

OBJECT-ORIENTED ARCHITECTURE OF CONTROL SYSTEM FOR AGILE MANUFACTURING CELLS

Felix T. S. Chan

Department of Industrial and Manufacturing Systems Engineering
The University of Hong Kong, Pokfulam Road, Hong Kong
E-mail: ftschan@hkucc.hku.hk

Jie Zhang

School of Mechanical Science and Engineering
Huazhong University of Science and Technology, China
E-mail: zhangjie@hust.edu.cn

H.C.W. Lau and A. Ning

Department of Manufacturing Engineering
Hong Kong Polytechnic University

ABSTRACT

The depiction of the agile manufacturing cell includes a synopsis of some of the change proficiencies obtained by the configuration. To achieve agile configuration, the cell control system for agile manufacturing must be rapidly and efficiently generated or modified. In this paper, the object-oriented architecture is defined that supports design and implementation of highly reconfigurable control systems for agile manufacturing cells, which is composed of database objects, control objects, and resource objects, so as to reduce costs and increase of control system's agility with respect to changing environment.

Keywords: object-oriented methodology, petri net, cell control system, agile manufacturing

1. INTRODUCTION

Agile manufacturing demands a manufacturing system to be able to produce efficiently a large variety of products and be reconfigurable to accommodate changes in the product variety mix and product designs. The manufacturing cell reconfigurability is critical in agile manufacturing. The agile configuration here brings additional values such as rapid new product introduction, accommodation to unpredictable demand, longer equipment applicability, etc. Reconfigurability is the ability of a manufacturing cell to be reconfigured at a low cost within a short period of time. This is achieved by rearranging the machine tools of the manufacturing cell. A manufacturing cell consists of physical equipment such as machining equipment,

material handling equipment, computer hardware, etc. Although these components in a manufacturing cell are flexible in themselves and can be used in a large variety of applications, the software required for integrated control of these components must develop specifically for each manufacturing cell. In order to implement agile manufacturing, the highly flexible control system is demanded. Therefore, the concept of reconfigurability has an impact on control system of manufacturing cell in agile manufacturing environment. The cell control system should be rapidly and efficiently generated or modified. To meet these requirements, the properties of cell control software such as modularity, reusability, and maintainability must be considered and the architecture for manufacturing system should be used to describe how the control system should be designed and implemented.

The major aim in this paper is to define a reference model that supports design and implementation of reconfigurable control software for manufacturing cells. Therefore, the object-oriented architecture is presented to reduce costs and increase the control system's agility with respect to changing environment. In order to rapidly and efficiently implement controller, the resource model is established by integrating object-oriented method and Petri net.

The rest of the paper is organized as follows. In section 2, an object-oriented architecture of cell control software is presented. The database objects, control objects, and resource objects are described in detail in section 3. The steps of development of cell control software are described in section 4. Finally, conclusions are given in Section 5.

2. ARCHITECTURE OF CELL CONTROL SYSTEM

2.1 Object-Oriented Architecture

The architecture for manufacturing cell controller depends on control mode adopted. There are three modes to manufacturing cell control [1]: centralized, hierarchical [2-4], and heterarchical [5]. Diltis et al. [1] evaluate the performances of these three control models. In order to support design and implementation of cell control systems for agile manufacturing environment, our approach can be considered as a hybrid control mode where individual objects (machines, robot, etc.) control their own actions and communicate with each other while the coordination among objects is left to the cell controller. It has the advantages of combining simplicity of the hierarchical controller with the message-passing efficiency of the heterarchical controller.

A system consists of a set of interacting objects in the object-oriented world view [6]. The object-oriented technology provides a natural one-to-one correspondence between components in manufacturing system and software objects that represent them. Mize et al. [7] classified the objects in a manufacturing system into three main types: physical objects, informational objects, and control/decision objects.

A typical object-oriented software consists of mutually communicating objects and their interconnection relations. Unlike traditional software structure, object-oriented software structure creates models that exhibit close resemblance to the real world systems. A real world interaction between the components of a manufacturing cell becomes a dialog (via message passing) between the corresponding objects. Software model of cell control system captures the relationships or interaction between these objects.

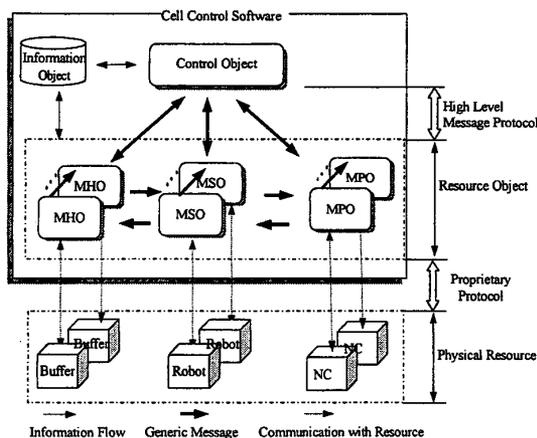


Figure 1 The object-oriented architecture of controller

This research focuses on the superior control-software in the cell level, not the lower level. The cell control software coordinates and controls the activities of the subordinate facilities. When designing cell controller, an object-oriented analysis and design method can be used to decide which component the system should consist of. The most probable outcome of such an object-oriented design of the controller is that each physical resource is represented by a specific software object in the controller called the resource object (RO). Therefore, an object-oriented architecture of cell controller is proposed, as shown in figure 1, which acts as the reference model for cell controller to design the special control architecture rapidly. The approach to implementing cell-control software is through the development of software library for objects.

In the object-oriented architecture, the objects are grouped in three types: control object (CO), resource object (RO) and information object (IO). The CO decides what to do and gives an order to execute the task. The RO executes specific resource tasks and keep track of the current state of physical resource.

2.2 the Feature of Architecture

The object-oriented architecture of cell control system may be characterized as follows: (a) The controller is object-oriented control structure with one object for each resource in the cell. Therefore, the flexibility of the control system will increase since objects can be replaced, reprogrammed, or added separately to the system without major changes. (b) The controller is of understanding of system configuration and function is simplified because the structure resembles the physical configuration of the system. (c) The controller has message-based, generic communication between objects. (d) The controller has objects suitable for reuse and therefore also be simpler to design and implement.

3. IMPLEMENTATION OF OBJECTS IN THE CONTROLLER

3.1 Database Object

A lot of information must be collected, stored, processed, and presented within a cell-control system. In order to increase reconfigurability and openness, the information should be stored in a database. True flexibility can only be achieved when different non-related aspects of a system are kept apart from each other as much as possible, so that a change only affects one part of the manufacturing system. Therefore, the information of the resources that are presented in the cell and the information of the products to be produced, should be kept apart. The linking between these is represented by the matching of product operation requirements with resource

operation capabilities. Thus, a new product should not affect anything else but the entering of a new product specification into the system, and likewise, a new resource should only add to the resource descriptions of the system.

Only some of the information needed by the control system will be briefly presented here: the product operation model, the resource model, and the handshake procedure information. The *handshake procedure* performed when products are loaded and unloaded in/from the procedures. The information in the database can also be divided into *static* and *dynamic* data. Static data is the templates which also exist before the individual products or resources are introduced into the cell. The dynamic data represents the individual products and the resources, that is, using the terminology of object-orientation, the instances of the templates.

3.2 Control Object

Control object comprises a knowledge base and some functional modules, as shown in figure 2. The knowledge base in turn consists of a domain knowledge base and a list of acquaintances. A domain knowledge base contains the essential data and knowledge required by the control object to perform its activity, such as scheduling rule, the capabilities of other objects. A list of acquaintances is the list of objects that it directly knows about. Control objects can communicate with those are in its acquaintance list. A list of acquaintances is related to the special manufacturing cell and is generated dynamically in running process for manufacturing cell so as to improve reconfigurability and openness of the object-oriented architecture. The knowledge component of the knowledge base might be continuously updated either by the object itself or by the system designer.

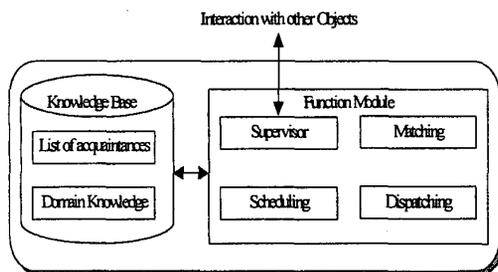


Figure 2 Control object

The function module contains Supervisor, Matching, Scheduling and Dispatching module. The supervisor module is responsible for interaction with other objects, and setup initial information based on the configuration of special manufacturing cell. The matching module matches product operation requirements with resource operation capabilities. It

produces some alternative schedules to the scheduling module. Based on the product and resource data found in the database, the scheduling modules develop a schedule on-line. The scheduling modules that handle the scheduling and control functions, including deadlock handling and optimization, so that a schedule is generated to dispatching module. The dispatching module decide what to do and give orders to the execution RO to execute the task; then supervisor control module send messages to related objects.

To implement rapidly cell-control software, the components of control object are stored in a repository, which allows for their logical organization.

3.3 Resource Object

3.3.1 Classification of resource

The object-oriented technology provides a natural one-to-one correspondence between components in a manufacturing system and software objects that represent them. For each physical resource in the manufacturing system there is a corresponding object in the cell controller called the resource object. These objects execute specific resource tasks and keep track of the current state of physical resource. The physical resources are classified according to their functionality: material processing resource, material storage resource, and material handling resource. Depending on what type of physical resource it represents, the resource objects are sorted in MPO (material processing object), MHO (material handling object), and MSO (material storage object).

3.3.2 Structure of resource object

Resource object comprises a knowledge base and some functional modules as shown in figure 3. The structure of the knowledge base is the same as the control object. A domain knowledge base contains the essential data and knowledge required by the resource object to perform its activity, such as processing times and set up times, the capabilities of other objects, type of message-passing. A list of acquaintances is the list of objects that it directly knows about. Resource objects can communicate with those who are in its acquaintance list. For A list of acquaintances is related to the special manufacturing cell and is generated dynamically in running process for manufacturing cell so as to improve reconfigurability and openness of the object-oriented architecture. For example, if a list of acquaintances in the MPO1 contains MPO2, MHO, MSO and CO, then the MPO1 can communicate with MPO2, MHO, MSO and CO; if a list of acquaintances in the MPO1 contains CO, then the MPO1 can communicate with CO. The knowledge component of the knowledge base might be

continuously updated either by the object itself or by the system designer. Between the controller objects, high-level messages are sent, but between each resource and its respective physical device, a message protocol, specific for the device, is used. To simplify the implementation, the function module of the resource object consists of a number of different software components is shown in figure 3. The various components are listed below:

- ↳ Supervisor
- ↳ Functional component
- ↳ Driver

Notice that these components exist in a hierarchical structure from supervisor through processor to driver.

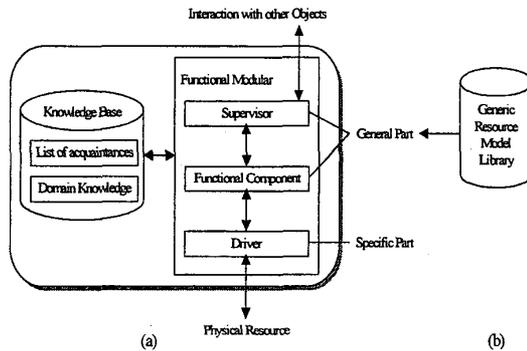


Figure 3 (a) Resource Object, (b) Generic Resource Model Library

Supervisor

The supervisor of resource object interacts with the other objects in the controller, exchanging information and requests for services. Its functions primarily to pass high-level messages into commands such that the internal components of resource object can utilize. It actually coordinates the action of the component related to the execution of incoming requests from other objects. The supervisor of a resource object breaks a request into one or more tasks, each with a priority and task identifier assigned. A task is then given to particular function for execution. In addition, supervisors also send message to other resource objects. Suppose that one manufacturing entity, such as CNC machine, needs to have service from a mover (e.g. Robot), specifically unload part, in order for this service to be performed, the supervisor of virtual CNC machine object communicates the request to the supervisor of virtual Robot object. The control object is provided a homogeneous message interface to all resource objects.

Functional component

A function is a functional component of resource

object that represents and accomplished some specific function. The function which is designed for a specific task, sends information requests and task commands to other functional component and/or execution to accomplish the task and returns appropriate information to the supervisor. A function is a set of operations of object. Design and implementation of function for resource object are guided by the function of physical object. In particular, the operations of resource object are identified in terms of state transition model or Petri net model of corresponding physical object [8].

Driver

Since the supervisor can handle only high-level messages and the physical resource can handle only device-specific, proprietary protocols, the messages have to be translated. The main function of the driver component is to translate the messages to and from the physical equipment. When the physical device lacks the functionality promised by the function of the virtual object, this functionality is implemented by the driver of the resource object.

All resources belonging to the same class have same supervisor and function. But each resource has specific execution. Therefore, resource objects are split in two parts, one general and one specific; see figure 3. Supervisor and Function is the General part, and Driver is the specific part. The implement of the general part of resource objects is based on generic resource model from a library.

3.3.3 Resource models

To support the design and implementation phases of the general parts of the resource objects, the general parts will be based on generic resource models, i.e. models describing resource belonging to certain classes. The generic resource model consists of the information model that includes resource descriptive information, resource status information, resource error-handling information, resource capability information and resource constraint information, and the control model that describes the dynamic behavior of resource and control logic. All resources can be controlled using the same interface and thus will be modeled in the same way. Our approach to faster development control systems is to create a library of generic resource models, as shown in figure 3 (b) to support the development of control systems. As a first step towards generic resource models we have analyzed thoroughly the resources included in cell system, then generalize their behavior into generic models. In this paper, the control model is represented OPN model that is established with integrating object-oriented method and Petri net model. For each physical object in manufacturing cell, the activities in it can be encapsulated and modularized by the proposed OPN. OPN describes

the dynamic behavior of resource object, control logic, exchanging messages with other objects. Using transformation rules, it is easy to convert an OPN model into software of resource object [8].

4. IMPLEMENTATION OF CELL CONTROL SOFTWARE

To illustrate the implementing cell-control software using object-oriented architecture, an example of manufacturing cell is presented. In the example, the establishing steps of development of cell-control software are highlighted. The example manufacturing cell is composed of two manufacturing centers and one loading/unloading station linked by robot, as shown in figure 4. The cell processes items such as steel part to produce an intermediate product. In this cell, some parts are processed first on lathe and then on mill, or first on mill and then lathe. Another parts are processed only one machine, such as lathe, or mill. Transport of parts between resources in this cell is supported by robot.

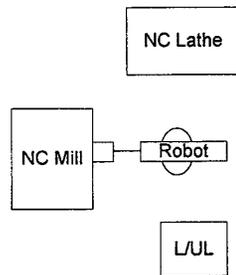


Figure 4 Example Manufacturing Cell

The establishing steps of development of cell-control software are as follows:

- 1) The first stage is to construct a special architecture of cell-control software by using object-oriented architecture in terms of special manufacturing cell. The special architecture of example manufacturing cell is composed of control object, database object, robot object, NC Mill object, NC lathe object, and L/UL object.
- 2) Once the architecture is carried out, the next step will reuse some component, such as database object and control object, from software library.
- 3) From generic resource model library that store OPN model for each resource object, the general part of resource is searched.
- 4) The special part of resource object is developed in terms of special physical device.
- 5) The newly created resource class is derived by inheriting characteristics of basic classes, yet contains the special part of a resource object.
- 6) Based on OPN models, message-passing between objects in control software is analyzed. Thus message-passing diagram is presented, as shown

in figure 5, which decides the execution sequence of message.

- 7) Using message-passing diagram, the main program is designed.
- 8) Encoding consists of the assembly of all components (reused, adapted and newly developed ones)

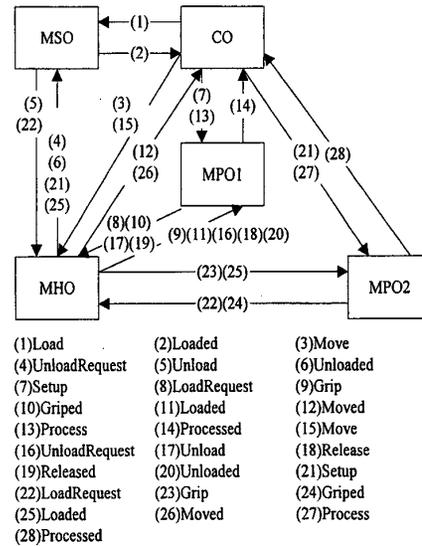


Figure 5 Message-passing Diagram

5. CONCLUSIONS

According to object-oriented method, this paper has presented an object-oriented architecture of cell-control software. It consists of a set of reusable objects, such as control object (CO), resource object (RO), and database object (DO). An important aspect of this architecture is that it supports the cell-control software of the agile reconfiguration because it has many properties such as modularity, reusability, and maintainability. The control systems therefore must be flexible and reusable in order for the whole system to survive changes in its environment. In order to implement the generic part of resource, the generic model of resource object is established by using Petri net so that it is easy to implement the change from model to control software and to support faster development of cell-control system.

REFERENCES

- [1] D. M. Dilts, N. P. Boyd and H. H. Whorms, "The Evolution of Control Architectures for Automated Manufacturing System," **Journal of**

- Manufacturing System**, vol. 10, no. 1, pp. 79-93, 1991.
- [2] M. R. Cutlosky, P. S. Fussel and R. J. Milligan, "The Design of a Flexible Machining Cell for Small Batch Production," **Journal of Manufacturing Systems**, vol. 3, no. 1, pp. 39-58, 1984.
- [3] H. Hammer, "Flexible Manufacturing Cells and Systems with Computer Intelligence," **International Journal of Robotics & Computer-Integrated Manufacturing**, vol. 4, no. 3, pp. 39-54, 1987.
- [4] J. A. Simposn, R. J. Hocken and J. S. Albus, "The Automated Manufacturing Research Facility of the Nation Bureau of Standard," **Journal of Manufacturing System**, vol. 1, no. 1, pp. 17-31, 1982.
- [5] N. A. Duffie, R. Chitturi, and J. Mou, "Fault-Tolerant Heterarchical Control Manufacturing system Entities," **Journal of Manufacturing System**, vol. 7, pp. 315-328, 1988.
- [6] C. Kim, K. Kim and I. Choi, "An Object-oriented Information Modeling Methodology for Manufacturing Information System," **Computers and Industrial Engineering**, vol. 24, pp. 337-253, 1993.
- [7] J. H. Mize, H. C. Bhuskute D. B. Pratt and M. Kamath, "Modeling of Integrated Manufacturing Systems Using an Object-Oriented Approach," **IIE Transactions**, vol. 24, pp. 14-26, 1992.
- [8] J. Zhang, K. Shi and P. G. Li, "Control system for agile manufacturing cells based on resource model," **Proceedings of Fifth International Conference on Mechatronics and Machine Vision in Practice, M2VIP'98**, Nanjing, China, pp. 327-332, September 10-12, 1998.