To appear in Proceedings of IEEE Conference on Robotics and Automation, San Fransisco, April, 2000

Application of the Holonic Component-Based Approach to the Control of a Robot Assembly Cell

Jin-Lung Chirn, Duncan C. McFarlane

Institute for Manufacturing
University of Cambridge
Mill Lane, Cambridge, CB2 1RX, UK

Abstract: *The aim of this paper is to implement a new approach for the introduction of so called "holonic manufacturing" principles into existing production control systems to cope with the increasing requirements of production change. A conceptual architecture is described and implemented in a robot assembly cell to demonstrate that this approach can lead to a manufacturing control system which can adapt relatively simply to long-term change. A design methodology and migration strategy for achieving these solutions using conventional hardware is proposed to develop execution level of manufacturing control systems. Finally, the ability of the control system to reconfigure in the face of changed is examined.*

1. Introduction

The increasing requirement for change faced by many manufacturing companies poses a significant challenge for traditional manufacturing control system architectures. The conventional Computer Integrated Manufacturing (CIM) approaches, while successfully providing for a standardised systems architecture, do not, in general, adapt simply; the cost of change is often expensive [9,15]. A recently proposed alternative approach to system architecture development, entitled the *holonic manufacturing* principle has promised to deal with this challenge. This approach was originally introduced by Arthur Koestler [10] and later developed by Suda [19,20] and Christensen [3] as a means of providing a building block or "plug and play" capability for developing and operating a manufacturing system in the future factory. The ultimate aim of this approach is to develop an architecture for highly decentralised manufacturing systems, built from a modular mix of standardised, autonomous, co-operative and intelligent components, in order to cope with the rapidly changing environment.

An increasing amount of research has been conducted in 1990s. Both the intra stricture [8] and inter structure [1] of holons have been investigated in manufacturing perspective. The concept of holons also has been applied to a diverse range of industries and applications, [8,12,22] to perform the different planning and control functions required for managing a production operation. The planning and control work to date has been focused on developing distributed and co-operative problem formulations and solution strategies [7,16,23]. However, there are at least two critical issues which are not yet to be addressed before holonic control solutions can be expected to play any significant part in next generation manufacturing systems [13]. The first issue is a migration to full holonic production and control. In general, research to date has explored solutions to individual manufacturing control problems. The development of manufacturing holons which can seamlessly integrate all of the different control functions into their operations at different layers should be considered next. The second issue is to establish suitable implementation approaches with existing and future commercial computing systems. There has been little or no work done in determining the compatibility of the holonic vision with the current or the next generation of industrial control and computing systems. Determining how to construct and interface systems capable of fully supporting holonic operations with existing legacy systems will also be a major issue as holonic systems capabilities reach industrial strength.

This paper begins in developing a holonic architecture by eliminating the gap between the current research and legacy systems which has been mentioned above. A new approach called *Holonic Component-Based Architecture* (HCBA) is first introduced and then implemented into the control of a robot assembly cell. However, this paper does not attempt to cover manufacturing activities in all layers thoroughly. Rather, this approach is intended to integrate the physical manufacturing plant and supporting control system and implement the control in the execution level. The ability of the control system to cope with long-term change is examined and the results of the testbed are discussed.

2. Holonic Component-Based Architecture

In this section, the background and the concept to support HCBA is introduced.

2.1 Component-Based Structure

Current manufacturing systems have generally been created using the traditional software engineering paradigm, which typically produces centralised, functionally decomposed, monolithic, and hierarchical software on a top-down design basis [21]. A top-down design approach is believed to result in better optimisation of the design, as a system is viewed from a more complete perspective. However, some authors have analysed that this approach suffers from sensitivity to changes both in user requirements and global conceptual design [1]. A traditional top-down development methodology considers the user requirements and the global conceptual design as the complete set of constraints and requirements that solutions should satisfy. As a

consequence, developers are pressed to develop subsystems that depend heavily on the context provided by the user requirements and the global conceptual design. This results in fragile solutions, which fail to cope with changes in their context. This is a plausible reason for explaining why the traditional CIM approach tends to yield a rigid architecture in manufacturing systems.

The idea of component-based development (CBD) [6,17], on a distributed and bottom-up design basis, is employed in this paper to devise a dynamic architecture which has the potential for continual change in the manufacturing environment. The idea of CBD can be traced back to the introduction of the concept of software integrated circuits to develop and package software components for later use just as hardware components are packaged for convenient use in integrated circuits [4]. Like hardware ICs, software ICs can be designed, tested, and maintained by suppliers. Component suppliers are good at developing their own technology. Therefore, they can deliver high quality and ingenious software ICs for users. System integrators may focus their efforts on building up their own applications and save much time by not needing to design the individual components.

2.2 System Components

As discussed above, the component-based architectural design uses a bottom-up approach to build a whole system. Thus, the composition of basic holons could initially be based on the fundamental elements of a physical plant. The physical objects of a manufacturing plant can be categorised into two general groups in terms of their properties. One is the *resource* which performs the manufacturing operations and the other is the *product* which accepts the manufacturing treatments.

In HCBA, resource and product components are regarded as the basic elements to make up a manufacturing system. These elements are similar to those introduced in the PROSA model [1]. *The resource component or resource holon* is a self-contained system component which can give treatments to works in process, such as fabrication, assembly, transportation, and testing. Typical resource components are machines, robots, AGVs, etc. Besides the visible physical part, a resource component contains an invisible control part which can perform its operations, decision making and communication ability by aid of its local database. On the other hand, *the product component or product holon* contains a physical part and a control part as well. A physical part may include raw material, parts and pallet/fixture. A control part may contain routing control, process control, decision making and production information.

The classification of these system components provides the potential to reconfigure a control system readily. Normally, a manufacturing plant is designed, integrated, and operated by different groups of experts. Resource and product components may be developed from different sources and maintained by different technologies and upgrade paths. From the view of component-based development, such a classification would provide a more coherent design of system components, each of which may be designed and maintained by adequate specialists. Moreover, the definition of system components in HCBA is also consistent with the nature of change. Two essential reasons for changes in the manufacturing plant are observed. The first is because of requirements of new manufacturing operations or technology. It results in upgrading or replacing the equipment. The second is due to the new requirements from customers. It leads to the release of revised or new products. The composition of a system with resource and product components is consistent with the way in which changes occur. It provides the potential to reconfigure a system with reduced effort. In addition, negotiation mechanisms [11,18] are available to support the interaction between distributed elements. Thus, the concept to implement the manufacturing control behaviour, such as scheduling and execution, by the co-operative decision-making between resources and products will be feasible.

2.3 Integration in HCBA

The integration of HCBA is described below by the aid of Figure 1. Initially, in HCBA, a holonic controller consists of a pool of separated and unorganised resource software components, which may be provided by resource vendors or equipment designers. They are one-to-one mapped to their associated physical equipment in the manufacturing plant. Thus, a resource holon contains these two main parts, a software part in the holonic controller for control and decision-making, and a physical part in the physical plant for actual fabricating. Each resource holon may not have the prior information of other holons. Therefore, it can not co-operate with other resources directly. Putting extended design into a resource holon so that it integrates with its neighbourhood resources is possible and straightforward if an overall system has been decided. However, the design of a holon will become more application-dependent and less cohesive in HCBA. Furthermore, the coupling effect between resources will become stronger. This leads to the fact that the resource holon can not be replaced or added to readily. This violates the requirements of HCBA. In order to avoid poor integration of dealing with long-term changes, resource holons are normally not allowed to communicate to each other directly in HCBA. As a result, the complete decoupled property of a resource holon could be obtained. This is called as *static integration* because there is not interaction between holons. This constrain brings the possibility that a resource holon can be replaced or added readily, without causing a global effect in this architecture.

The dynamic interaction between holons in HCBA is initiated once product holons are introduced. A product holon which may be provided by the product/process designers, or OEM customers, carries a detail process plan. The bill of material (BOM) may pass to the logistic team for them to prepare the associated raw material. The

software part will then be plugged into the holonic controller. The software part of the product holon will try to make use of resource holons by proceeding with a series of negotiations. This stage is called *dynamic integration*. The holonic controller will return to the stage of static integration if all product holons have finished their tasks; this is because no interaction is now generated.

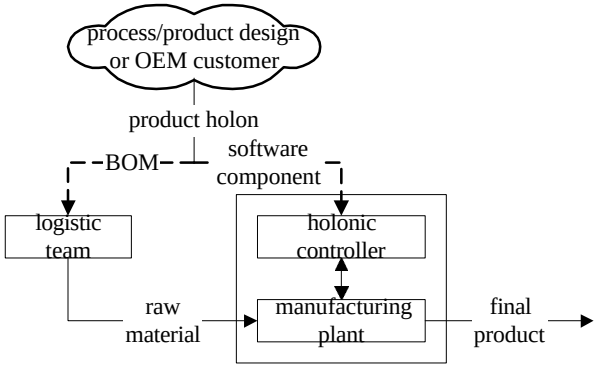

Figure 1. Operation of a holonic plant

3. Test Bed Description

A robot assembly cell which is used to implement the conceptual architecture introduced above is described in this section. The job of the robot assembly cell is to complete the assembly of a meter box. A Puma robot arm with vacuum suckers is employed to pick and place parts among buffers. A rotary table equipped with two jigs, is applied to turn through 180 degrees for swapping the position of each jig. A Hirata Cartesian robot which has an automatic screwdriver attached can assemble two separated parts put together into the jig. A flipper unit with two rotating clamps is utilised to hold a part and then flip it upside-down.

Three kinds of parts annotated as Parts A, B, and C are used to produce two kinds of products. One is denoted as Product AB, which is assembled from Parts A and B. Part A is the main housing of the meter box. Part B is a small access cover. Product AB is assembled by putting part B on top of part A and screwing B into A with a single screw. The other product annotated as Product ABC is assembled by attaching a transparent plate to the product AB by further four screws. The four screws attaching the transparent plate are fed in from the opposite side of the box to the single screw attaching the access cover. For this reason, it is necessary to flip Product AB during assembly using the flipper unit.

The control architecture of the cell is depicted in Figure 2. The cell PLC (Omoron C200HE) is employed to integrate all machines in this cell and then link upward to the resource controller. Two main reasons for using the cell PLC are

(1) to provide the control codes for implementing the operations of rotary table and flipper because they do not have their own embedded controller, and

(2) to be a bridge between two systems-the computing system and the physical plant for passing the real-time message.

Both a resource controller and a production controller make up the holonic controller to perform the manufacturing execution of the cell. A *resource controller* linking with the physical plant is employed to implement the static integration of HCBA. This controller could be placed on the shop floor. A *production controller* is applied to run software components of product holons, in order to generate the dynamic integration with the resource controller. Production controller may be situated in the design office to remotely control the resource controller via the Either net.

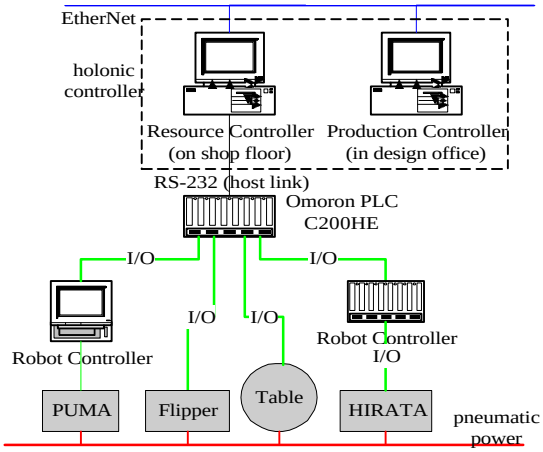

Figure 2. Control Architecture of a Robot Assembly Cell

4. Design Methodology

In this section, the design methodology to be used to implement the concept of HCBA into this testbed is proposed.

4.1 Software Infrastructure

Firstly, a software infrastructure is developed to migrate the existing PLC based control architecture to HCBA (see Figure 3). Two communication mechanisms, the *Blackboard Systems (BBS)* and the *Message Broker (MB)* are developed to support the operations of holons. The boundary of a holon is also identified. A detailed development of migration approach to the existing production control system and software infrastructure is referred to [2].

The BBS is designed as an intra-holon communication mechanism to link two parts of the holon, which are situated in the software and physical environment respectively. The BBS can be considered as a big blackboard which mirrors real-time messaging of the cell PLC. Each holon is allocated a range of areas in the BBS for updating its message. Therefore, the software part of a holon could access the information of the physical part

from its assigned area on the BBS. Moreover, the BBS is able to provide a simulation environment for testing the holonic design by exploiting the function to switch the downward communication channel from a physical machine to its virtual machine which is a piece of software to emulate the event behaviour of a real machine. As a result, each software component in this infrastructure does not know if it is communicating with the real machine or it virtual because the BBS can control the communication flow between them.

On the other hand, the MB is designed to provide an inter-holon communication mechanism. In order to provide enough information for the MB to manage the inter-holon communication, each holon should register with the MB when first logging in. Likewise, each should unregister with the MB before logging out from the system.

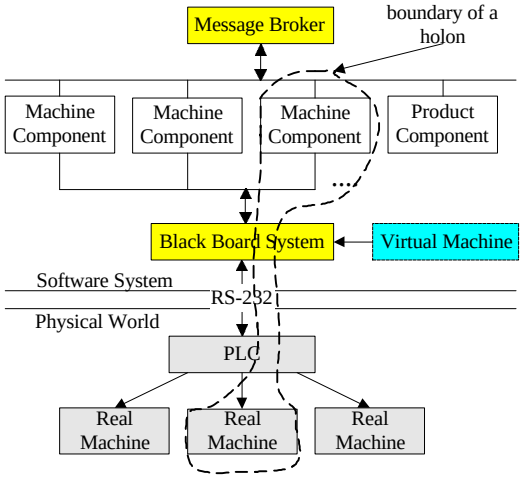

Figure 3 Software Architecture of HCBA

The BBS is placed in the test bed's resource controller, so that the BBS can straightforwardly interface the resource software components and the cell PLC. In order to balance the communication load, the MB is placed in the production controller to reduce the communication loading of the resource controller.

4.2 System Development Procedure

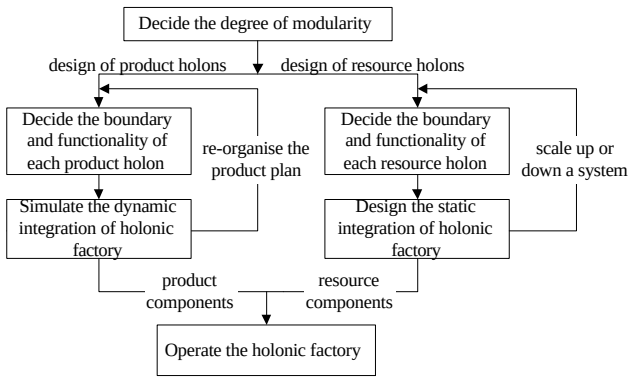

Figure 4. Procedure for Developing a Holonic Factory

The procedure for developing a holonic component-based system is shown in Figure 4. The holons in the robot cell are first decided before designing and integrating these basic components. The manufacturing equipment comprising the Puma robot, Hirata robot, rotary table, and flipper are regarded as the basic resource holons, while the assembled productions of AB and ABC are regarded as the basic product holons. The design of resource and product holons can be developed separately and concurrently afterwards. Moreover, the design of individual holons and integration of the application can also be achieved by different developers.

The design of a holon is presented next. The physical part is introduced first and then the development of the software part is described. The physical part of a resource holon which is placed in the cell includes two main elements. One is its physical mechanism, such as robots arms and gripper. The other one is the embedded machine controller. The controller can be used to control the operations of mechanisms to achieve the adequate manufacturing treatment. The controller can receive different recipes from its software part in order to perform different operations. However, in this cell, the physical part of a product holon is much simpler. As no extra accessory is attached, the unassembled part is the physical part of a product holon.

Furthermore, the software component situated in the PC environment is an extension of the physical part. It enhances the intelligence and autonomy of its physical part to make it capable of co-operating with other holons and achieving the holonic operations. The resource software part is placed in the resource controller to make up the virtual holonic controller. Meanwhile, the product software part is put in the product controller so as to operate the resource controller. The main functional modules of the software part of a holon are described below.

(1) Communication: This module helps its associated holon communicating with the BBS and the MB to assess the information with its physical part and other holons.
(2) Execution: This module is used to control and synchronise the execution sequence. The Petri-Nets model [5] is employed to describe the discrete events of machine operations and process sequences in resource and product holons respectively.
(3) Monitoring: This is the complementary function to the execution module. Both functions are used to deal with the real-time control. However, the execution module is used to perform the expected plan; the monitoring module is used to detect and diagnose an unexpected status.
(4) Logistics: The purpose of the logistic module is to maintain the life of a holon and keep all its parts available and working properly over time. This module does not involve the real-time manufacturing operation directly but may provide long-time observations for proactive management to predict the possible

degrading properties and protect the holons from potential breakdown or from defects during operation.

(5) Co-ordination: The co-ordination module is the co-ordinator of a whole holon. It can deal with the conflict of multiple operations commanded by other holons, and can negotiate with other holons to obtain its required service. The plan formed through the negotiation procedure will then be passed to the execution and monitoring modules.

The integration of resource and product holons is developed next. In HCBA, each resource software component maps to its physical equipment via the BBS. However, a product holon is responsible for controlling a batch of products which have the same manufacturing plan and process. Therefore, the product holon may contain more than one physical part. The number of parts is dependent on how many works in process (WIP) of this product there are in the plant. As different physical parts may enter the plant at different times and the availability of resource holons is changing over time, each physical part should be escorted individually. Thus, a *WIP agent* is generated dynamically by the product holon to manage the individual behaviour of each physical WIP part during its operation.

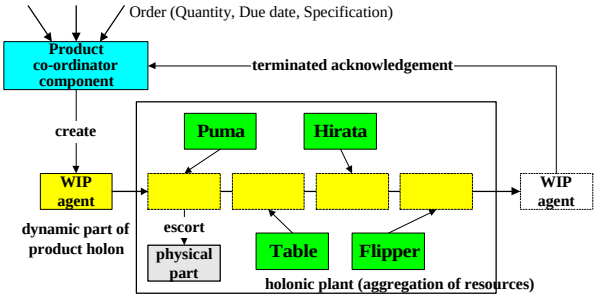

Figure 5. Interaction among product and resource holons

Figure 5 shows the dynamical interaction of product and resource holons. The original software part of the product holon which can dynamically generate WIP agents is regarded as the co-ordinator component in Figure 5. It is able to track the status of WIP agents it creates in the holonic controller and monitor the progress of the order. Furthermore, regarding the WIP agent, its mission is to escort a unit of physical part to pass through the manufacturing plant by following a given goal which is proposed by the co-ordinator component of the product holon. A WIP agent is responsible for negotiating with the resource community to decide the best manufacturing treatments during its life cycle. It is able to access the information or database from its associated product holon to decide its control strategy during operation. The WIP agent will leave the holonic plant and destroy itself when its associated product has been manufactured. As introduced above, a WIP agent includes communication, execution, monitoring, and co-ordination modules, while the logistic module is situated in the co-ordinator component.

The negotiation procedure, for executing a task between the WIP agent and the resource holon, is described in Figure 6. The procedure is controlled by the co-ordination module in the WIP agent and resource holon respectively. A WIP agent plays as an active role to negotiate with resource holons. A resource holon reacts like a server which provides a variety of services to their customers if the services are available.

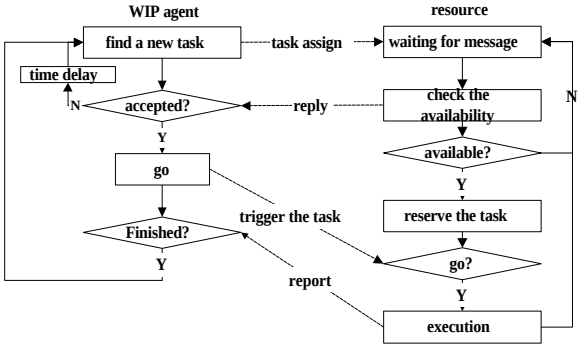

Figure 6. Communication procedure for a task

5. Verification and Discussion

The results of the testbed are verified and discussed below.

(1) A new way of system integration:
In order to implement the system development procedure, the design of testbed is deliberately separated into two groups to test the new approach of system integration. The first group is to develop resource holons and the static integration in the resource controller. The second group is to develop product holons and the dynamic integration in the production controller. The complete integration could be achieved after both groups have provided their final design. This approach verifies that the integration of HCBA which will not be dominated by one designer or a specific team is possible. System users are able to reconfigure the system by the self-contained and ready-to-run holonic components which could be provided by different experts.

(2) The communication infrastructure
Unlike the monolithic application in traditional design, this communication infrastructure builds the virtual control system by the loose relation of system components. The function of the infrastructure which can accommodate the self-contained system components has been tested in both the simulation and the real operation. The functionality is proved that it can migrate the PLC based manufacturing control system into the HCBA. However, it is also found that the inter-holon communication is not very efficient. The first reason is that a holon is designed to negotiate its task by the pooling method. Thus, if the negotiation is not successful, it will repeat it frequently until the task has been done. The other reason is that the message for real-time operations often flows from the physical plant to the software part of the resource holon via the BBS and then pass to the product holon via the MB. Likewise, the decided command may flow back by the reverse path to another resource holon. The overhead of communication seems to be high because of many intermediary mediums. The name

server function of the MB and the virtual wiring function of the BBS can be used to reduce the overhead of communication [2]. Moreover, the scheduling operation will be applied in the future work so that holons may obtain their operating time in advance by only once of negotiation.

(3) System reconfigurability:
A scenario is designed to test the system reconfigurability when the manufacturing equipment or the production process is changed. The testbed is initially operated to assemble the Product AB by using the Puma robot, the rotary table and the Hirata robot. Later, a new Product ABC is introduced into the testbed instead of the Product AB. The flipper unit should be employed to perform a new operation to achieve this plan. The change of the holonic controller is to plug the software component of the flipper unit into the resource controller and to plug the software component of Product ABC into the production controller instead of the component of Product AB. The manufacturing plan will be immediately changed without causing side effects to the other parts of the system. Excluding the effort to set up the physical parts, the effort to reconfigure the holonic controller is much less than that of previous design [?], which was based on Petri-nets modelling of the overall manufacturing behaviour. The property of component-based structure is therefore verified in the testbed.

(4) The ad hoc design
An ad hoc design is inevitable and necessary to meet requirements which are not specified in the original specifications. However, the ad hoc design is the main consequence of poor systems integration [14]. It will give a system more complexity and will make it become rigid over time. In our approach, an ad hoc design can be placed in the manufacturing plan of the product holon so that it can perform some special operations. When this product is no longer assembled, its associated component can be removed from the production controller. The legacy design not longer reside in the system. Thus, this architecture promises a longer life cycle.

References

1. Brussel, H. V., Bongaerts, L., Wyns, J., Valckenaers, P. and Ginderachter, T. V., "A Conceptual Framework for Holonic Manufacturing: Identification of manufacturing holons", *Journal of Manufacturing Systems*, Vol.18, No.1, pp.35-52, 1999.
2. Chirn, J-L and McFarlane, D. C., "A Migration of Holonic Approach to Production Control Systems", *1st IFAC Workshop on MAS'99*, Vienna, December, 1999.
3. Christensen, J., "Holonic Manufacturing Systems - Initial Architecture and Standards Directions", *First European Conference on Holonic Manufacturing Systems*, Hanover, Germany, 1994.
4. Cox, B.J., *Object-Oriented Programming - An Evolutionary Approach*, Addison-Wesley Publishing Company, 1986.
5. David, R. and Alla, H., *Petri Nets and Grafcet: Tools for Modelling Discrete-Event Systems*, London:Prentice-Hall, 1992.
6. Edwards, J., Clements, P., Gascoigne, J. and Coutts, I., "Component-Based Systems The Basic of Future Manufacturing Systems", *CUC96: Component-Based Software Engineering*, Cambridge University Press, 1998.
7. Gou, L., Luh, P. B. and Kyoya, Y., "Holonic Manufacturing Scheduling: Architecture, Co-operation Mechanism, and Implementation", *Computers in Industry*, Vol. 37, pp. 213-231, 1998.
8. Heikkila, T., Jarviluoma, M. and Juntunen, T., "Holonic Control for Manufacturing Systems: Functional Design of a Manufacturing Robot Cell", *Integrated Computer Aided Engineering*, Vol.4, pp.202-218, 1997.
9. Kidd, P. T., *Agile Manufacturing - Forging New Frontiers*, Addison-Wesley Publishing Company, 1994.
10. Koestler, A., *The ghost in the machine*, Arkana Books, 1989.
11. Markus, A., Vancza, T. and Monostrori, L., "A Market Approach to Holonic Manufacturing", *Annals of CIRP*, Vol.45, pp.433-436, 1996.
12. McFarlane, D., Marett, B., Elsley, G. and Jarvis, D., “Application of Holonic Methodologies to Problem Diagnosis in a Steel Rod Mill,” presented at *IEEE Conference on Systems, Man and Cybernetics*, Vancouver, Canada, 1995
13. McFarlane, D. C. and Bussmann, S., "Developments in Holonic Production planning and Control", *Submission to Production Planning and Control*, 2000.
14. Mowbray, T.J. and Zahavi, R., *The Essential CORBA systems Integration Using Distributed Objects*, Object Management Group, 1994.
15. *Proceedings of the First European Conference on Holonic Manufacturing Systems*, Hanover, 1994.
16. Ramos, C., "A Holonic Approach for Task Scheduling in Manufacturing Systems", *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pp.2511-2516, 1996.
17. Redman, J. and Cooper, E., "Component Software for Manufacturing Applications", *Proceedings of the Industrial Computing Conference*, Vol. 6, No. 1, pp. 67-72, 1996.
18. Smith, R. G., "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Transactions on Computers*. Vol.C-29, No.12, pp.1104-1113, 1980.
19. Suda, H., "Future Factory System Formulated in Japan", *Japanese Journal of Advanced Automation Technology*, Vol. 1, pp. 67-76, 1989.
20. Suda, H., "Future Factory System Formulated in Japan (2)", *Japanese Journal of Advanced Automation Technology*, Vol. 2, No1. pp. 60-68, 1990.
21. Sommerville, I., *Software Engineering*, 3rd Edition, Addison-Wesley Publishing Company, 1989.
22. Tanaya, P., Detand, J. and Kruth, J.-P., “Holonic Machine Controller: A study and implementation of holonic behaviour to current NC controller,” *Computers in Industry*, vol. 33, pp. 325-333, 1997
23. Zhang, X. and Norrie, D., “Holonic control at the production and controller levels,” *submitted to IMS'99*, Leuven, Belgium, 1999.

To appear in Proceedings of IEEE Conference on Robotics and Automation, San Fransisco, April, 2000