'93 KACC (1993.10.20~22)

Effective Modular Assembly Line Control System

° Du-Gan Eom, Dong-Shin Kim, In-Seok Ahn, Jong-Oh Park,
Young-Tae Kim* and Sung-Kwon Kim*
Robotics & Fluid Power Control Lab., KIST
* FA Research Institute, Samsung Electronics Co.

ABSTRACT

In this paper, a new design method for Assembly Line Control System(ALCS) is presented. This system consists of five independent modules having their own specific functions such as production management, facility management, material management, quality management, and remote control. To implement the ALCS, we propose design of the common data management module(CDMM). This module has the roles of integrating the above five modules and of communicating the common data between them. Using this method, we realize an information management method in the view of CIM. In addition, we standardize the inter-communication of common data between machines having different interface protocols.

1. Introduction

In recent years, the performance of electronics assembly line is beyond the level of robotics automation and is on the level of rational line structure design and coordination of information in the field of manufacturing. So the necessity of system change to coordination of line information and effective line management is highly required. The performance of various production management system have been consecutively developed for a long time. To adjust production management system to various assembly line , we must have modified and newly developed in some parts in considerations of its characteristics of dependency.

In this paper, we suggest the multi-module assembly line system which is divided into independent modules according to their functions. Therefore it provides the optimal solution to the object assembly line with the advantage of maintenance, repairmen and expandability. The environment of this assembly line control system uses MS-WINDOWS and C++ language which is nowadays the standard of industries to improve the efficiency and the convenience of user interface. The reliability of this system is an another important point of view because this assembly line control system is not developed for laboratory level but for the commercialization.

2. System Configuration

The overall environment of Assembly Line Control System (ALCS) is shown in Fig.1.

The overall layout of Assembly Line Control System is shown in Fig.2.

The basic concepts are as follows.

1) The development of ALCS is focused on the generalization of system.
2) Both automatic input of information for external information management system and manual input of information for stand alone operation are implemented.
3) System development environment is as follows.
   -. SYSTEM H/W : PC 386 DX
   -. SYSTEM S/W : MS-WINDOWS 3.1
   HANME HANGUL for WINDOWS
   -. Development TOOL : MS-WINDOWS SDK
   MS-EXCEL Macro, MS-C/C++
4) The communications of information between modules are executed by CDMM.

5) The communications of information modules are performed by Software Bus.
6) When the information is to be reported, the required data is saved automatically to the file from CDMM, and EXCEL is executed automatically.
7) This system is modularized for system maintenance, repairment and expandability.

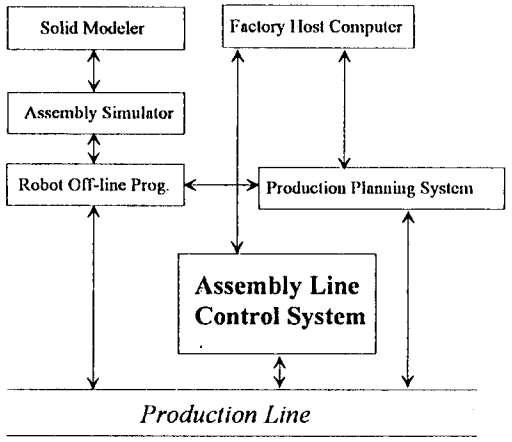

Fig. 1 Environment of the Assembly Line Control System

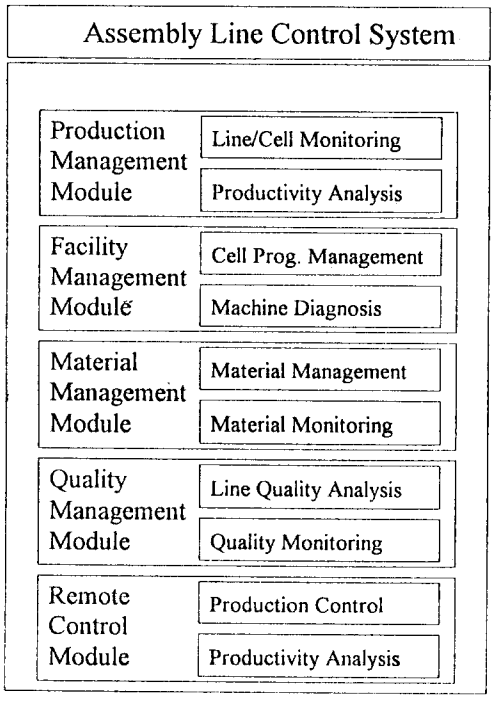

Fig. 2 Layout of Assembly Line Control System

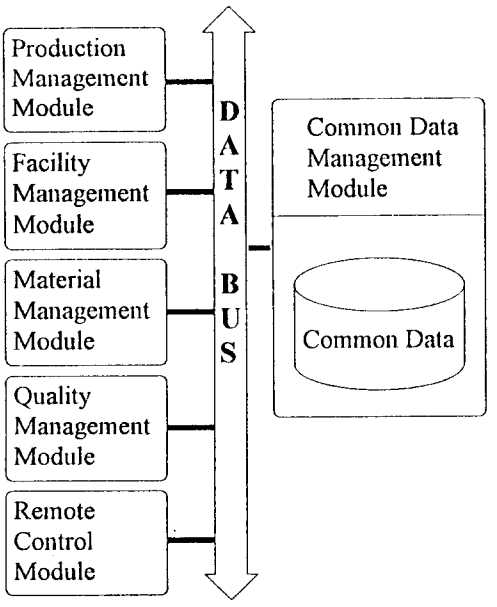

Fig. 3 Data Exchangement between Modules

3. Common Data Management Module(CDMM)

3.1 Algorithm for CDMM

Defining the data processing procedure consistently gives better expandability of the common data for further use.

Input/Output Device

The contents of I/O are the transferred messages and the data required or transferred after analyzing the messages.

Processor

1) Common Data

-. Data Space : the collection of data shared by the modules in use.

-. Data Map : Information of the given priority to each item and mutual relations for deciding necessity of updating.

-. Data Flag : Symbolized information of item for its state during task.

2) Message Analysis Procedure

: After referring and analyzing data map and flag, further transaction is confirmed and the required treatments are decided.

3) Data Generation Procedure

: The data to be stored or transferred is generated according to the result of the message analysis.

4) Data Retrieve Procedure

: If a certain data is to be retrieved, that data is searched from common data.

5) Data Store Procedure

: If a certain data is to be stored, it is stored in common data.

6) Data Transfer Procedure

: If a certain data is to be transferred, it is transferred to the related module.

7) Data Receive Procedure

: CDMM received the data from the other modules.

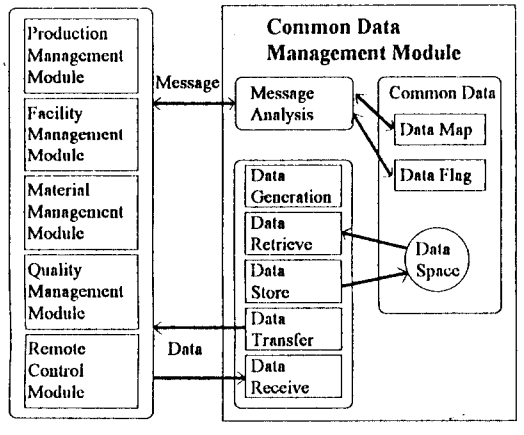

Fig. 4 Flow chart of Common Data Management

3.2 Structure for CDMM

Common data is to be accessible to high priority client and needs to be generalized. It is necessary to make the data format consistently for multiple accesses from clients upper level. To provide all the clients with the same access protocol, the three common data components are suggested.

-. Data Space
-. Data Map
-. Data Flag

Referring to data flag and data map, common data management module accesses to the data space where the common data is stored. CDMM can read or write at any cell of data space exclusively.

Data map and data flag is initialized when CDMM is executed, and according to this, data space is dynamically allocated, data request and store signal is generated, and the contents of data space are generated or deleted.

Data Space

Data space is where common data is stored, and is made up of two-dimensional array to implement tree structure according to data priority. If time axis is added for schedule management for job replacement, three-dimensional array is required.

Required memory is allocated dynamically according to the priority level calculated from data map. The data space has the following memory space scheme.

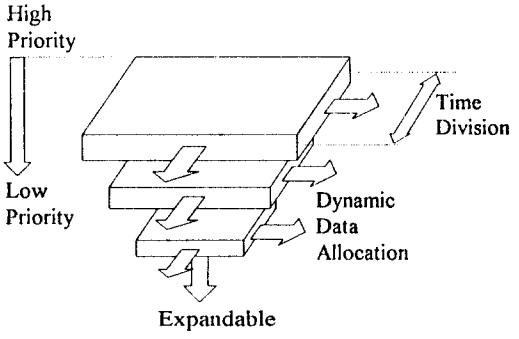

Fig.5 Structure of Data Space

Data Map

1) Structure of the Data Map

: Data map is a kind of database that defines the relations of the various data in data space, so the database index table sets the globally defined variable names as key fields. Index table is a binary tree structure for faster processing. Common data module can recognize the meaning of information stored in data space, referring to data map via index table.

Data map also serves as a useful tool for using data space more consistently. Data map is initialized by reading the information on the data map stored in the permanent data storage when CDMM is executed. The information on the data map can be inserted or updated through user interface supported by CDMM. Shown below is the structure of data map and the implementation action method.

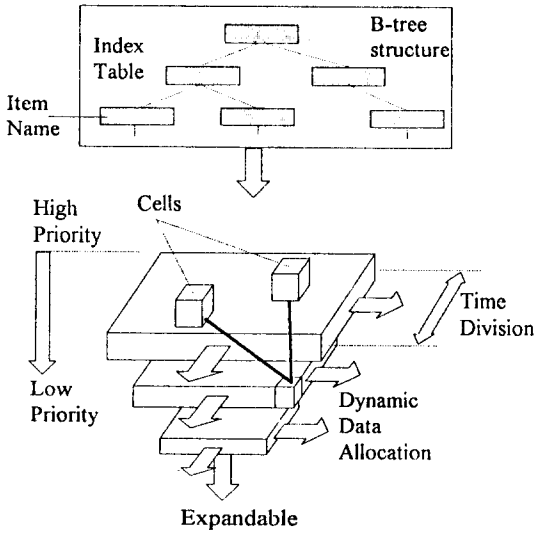

Fig. 6 Structure of Data Map

As above, data map consists of three-dimensional array cells that define the cell relations, and is initialized referring the data storage when CDMM is executed. Four kinds of information are stored in each cell of data map.

-. related cell address of high priority
-. mutual semaphore information
-. binary semaphore information
-. count semaphore information
(updating count of all)

Mutual semaphore contains information of whether a given cell is included in a certain data set and it also contains information on the data set number on data flag.

Binary semaphore is used when synchronization of two or more higher priority client modules are required. Data map manages the binary semaphore and its state(on/off), and when binary semaphore is set to off, it sends signals pending to high priority client which is waiting for data. Using data map which contains the information mentioned above, CDMM executes as follows.

-. When high priority client requires data, CDMM refers to index table and finds the location of the cell in the map database of the corresponding item.

-. Referring to the cell address of the higher priority, CDMM gets count semaphore from each upper level cells.

-. If the count semaphore of the high priority level cell is less than the count of the required cell, client access to CDMM is denied and pending signal is given to client module.

-. If the count semaphore of the higher priority level cell is greater than the count of the required cell, data space is accessible in use of data retrieve function or data store function.

-. When data transfer to higher priority client is required , data is transferred after modified to meet DDE protocol by data transfer function.

2) Data Map Implementation Method

: Two important information stored in data map are processed as follows.

-. Information on the count semaphore of the corresponding cell in the data space.

-. cell address in the corresponding higher priority level.

: As the cell address in the high level can be represent ed by two-dimensional integer, it is allocated dynami cally in the structure type variable containing two inte gers at permanent data storage. The first byte of the dynamically allocated memory represents the allocat ed space of the corresponding cell, and if the reference to the address of the higher priority level is required, CDMM gets the number of the related cells from the first byte, and can refer the number of count sema phore of the corresponding address of higher priority level. The structure is as follows.

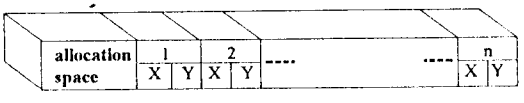

Fig. 7 Structure of Address Table

Data Flag

Binary semaphore and count semaphore are mentioned in data map, and are omitted here. Mutual semaphore is used to avoid data conflict when two or more upper level client module attempt to access the

data space at the same time. Most of upper level client module has to get the right to access data set which is protected by mutual semaphore if necessary. The data format related to mutual semaphore is as follows.

-. data set which needs simultaneous access for their mutual relation
-. data set which needs sequential access

Data process mechanism for implementing mutual semaphore is as follows.

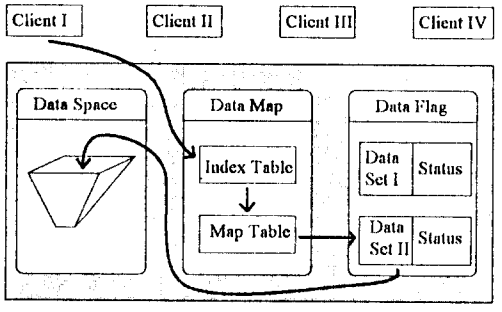

Fig. 8 Data Processing Mechanism of Mutual semaphore

As seen above, if CDMM gets the data requirement from high priority client module, CDMM refers to index table, searches the corresponding cell in map database, and decides whether the data is included in the data set. If included, data space is available only when the mutual semaphore is set to off after referring data flag and the condition of the corresponding data set.

If the corresponding data set is already occupied by other client module, common data module gives pending signal to the data requiring client and when the semaphore is released, common data module resumes the client module. The mutual semaphore structure is as follows.

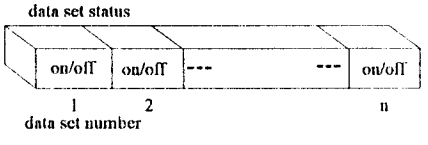

Fig. 9 Structure of Mutual semaphore

4. Summary

In this paper, the concept of Assembly Line Control System which is developed into several independent modules for control function is suggested. Entire system has been developed and implemented to substantial manufacturing field. the important issues of modern Line Control System, partially constructed in this paper are as follows.

- Information management in the view of CIM
- Standardization for data intercommunication between machines offering deferent interface protocols
- Modularization of each control functions and organic integration of each modules
- Common data management module is necessary for inter task communication between several modules according to their unique functions

The concept of CDMM is recommended and developed to support multiple task operation and real time line control in MS-Windows environment because of its user friendliness and easy operation facility.