공학 연구논문집 제29권 2호 pp. 511~517, 울산대학교.
Journal of Engineeringl Research Vol. 29, No. 2, pp. 511~517, 1998
University of Ulsan.

DESIGN OF FMS CONTROL SYSTEMS

Yoonho Seo
yhs@uou.ulsan.ac.kr
†Dept. of Industrial Engineering, University of Ulsan,
Namku, Ulsan, Korea (680-749)

<ABSTRACT>

This paper addresses the development of a model for FMS control systems. The FMS control systems exercise their control by tracking the states of the parts. Part state transition diagram is defined and derived to express part states and their transition from process plan. Also, the methods to generate/exchange messages between shop- and workstation-level control systems are specified using "part state transition".

INTRODUCTION

For an unmanned or semi-automated manufacturing system that is designed to gain both flexibility and efficiency through production of arbitrary mix of various parts, a control system plays a critical roll. It should not only coordinate and monitor the complex interaction of component's operations, but also make decisions in every stage of planning, scheduling and dispatching. Also, it should be generic in the sense that it can be used in any variety of shop floor configuration and part mix.

Since a lot of research attentions have been paid to develop manufacturing control system models, two approaches are used: one is to define a language that is capable to describe factory floor and process plans (Bourne 86, Graham & Saridis 82) and the other for formal model of factory floor behavior (Mettala 89, Naylor 87, Smith 92). Since the most are based on the resource model and their behavioral description, it is still unsolved problem to provide a generic model of process plan interface to the control system (Lee et al. 94).

The goal of this research is to present an alternative modeling method of manufacturing control systems which exercise their control by tracking the state of the parts in shop. Rather modeling the shop floor behavior, this paper focuses on formalizing the part states and defining the manufacturing task. Part state transition diagram is modeled as a finite state machine in which all the state changes are defined with the awareness of the message sending or receiving. Finally, manufacturing control model to use the part state transition is presented.

GENERIC CONTROL ARCHITECTURE OF FMS

A control architecture functionally describes the control components of FMS as well as organizes their complex interactions. Basically, centralized, hierarchical, and heterarchical control models have been identified. Although the heterarchical control demonstrates the control architecture to be well fitted into the distributed systems, the use of a hierarchical model is dominant because of the need for global information for planning and scheduling (Smith 92).

Generally, automated manufacturing systems are composed of four classes of components such as material processors, handler, transporter, and storage. To be completed, material iteratively goes a cycle to be processed at a processor and moved to the next processor. A manufacturing unit which is composed of multiple material processors and single handler/transporter can be found in the manufacturing system hierarchy. For example, an FMS can be composed of multiple cells and an AGVS which interrelates the cells. Also, an automated manufacturing workstation can include the multiple machining centers and material handling robots.

As depicted in Fig. 1, a generic two-level control architecture of an FMS is proposed. The manufacturing system level represents a group of logical or physical equipment with direct interaction with one another. The proposed control architecture is generic in the sense that it can be mapped into any manufacturing control unit which shares material handler, *i.e.*, the manufacturing system-level control module (MSCM) in the architecture can be a control system module of *factory-*, *shop-*, *cell-*, *workstation*-level in the manufacturing hierarchy only if the level is composed of multiple material processors and material mover.

Functions of manufacturing control systems can be classified into *forward* functions which are responsible for planning, scheduling, executing the production orders and *backward* functions for status monitoring, material tracking and performance recording. In the 2-level architecture, the manufacturing system control module is responsible for decomposing the production orders into lower-level material processing and handling (including transporting) tasks, sequencing them in order to meet manufacturing goals, and executing them through the message exchange with lower-level modules, MPCMs and MHCM.

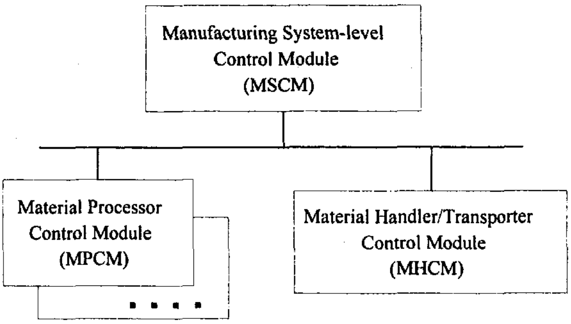

Fig. 1. Generic 2-level architecture for automated manufacturing systems

FMS CONTROL MODEL

The manufacturing task of FMS is defined as the total tasks required to perform the production order assigned to FMS not accomplished yet, i.e., the sum of the remaining work contents for the all parts in progress in FMS. The manufacturing task is intrinsically dynamic in that FMS receives production orders from the upper level planning system and continuously performs the remaining production orders. Part state transition diagram (PSTD) is used to express the progress of single manufacturing task for each part. The total manufacturing tasks for all parts is dynamically defined by using all the PSTDs for all parts. Part state transition list (PSTL) is a data structure which contains information on all the PSTDs for all parts.

Process Plan Representation

A process plan provides the essence of what must happen in a manufacturing system in order to produce parts. To be used in FMS whose control components are hierarchically organized, process plan should be hierarchically expressed by the lower level components whether they are logical or physical (Lee et al., 94, Derebail et al., 94). In the context of the 2-level generic control architecture, a manufacturing system-level process plan for a part is represented as a series of lower level material processors visited and their manufacturing instructions.

Since parts to be manufactured have a typical operational structure constrained by sequentially and/or parallelly related operations, a process plan representation should have capability to represent all possible precedences that occur among the planning and processing decisions. An AND/OR directed graph is used to represent the sequential and parallel structure of machining processes. Let g_{ki} and m_{ki} be respectively *i*th operation of part *k* and selected machine tool (or material processor, MPCM) for operation g_{ki}, where g_{k0} means 0^{th} operation (i.e., raw material state) of part *k*. For instance, let us assume that there is simple process plan that composed of operation set $\{g_{11}, g_{12}, g_{13}, g_{14}\}$ and precedence set $\{g_{11} \rightarrow (g_{12} \ \& \ g_{13}), (g_{12} \ \& \ g_{13}) \rightarrow g_{14}\}$, where $a \rightarrow b$ means that operation *a* must precede operation *b* and & means logical AND. The process plan is graphically represented as AND/OR digraph as in Fig. 2. In the Fig. 2, feasible sequences are $\{g_{11} \rightarrow g_{12} \rightarrow g_{13} \rightarrow g_{14}, \ g_{11} \rightarrow g_{13} \rightarrow g_{12} \rightarrow g_{14}\}$.

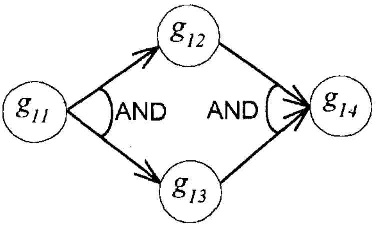

Fig. 2. AND/OR graph representation

Part State Transition Diagram

From the controller's point of view, information needed to control the manufacturing task to process a part in FMS is process of part in progress, position of part, and command issuance. Through combination of these three pieces of information, part state is represented, p e.g., *part in input_buffer of workstation_#3 for milling operation of pocket_#2*. Part state transition diagram which can be derived from the process plan defines the part states and their transitions. A part in the state of *part in input_buffer of workstation_#3* satisfies the necessary condition to transit the next state of *part in loading into the machine_#2 of workstation_#3 by the handling robot of workstation_#3.* For a part in the state *"part in input_buffer of workstation_#3"*, a MSCM which is responsible to issue and distribute the messages to MHCM generates a message *"move the part_id from input buffer to machine_#2 in workstation_#3"*. The recognition of the message issuance results in the state transition to the next state in part state transition diagram. Let a part state s_{kimn} be the state in which part *k* is in *m* (position) and *n* (message issue) after ith operation, where

$$m = \begin{cases} 1, \text{ if part is in output buffer of machine and} \\ 0, \text{ otherwise.} \end{cases}$$

$$n = \begin{cases} 0, \text{ if the command dose not issue yet} \\ 1, \text{ otherwise.} \end{cases}$$

For the process plan in Fig. 2, the part state transition diagram is derived and depicted in Fig. 3. The PSTD is a finite state machine with the initial state s_{1010} and the final state s_{1500}. When the message is sent to (or received from) lower-level control modules in the node state with s (or r) labeled arc, the state transit to the next state. Part state transition diagram expresses information on a life cycle of workpiece to reach to output port through a series of processing and moving.

In the Fig.3, the part state s_{1010} means that the part in raw material state is in system input port and does not issue the command *"move part_id from input port to $I(m_{11})$"*. When the command is sent to MHCM, the state changes into s_{1011}. The state lasts until the response from MHCM *"move-completion"* is coming.

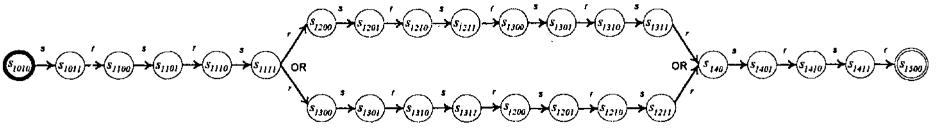

Fig. 3. Part state transition diagram

Message Issuance

Using PSTDs, MSCM can generates and send the messages to MPCM or MHCM in the generic 2-level architecture. Algorithmic description to generate messages and change part states is presented below. Notations $O(m_{ki})$ and $I(m_{ki})$ are introduced to identify the output and input buffer of machine (or MPCM) m_{ki}.

For the Current State s_{kimn},

- if (m = 0 & n = 0), *send_message "process part_k" to* m_{ki} *and change_the_state_to* s_{ki01}.
- if (m=1 & n=0), send message *"move part_k from $O(m_{ki})$ to $I(m_{ki+1})$"* to MHCM and *change_the_state_to* s_{ki11}.
- if (m = 0 & n = 1), *wait_for_response_from* m_{ki}.
- if (m = 1 & n = 1), *wait_for_response_from* MHCM.

For the response r from m_{ki} or MHCM.

- if (r = *Process_Completion_of_*g_{ki}), *change_the_state_to* s_{ki10}.
- if (r = *Move_Completion_of_*g_{ki}), *change_the_state_to* $s_{k(i+1)00}$.

Message Exchange

Basic operations to exchange messages are send and receive. If no message is present when a receive is executed, the receiver waits until a message is sent.

Messages are sent to and requested from special processes called link processes (message queue). Link processes provide what is essentially a bag of messages which have been sent and not yet received or requests for messages from receives which have been made but not yet satisfied. Figure 4 conceptually expresses that the MPCMs, MHCM and MSCM exchange messages through link processes L1, L2 and L3. The link process L1 is for transferring the message from MSCM to MPCMs and MHCM, and L2 is used in reversed case. Also, link process L3 is used to send messages to a monitoring system.

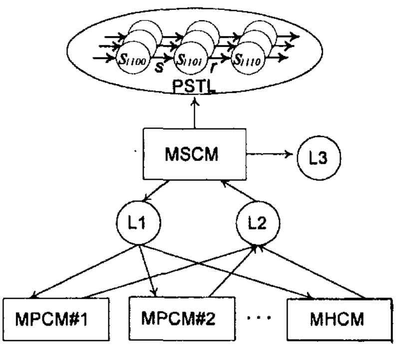

Fig. 4. Message exchange between MSCM and MPCM/MHCM

Message exchanging mechanism between MSCM, MPCM and MHCM s described below.

MSCM

```
A1: Set COMMAND;
        state_change PSTL;
        send L1;
        receive L2;
        unless RESPOND A2;
        state_change PSTL;
        go_to A1;
A2: send L3;
End;
```

MPCM or MHCM

```
A1: receive L1;
    if_internal_test A2;
    set RESPOND;
    go_to A3;
A2: set MONITOR;
A3: send L2;
    go_to A1;
End;
```

CONCLUSIONS

This paper presents an alternative modeling method of FMS control system design. To describe the control model of FMS, generic 2-level control architecture is defined. Since process plan is also represented by AND/OR directed graph, it is used to express a general structure of machining precedence. Rather developing a formal model of shop floor behavior, this paper focuses on the part state transition and message generation mechanism. A FMS control model is proposed by using part state transition diagram. The part state transition diagram contains information on a life cycle of workpiece to reach to output port through a series of processing and moving. The proposed control model is intended to be independent of shop floor configuration and part mix. The control model is experimentally implemented on a group of micro computers on TCP/IP network.

REFERENCES

1. Bourne, D. A. (1986). CML: A Meta-Interpreter for Manufacturing. *AI Magazine*, 86 Fall, 86-96.
2. Derebail, A., H. Cho and R. Wysk (1994). Process plan representation for shop floor control. *Computer control of flexible manufacturing systems: research and development* (S. B. Joshi and J. S. Smith Ed.), Chapmand & Hall, 379-404.
3. Graham, J. H. and G. N. Saridis (1982). Linguistic Decision Structures for Hierarchical System. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-12(3), 325-333.
4. Lee, S., R. A. Wysk and J. S. Smith (1994). Process plan interface for a shop floor control architecture for computer-integrated manufacturing. *International Journal of Production Research*, 33(9), 2415- 2435.
5. Mettala, E. G. (1989). *Automatic Generation of Control Software in Computer-Integrated Manufacturing*. Ph.D. Thesis, The Pennsylvania State University.
6. Naylor, A.W., and Volz , R. A. (1987). Design of integrated manufacturing system control software. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17(6), 881-897.
7. Smith, J. S. (1992). *A Formal Design and Development Methodology for Shop Floor Control in Computer Integrated Manufacturing*. Ph.D. Thesis, The Pennsylvania State University.
8. Xiang, D. and O'Brien, C. (1995). Cell control research. *International Journal of Production Research*, 33(8), 2325-2352.