Workstation Control in a Computer Integrated Manufacturing System

May 1984

Harry Scott
Kathleen Strouse

National Bureau of Standards
Manufacturing Systems Group
Washington, D.C. 20234

WORKSTATION CONTROL IN A COMPUTER INTEGRATED MANUFACTURING SYSTEM

1. INTRODUCTION

The Manufacturing Systems Group of the National Bureau of Standards has developed a real-time control system to coordinate the equipment in a machining workstation.

The control concept used is that of decomposing high level tasks through several levels of workstation control into commands which can be issued from the lowest level and acted on by the equipment components in the workstation. A finite state machine has been implemented to perform the task decomposition at each level of workstation control. Reference 1 discusses this control concept further.

The Workstation Controller (WSC) architecture was designed using the following guidelines:

1. The WSC uses the task decomposition techniques previously mentioned for control.

2. A finite state machine implementation of task decomposition is employed.

3. Although the current implementation of this WSC is in a horizontal machining workstation (see Section 2), the architecture is generic. That is, the controller is applicable to other real-time control tasks.

4. Consistent with #3, a set of software tools was developed for building such control systems, and then used to create a Horizontal Machining Workstation Controller (HWSC). All of the tools are available in the completed controller, allowing easy modification.

5. The control system is partitioned into modules that each do a small and comprehensible amount of processing and that have clearly defined interfaces to other modules.

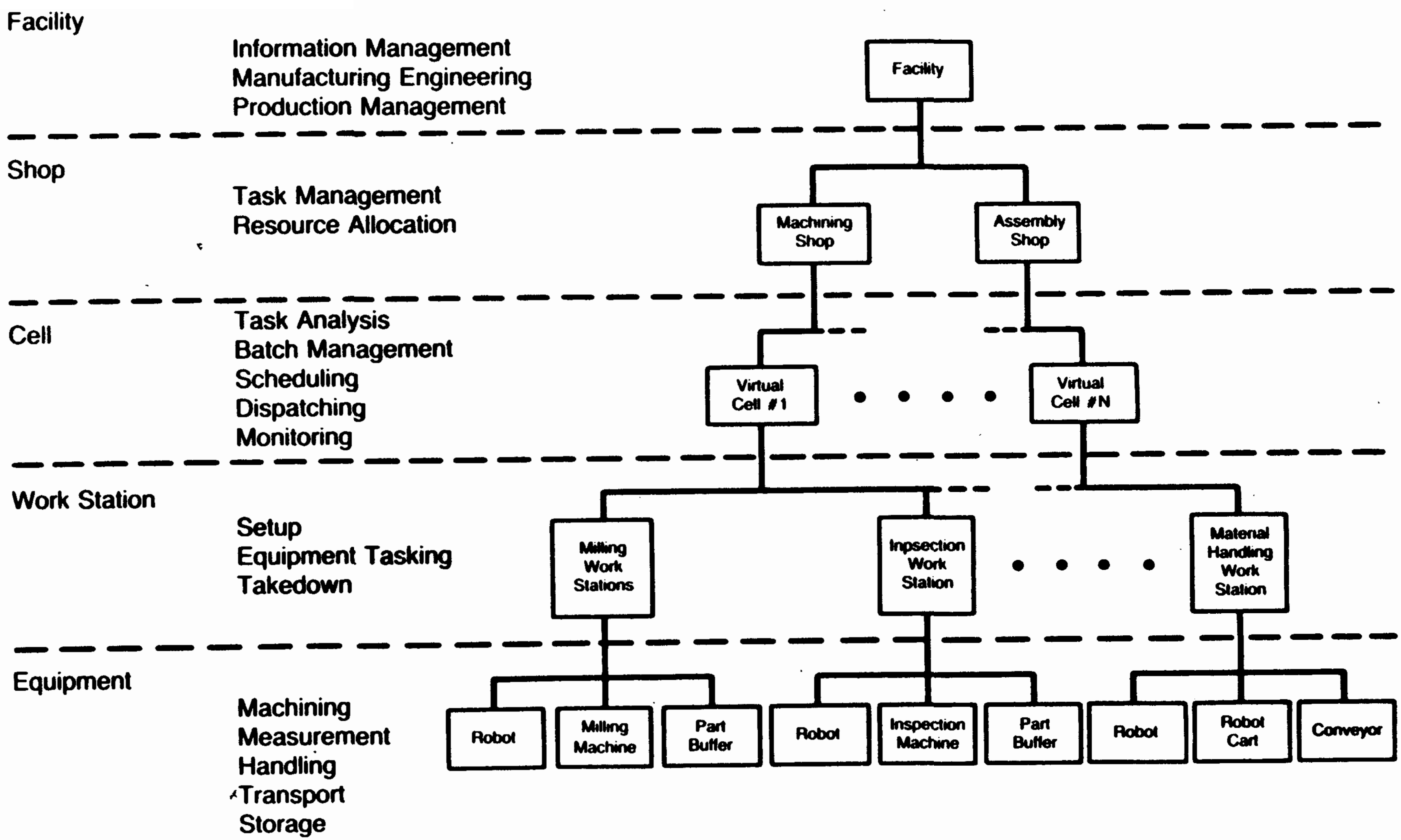
6. The system is highly interactive. All modules are individually executable from the keyboard to aid in debugging. This is true for the low level modules and for the higher level modules that execute groups of low level modules. Control levels may be single stepped and the entire controller may be single stepped.

7. Interfaces between modules are easily examined during development and execution.

8. In multiple-level controllers (the HWSC has three levels), all levels have the same architecture.

9. The architecture supports changes in the equipment components in the workstation and is easily extensible to incorporate new sensor systems and such software additions as new equipment component commands or status feedback information.

The AMRF Control Hierarchy



A control heirarchy for an Automated Manufacturing Research Facility (AMRF) being constructed at the National Bureau of Standards.

Figure 1.

Figure 3 shows the organization of these levels and some of the commands provided to each level. Each level functions the same way. During each control cycle each level examines the command from the level above, the status feedback from the level below, any sensory information available, and its internal state. Then based on this information, it issues a command to the next lower level and a status report to the next higher level, and updates its internal state. This is repeated each control cycle for every level. The selection of the proper outputs based on the input information is performed via a state table search (Ref. 10). Figure 4 shows how a state table is used to implement such a control decision level.

As an example of the task decomposition, consider the execution of a "MAKE_BATCH partname" command received by the HWSC.

Level 0 is responsible for coordinating the retrieval of control data (state tables) with the production of the part. Since the state tables are essentially the task program for making this particular part, they must be retrieved from the Data Base System before any other operations take place. This retrieval is performed by the Process Initialization Module. When the control data retrieval is complete Level 0 issues a MAKE_BATCH command to Level 1.

Level 1 decomposes the MAKE_BATCH command into pieces that reflect major portions of the overall task. These are LOADing the part into the machine, one or more MACHINEing operations on the part, REFIXTUREing the part if needed for various machining operations, and UNLOADing the part into an outgoing tray. These commands are issued to Level 2 when appropriate.

Level 2 decomposes the particular command received into equipment level commands needed to perform it. For example, when a LOAD command is received by Level 2, an APPROACH command is sent to the fixturing and a POSITION PALLET B command is sent to the machine tool. After a few more commands to the robot and fixturing, the part is fixtured in the machine and the LOAD operation is completed. The status being sent to Level 1 is then changed from EXECUTING to FINISHED.

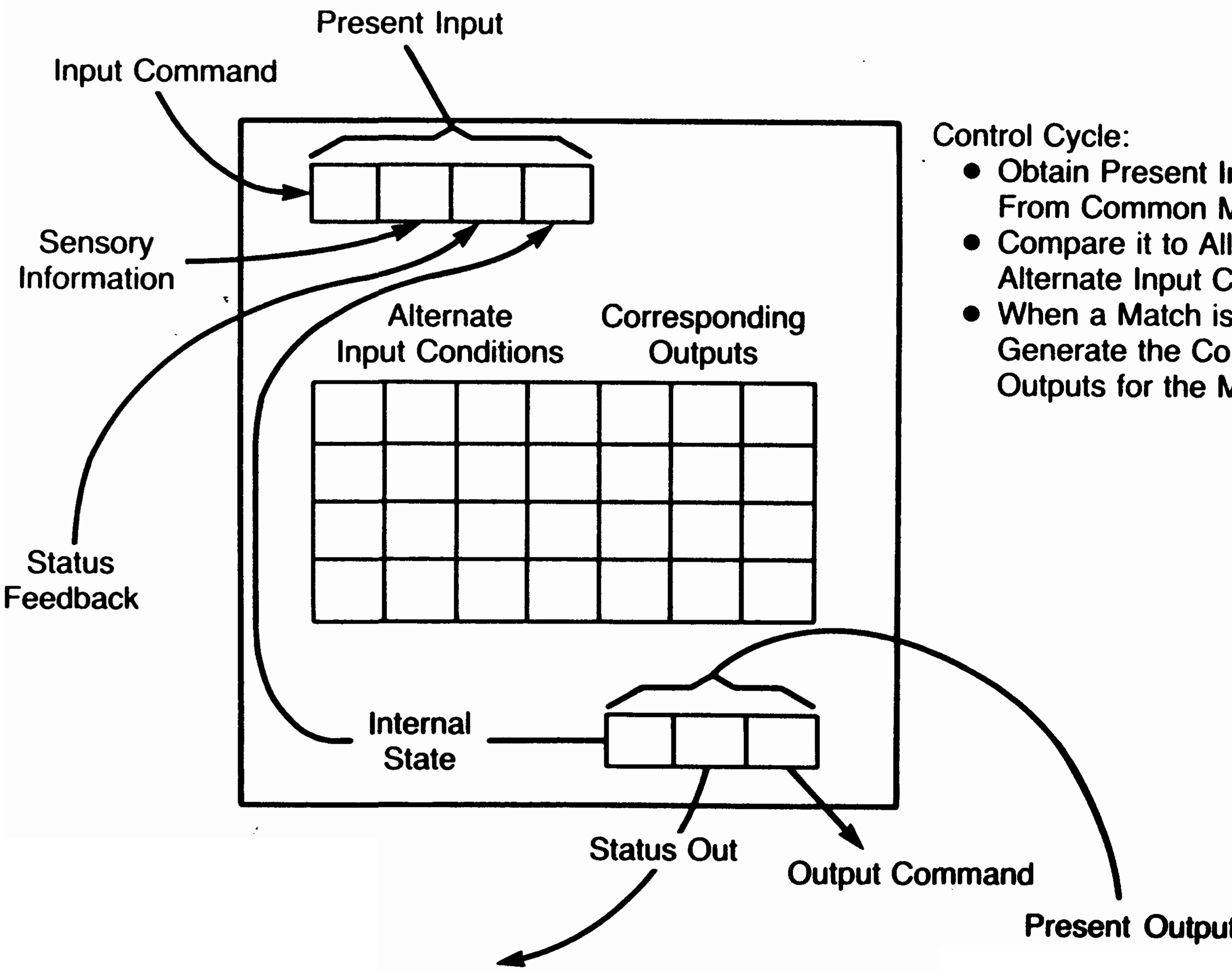
The many different individual operations and the sequences of operations that might be needed to make different parts are specified by information in the state table, not by code in the Workstation Controller software.

Figure 5 is a block diagram of the Workstation Controller showing how the three control decision levels fit into the WSC architecture. The software architecture of each of these levels is identical and will be discussed next. The other major functional blocks shown support the interfacing of the WSC with other control systems. In the case of the HWSC, these are the cell control system, the robot control system, the machine tool control system, the fixturing control system, and the materials transfer interface. These functional blocks are discussed later (Section 4.3).

4.2 Level Architecture

Each control level of the Workstation Controller has the same architecture. This is shown in Figure 6. The Input Interface contains the current information on the basis of which each control level makes its decisions. The Search Handler obtains a copy of the Input Interface information and searches the inputs side of the state table for a match. When a match is found it copies the corresponding output information to the Output Interface. The size and field definitions of the Input Interface and Output Interface are specified during creation of the particular controller. This is described in Section 7.

State Table Implementation



Control Cycle:

- Obtain Present Input From Common Memory.
- Compare it to All Alternate Input Conditions.
- When a Match is Found, Generate the Corresponding Outputs for the Matching Row.

Figure 4

Architecture of a Workstation Control Decision Level

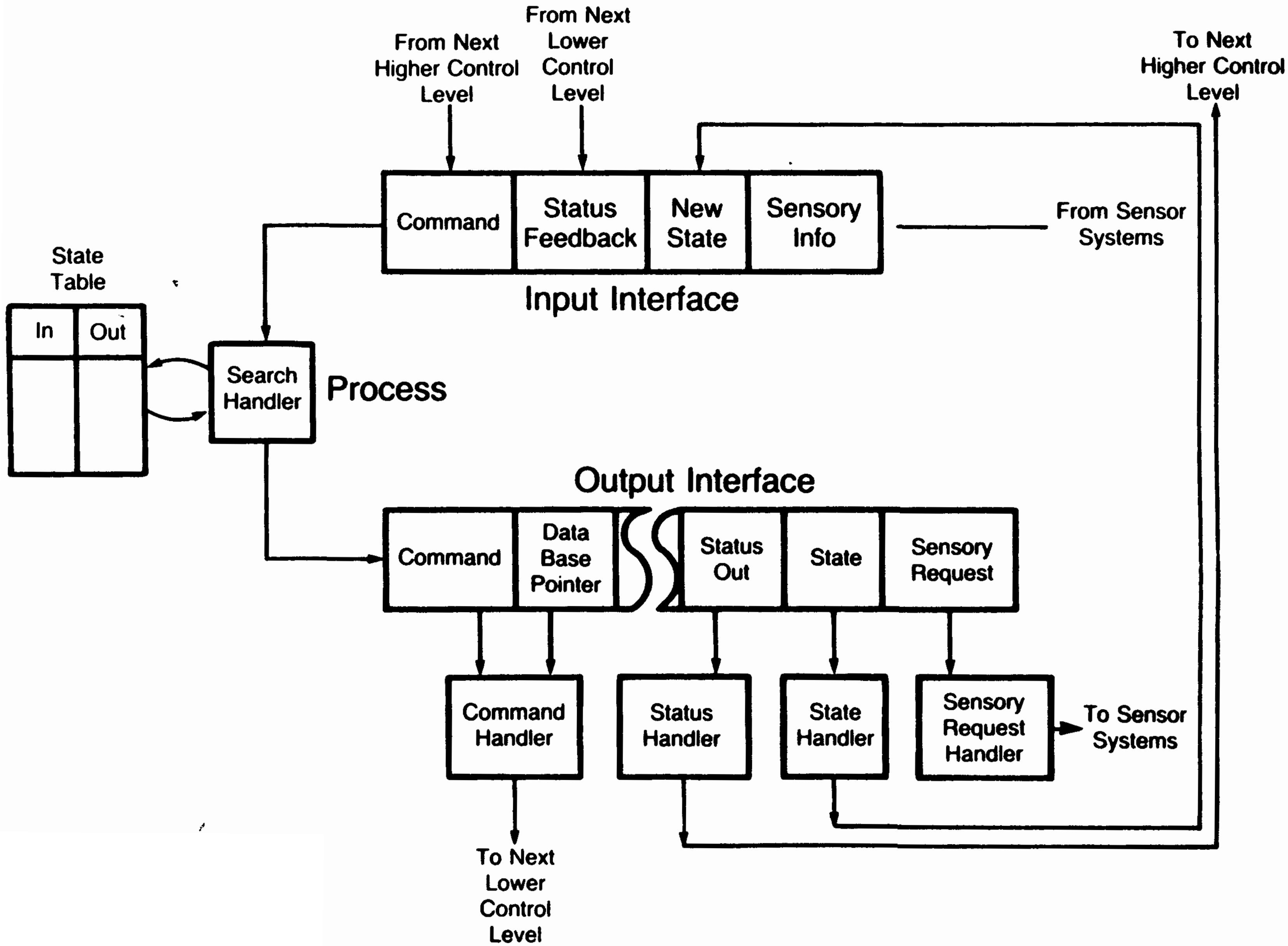


Figure 6

Work Station Control Interface to Another System

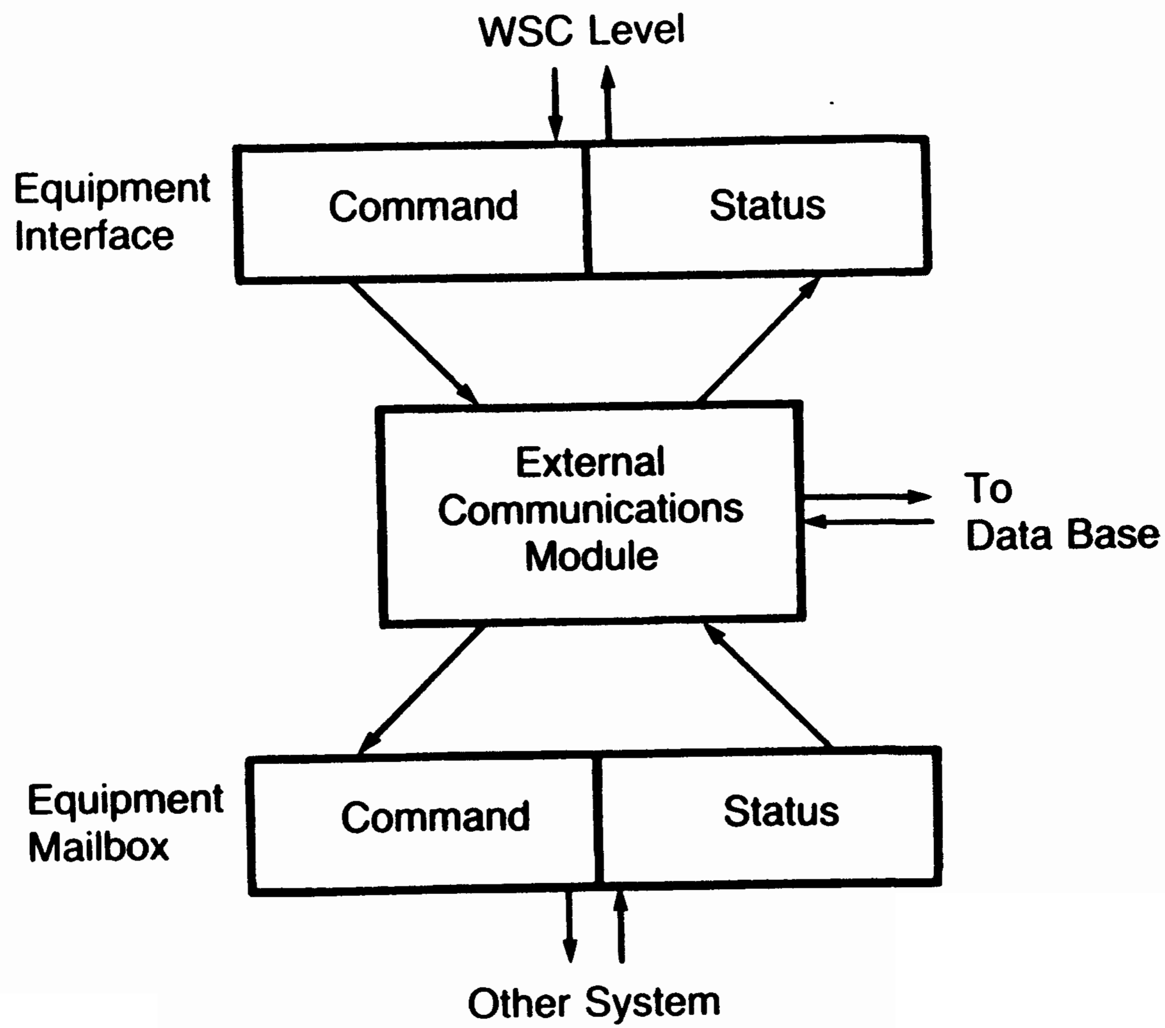


Figure 7

Horizontal Machining Center Workstation Controller

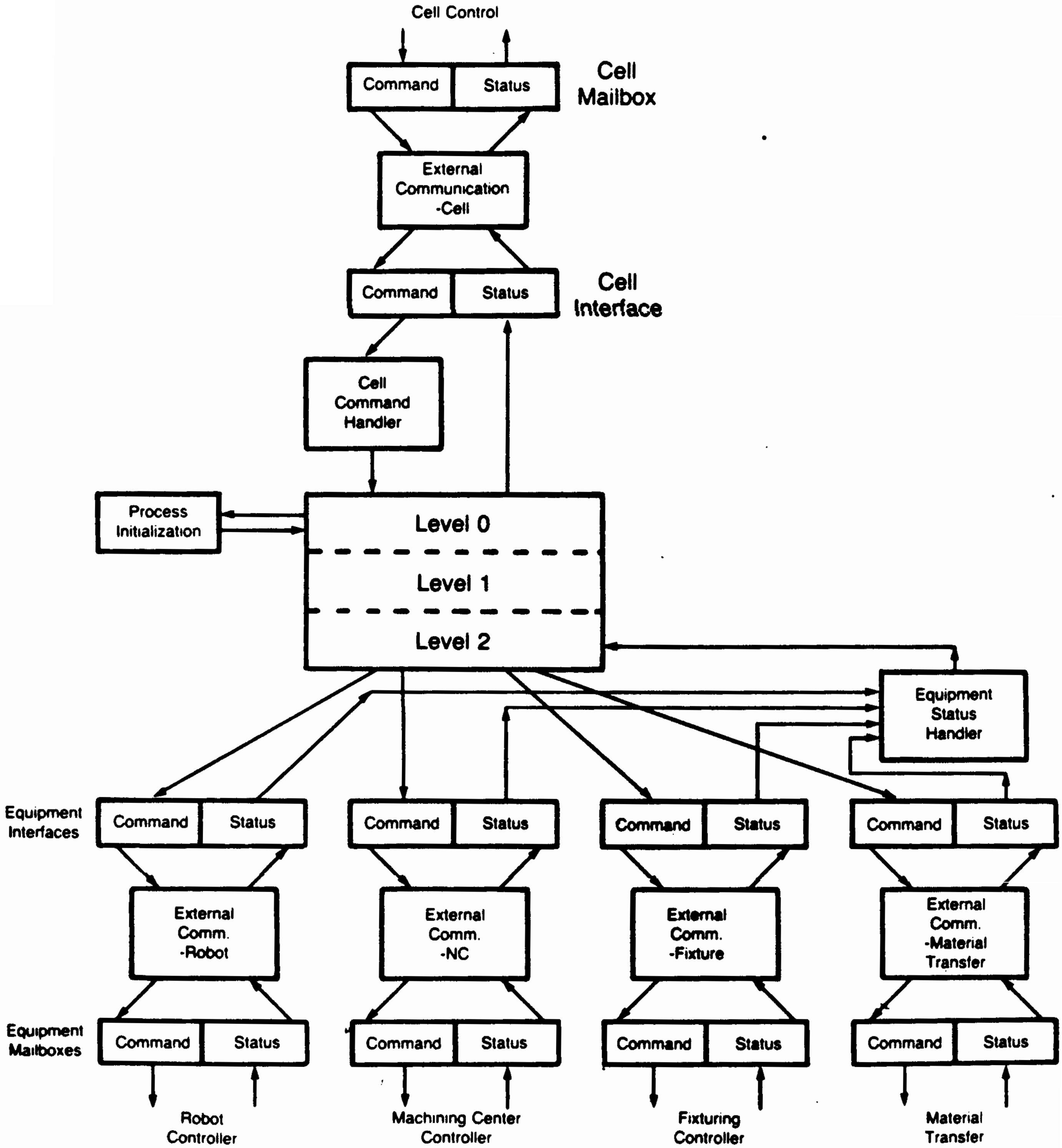


Figure 8

Names of Internal Interfaces and Storage Areas for HWSC

	Inputs	Outputs	Controldata
Level 0	L0-Inputs	L0-Outputs	0Table
Level 1	L1-Inputs	L1-Outputs	1Table
Level 2	L2-Inputs	L2-Outputs	2Table

Figure 9

* LOWER -- the name of the input interface of the next lower level, or the name of the common memory communications interface below

Level Definitions are used to create similar data structures for each control level of the WSC. Note that the Level Definitions describe the relationship between levels by specifying, for each level, the input interfaces of the next higher and next lower levels.

7.4 Offset Definitions

Offset Definitions are used to specify the meaning of the various items in the data structure created by the Level Definition. Figure 11 shows the Offset Definitions for Level 1 of the HWSC. The defining word "(LEVEL)" is used to create a pointer to an item in the level data structure using a byte offset for that item. For example, for Level 1, LOWER is the first item in the level data structure (0 bytes away from the beginning of the level data structure), HIGHER is the second item (2 bytes away from the beginning of the level data structure), CONTROLDATA is the third item, OUTPUTS is the fourth, and INPUTS is the fifth. (In the HWSC, two bytes are used to store each item in the level data structure.)

Offsets can be defined once and then used for all control levels, or the system programmer can define a separate set of offsets for each control level. If different sets of Offset Definitions are used, a different number of items can be stored in each level data structure. This feature is especially useful at the lowest control level if the WSC is used to communicate with several other controllers. For example, Level 2 of the HWSC sends commands to several equipment controllers rather than to another WSC control level. The interfaces through which equipment commands are sent are all considered "LOWER" in the Level Definition for Level 2 (Fig. 10B). The names of these interfaces --ROBOT-INTERFACE, MACHINE-INTERFACE, and FIXTURE-INTERFACE -- are all included in the data structure for Level 2. The Offset Definitions for Level 2 are used to locate the names of all three LOWER interfaces in the level data structure.

By specifying new Level Definitions and Offset Definitions, the system programmer can change the number of other controllers with which the WSC can communicate, redefine the relationship between the control levels, or even add or delete control levels from the WSC.

7.5 Field Definitions

The Input and Output Interfaces of a control level are made up of a number of fields, each of which contains a different kind of input or output data. Fields are defined for each interface using a set of Field Definitions. Each Field Definition creates an index which gives the relative position of a field within an interface. Since a field is defined as an index rather than a specific location in common memory, a single set of Field Definitions can be used to describe several different interfaces.

Figure 12 shows the Field Definitions for the Input and Output Interfaces of Level 1 of the HWSC. The defining word "NUMERIC" is used to create a two-byte field for the given field name. Other defining words are used to create fields of different sizes.

7.6 Data Access

To access any field within the Input or Output Interface of a level, the system programmer uses an English-like phrase which selects a level, an item within that level, and a field within that item. For example, to access the status feedback field

Field Definitions for the Input and Output Interfaces of Level 1 of the HWSC

Inputs	Outputs
Numeric Command	Numeric Command
Numeric Feedback	Numeric DB-Pointer
Numeric New-State	Numeric Level-Status
Numeric Sensory-Info	Numeric State
	Numeric Sensory- Request

Figure 12

examined by the diagnostic modules during every control cycle, providing extensive diagnostic and status-reporting capabilities. Since the diagnostic modules have access to the system dictionary, they use the same English-like phrases used in the Handler modules.

The Level Display module (Fig. 13) displays the contents of the Input and Output interfaces of any control level. (The Level 0 display also includes the contents of the Cell Interface; the Level 2 display includes the contents of the Equipment Interfaces.) The input and output fields shown in the display correspond to input and output columns in the state-table of the level that is displayed. The user can select one level at a time for display. The display is updated in real-time, allowing the user to see the incoming command and status feedback information and the outputs generated as a result of the state-table search. The user can display another level by pressing a key at the user terminal.

The Mailbox Display module displays the contents of any mailbox used by the WSC. The Cell-Status Display is used to display the status messages that are sent to Cell Control by Level 0 during every control cycle. The display is updated in real-time.

The WSC Monitor was designed to emulate the operation of any or all of the Equipment Controllers or the Data Base System during real-time execution of the WSC. The Monitor enables the user to enter status information that would normally be sent to the WSC by the Equipment Controllers or the Data Base System, thus allowing full operation of the WSC in the absence of any of the interfacing control systems.

The Monitor detects all new commands that are sent by the WSC to the control systems being emulated and displays appropriate status-request screens so the user can type in status information. Example display screens are shown in Figure 14. Status information entered through the Monitor is deposited directly into the appropriate communications mailbox. Thus, the WSC responds exactly as it would have if the emulated controller had sent the status information. The user can enter status information for an emulated control system even if a new command has not been sent by using the Mailbox Menu shown in Figure 15.

The Monitor allows efficient verification of control data since status information indicating that an equipment component has completed a command can generally be furnished much more quickly by the user than by the equipment component actually performing the command.

Examples of Monitor Display Screens

*****Robot Needs Attention*****

Robot Command MBX Contains))) Transfer
Hous
Buffer PED-V
PED 2

Build Robot Status Mailbox))) | |***| | | | |
left-justify all fields

Robot-Stat M# B1 B2 B3 B4

*****Data Base Needs Attention*****

DB Command MBX Contains))) Select Resource
Where PTGEOM EQ
Hous

Build DB Status Mailbox))) |***| | | |
left-justify all fields

M# Error-F Code E-Command

Figure 14

·IBLIOGRAPHY

. Barbera, A. J., Fitzgerald, M.L., Albus, J.S., Concepts for Real-Time Sensory-Interactive Control System Architecture, Proceedings of the 14th Southeastern Symposium On Systems Theories April 1982.

. Albus, J.S., Barbera, A.J., Fitzgerald, M.L., Kent, E., McLean, C., McCain, H., Bloom, H., Haynes, L., Furlani, C., Barkmeyer, E., Mitchell, M., Scott, H., Bloomquist, D., Kilmer, R., A Control System for An Automated manufacturing Research Facility, Robots 8 Exposition and Conference, Detroit, Michigan, June 4-7, 1984.

3. Barbera, A.J., Fitzgerald, M.L., Albus, J.S., Haynes, L.S., RCS: The NBS Real-Time Control System, Robots 8 Exposition and Conference, Detroit, Michigan, June 4-7, 1984.

. Kent, W.K., Albus, J.S., Servoed World Models as Interfaces Between Robot Control Systems and Sensory Data, Robotics, V2, No. 1 (in press)

. Meyers, D., Servo Control for An Instrumented Pneumatic Gripper.

. Kilmer, R.D., McCain, H.G., Juberts, M., Legawik, S.A., Watchdog Safety Computer Design and Implementation, Robots 8 Exposition and Conference, Detroit, Michigan, June 4-7, 1984.

. Vranish, J., Quick Change System for Robots, Robots 8 Exposition and Conference, Detroit, Michigan, June 4-7, 1984.

. Rippey, W.G., Control Interfaces of An NC Machining Center for Automated Operation, presented at Machining Centers: Current and Future Technologies, Indianapolis, Indiana, April 17-18, 1984.

. Barkmeyer, E. J., M. J. Mitchell, Data Distribution in the NBS Automated Manufacturing Research Facility. Proceedings of IPAD II Conference, Denver, Colorado, April 1984

0. Rippey, W.G., Scott, H.A., Real-Time Control of a Machining Work Station, presented at Numerical Control Society's 20th Annual Meeting and Technical Conference, Cincinnati, Ohio, April 10-13, 1983.