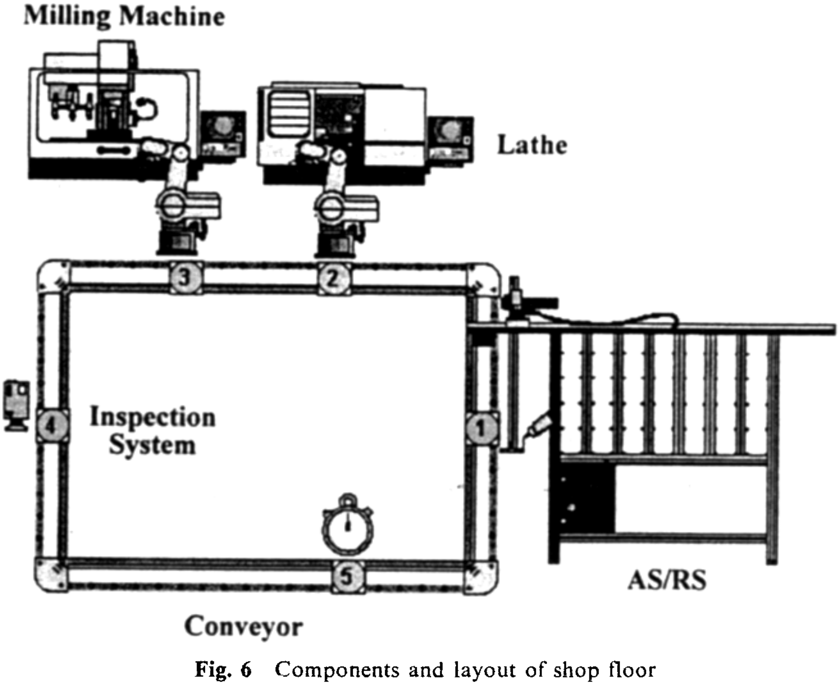


Fig. 6 Components and layout of shop floor

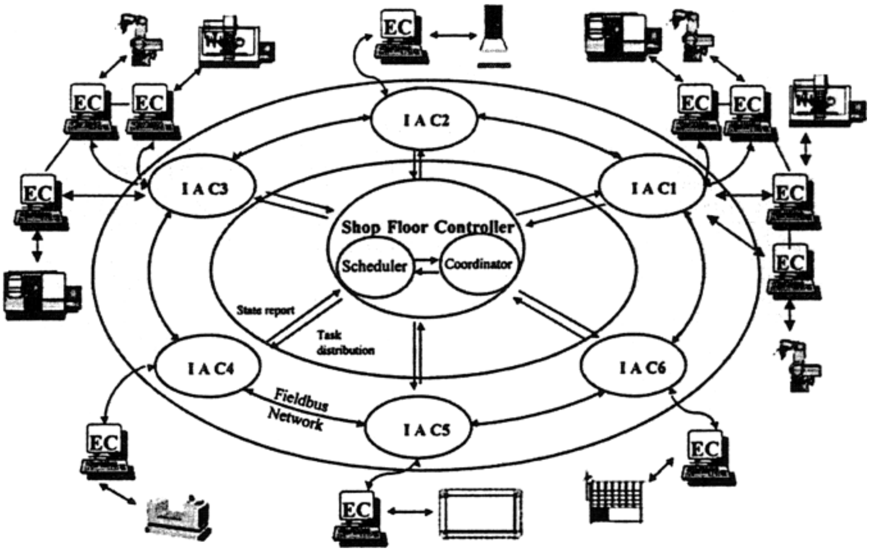


Fig. 7 Multi-agent hybrid shop floor control system structure

Sd. During the execution of a task, the time bound check mechanism is employed to detect the occurrence of an error and a deadlock problem.

This state is classified into two types: rest and wait. A rest state indicates that a controller must wait for commands from the upper level controllers, the shop floor controller and the intelligent agent controller, before its next task can be initiated. In contrast, the wait state indicates an interaction between same level controllers and hence coordination among them.

5. Case Example

In order to verify the applicability of the suggested hybrid shop floor control system, a flexible manufacturing system as a hypothetical shop floor is considered, consisting of two NC machines, two robots, AS/RS, an inspection cell, and a conveyor, as shown in Fig. 6. Table 1 shows information about products to be ordered by SFC, including the quantity and the required processes. Each product requires the specific process that is manipulated by a lathe or a milling, or both of them.

The structure of the multi-agent hybrid shop floor control system developed for the given flexible manufacturing system is illustrated in Fig. 7. The control architecture is a hybrid architecture where the equipment controllers (EC) and the intelligent agent controllers (IAC) interconnect with each other as heterarchical system and they have a hierarchical architecture with the shop floor controller(SFC). In order to effectively connect these different level controllers, an appropriate communication protocol should be determined based on the communication time. Two kinds of communication protocol are employed here such as fieldbus and ethernet. The fieldbus protocol is designed to support time-critical communication to and from devices, and is used where needed in real-time communication like between robots and NC machines, or lower level controllers. The ethernet protocol is utilized in the non-real time communication, for example, tasks or data transferring from CAPP, MRPI/II, CAD/CAM, etc. to the shop floor controller.

Among the intelligent agent controllers, the IAC1 synchronizes the actions required for coordinating the transfer of parts between processing machines (e.g., NC machines) and material handling equipment (e.g., robot). Since the equipment controller of NC machines and the robot are responsible for completing their tasks once the tasks have been assigned by the IAC1, The IAC1 is not responsible for loading,

Table 2 Tasks and operations

Cell Tasks		Operations	
Name	Description	Name	Description
T1	Load part in NC_lathe with Robot1	OR$_{R1}$	Moving in NC with part
T2	Machining part in NC_lathe	OP$_{R2}$	Gripper opening & moving back
T3	Unload part in NC_lathe with Robot1	OP$_{R3}$	Moving in NC without part
T4	Load part in NC_milling with Robot2	OP$_{R4}$	Gripper openning
T5	Machining part in NC_milling	OP$_{R5}$	Moving back and put down
T6	Unload part in NC_milling with Robot2	OP$_{R6}$	Picking
Td	Raw material delivering	OP$_{N1}$	Vice closing
Tt	Transferring part	OP$_{N2}$	Door closing
Tm	Prduct moving	OP$_{N3}$	Machining
		OP$_{N4}$	Door opening
		OP$_{N5}$	Vice opening

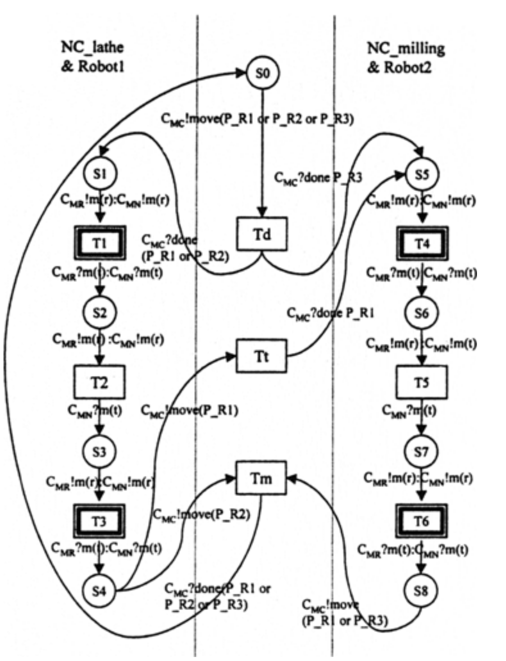


Fig. 8 MTD model of Intelligent agent controller (IAC1)

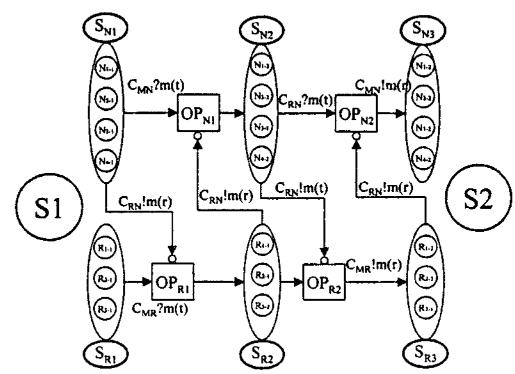


Fig. 9 MTD model of equipment controllers for unloading task(T3)

unloading, machining, and to monitoring the operation of the machines or the robot directly. Figure 8 shows MTD model representing the behavior of IAC1. This model is used to control and manage the robot and NC machines. Tasks T1(loading) and T3(unloading) in Fig. 8 are composite tasks that consist of a concerted group of subtasks or operations involving a robot and NC machines. The communication channel between two controllers is depicted as C12, for example, CMN and CRN indicate channels between IAC and a NC machine controller, and between a robot controller and a NC machine controller, respectively. Messages via a channel are classified into three meanings, a report m(r), a trigger m(t), and a specific command like move (PR3). Also, states are given at any instance by the collection of the states of its constituents. All the operations are described in detail in Table 2.

Figure 9 illustrates a control model of the equipment controller of a robot and a NC machine for unloading task. It describes the real-time based operations triggered by sensory information. Each equipment controller of a

Table 3 Robot composite states

STATE	Gripper	Part	Location	Name
S_{R1}	Closed (R_{1_1})	Present (R_{2_1})	NC_Outside (R_{3_1})	Ready to load
S_{R2}	Closed (R_{1_1})	Present (R_{2_1})	NC_Outside (R_{3_2})	In NCi with part
S_{R3}	Open (R_{1_2})	Absent (R_{2_2})	NC_Outside (R_{3_1})	Ready to unload
S_{R4}	Open (R_{1_2})	Absent (R_{2_2})	NC_Inside (R_{3_2})	In NCi without part
S_{R5}	Closed (R_{1_1})	Present (R_{2_1})	NC_Inside (R_{3_2})	In NCi with part
S_{R6}	Closed (R_{1_1})	Absent (R_{2_2})	Above conveyor (R_{3_3})	Ready to unload

Table 4 NC composite states

STATE	Vise	Part	Door	Activity	Name
S_{N1}	Open(N_{1_1})	Absent(N_{2_1})	Open(N_{3_1})	Rest(N_{4_1})	Ready to load
S_{N2}	Closed(N_{1_2})	Present(N_{2_2})	Open(N_{3_1})	Wait(N_{4_2})	In part
S_{N3}	Closed(N_{1_2})	Present(N_{2_2})	Open(N_{3_2})	Wait(N_{4_2})	Ready to machining
S_{N4}	Closed(N_{1_2})	Present(N_{2_2})	Open(N_{3_2})	Wait(N_{4_2})	machining completed
S_{N5}	Closed(N_{1_2})	Present(N_{2_2})	Open(N_{3_1})	Wait(N_{4_2})	Ready to unload

robot and a NC machine manages and controls operations, unloading parts from NC machines. The equipment controller receives the initiation commands for these operations from IAC1, and also reports the status and the completion of operation to the IAC1. A robot and a NC machine must be in the rest state ($SR1$, $SR3$, $SN1$, or $SN3$), waiting for commands from the IAC1 before its next task can be initiated. In contrast, the wait state ($SR2$ or $Sn2$) indicates an interaction between a robot and a NC machine and hence the coordination among them. The state set is given at any instance by the collection of states of its components, such as, gripper openness, position, and part existence. All state descriptions are listed in Tables 2 and 3.

6. Conclusion

In this paper, a multi-agent hybrid control system for a shop floor or manufacturing systems has been investigated and modeled by using an agent and object-oriented paradigm. As a modeling technique, UML (Unified Modeling Language) is employed and provides reusability, extendibility and modifiability of the resulting software design. The hybrid shop floor control system developed consists of three levels of controllers : the Equipment, the Intelligent Agent, and the Shop Floor Controllers. With benefits of an object-oriented approach the shop floor control system is designed to adapt to an unstable environment and will become an independent, distributed, cooperative system as well as an efficient system. As a formal representation of the operational behavior of a shop floor control system, a new methodology called MTD (Multi-Layered Task initiation Diagram) has been developed and regards the tasks to be performed by the shop floor or any of its constituent machines as being primal. After testing with the simulation of a simple hypothetical shop floor, MTD proves that it provides efficient representation of various activities and control logic associated with the hybrid shop floor control system.

Reference

Buchholz P., 1999, "Hierarchical Structuring of Proposed GSPNs. *IEEE Transactions on Software Engineering*," Vol. 25, pp. 166~181.

Deshmukh, A.V., Benjaafar, S., Talavage, J.J., Barash, M.M., 1995, "Comparison of Centralized and Distributed Policies for Manufacturing System," *Procs. 4th Industrial Engineering Research Conference*, Nashville, TN., pp. 744~748.

Dilts, D.M., Boyd, N.P., and Whorms, H.H., 1991, "The Evolution of Control Architecture Automated Manufacturing System," *Journal of Manufacturing Systems*, Vol. 10, pp. 79~93.

Duffie, N. A., and Prabhu, V., 1994, "Real-Time Distributed Scheduling of Heterachical Manufacturing Systems," *Journal of Manufacturing Systems*, Vol. 13, pp. 94~107.

Franklin, S., and Grasser, A., 1996, "Is It an Agent, or just a Program ?," *Proceedings of the Third International Workshop on Agent Theories*, Architectures, and Languages, Springer-Verlag, pp. 54~66.

Kouiss, K., Pierreval, H., Mebarki, N., 1997, "Using Multi-Agent Architecture in FMS for Dynamic Scheduling," *Journal of Intelligent Manufacturing*, Vol. 8, pp. 41~44.

Liu, C.M., Chien, C.F., Ho, I.Y., 1998, "An Object-Oriented Analysis and Design Method for Shop Floor Control System," *International Journal of Computer Integrated Manufacturing*, Vol. 11, pp. 379~400.

Lin, Grace Y.J., and Solberg, J.J., 1991, "Effectiveness of Flexible Routing Control," *International Journal of Flexible Manufacturing Systems*, Vol. 3, pp. 189~212.

McDonnell, P. and Joshi, S.B., 1995, "The Intelligent Hierarchy: A Framework for Distributed Shop-Floor Control," *4th Industrial Engineering Research Conference*, Nashville, TN., pp. 808~816.

Ming, L., 1996, "Agent-Oriented Analysis Methodology in Intelligent Manufacturing System," *The 4th International Conference on Control, Automation, Robotics and Vision* (ICARCV' 96), pp. 254~258.

Mize, J.H., Bhuskute, H.C., Pratt, D.B., Kamath, M., 1992, "Modeling of Integrated Manufacturing Systems Using an Object-Oriented Approach," *IIE Transactions*, vol. 24, pp. 14~26.

Ostroff, J.S., and Wonham, W.M., 1987, "Modeling, Specification, and Verifying Real-Time Systems," *Symposium IEEE Computer Society*, San Jose, CA, pp. 132~142.

Valette, R., 1983, "A Petri Net Based Programmable Logic Controller," *Computer Applications in Production and Engineering*, Vol. 11, pp. 103~115.