NBSIR 88-3802

ARCHITECTURE AND PRINCIPLES OF THE INSPECTION WORKSTATION

June 8, 1988

By:
Howard T. Moncarz

U.S. DEPARTMENT OF COMMERCE National Bureau of Standards Gaithersburg, Maryland

ARCHITECTURE AND PRINCIPLES OF THE INSPECTION WORKSTATION

Howard T. Moncarz

June 9, 1988

TABLE OF CONTENTS

LIST OF FIGURES

ARCHITECTURE AND PRINCIPLES OF THE INSPECTION WORKSTATION

I. INTRODUCTION

1. WHAT THIS DOCUMENT IS ABOUT

The main purpose of this document is to present a top level description of the Inspection Workstation (IWS), in particular its control architecture. In addition, the design goals established at the outset of this project as well as the guiding principles used to accomplish these goals are discussed.

2. AUDIENCE

This document will be useful to anyone who would like a general overview of the IWS design and implementation without getting into the implementation specifics.

For those who need a more detailed description of the implementation, this document will serve as a useful introduction to other documents in the IWS series. Those documents are listed in Appendix A.

3. OVERVIEW

The next chapter (Chapter II) gives a brief description of the IWS as it is currently implemented. It serves as the basis for explaining the IWS design and implementation in succeeding chapters. Chapter III presents the design guidelines. It begins by describing the goals of the design, and then explains the methods for achieving them. Chapter IV breaks down the design into its main components and then describes them. The next chapter, Chapter V, describes the ECS program which embodies the main principles of the AMRF philosophy and is intrinsic to the IWS software architecture. This program takes charge of loading and running each of the four controller programs that, taken together, encompass the IWS implementation. These controller programs are described in Chapter VI. Finally, Chapter VII presents a scenario of the IWS implementation in operation. This example should serve as a reinforcement of the concepts and description of the IWS implementation as herein presented.

The appendix contains a list of the documents in the IWS series (Appendix A), a list of other references (Appendix B), a glossary of terms used in this document (Appendix C), and a reader comment form to solicit feedback.

II. BRIEF DESCRIPTION OF THE IWS

1. WHAT IS THE INSPECTION WORKSTATION (IWS)?

The IWS inspects parts delivered to it by the AMRF Material Handling System. With the equipment currently installed, the IWS can perform two types of part inspection: inspecting dimensional tolerances and measuring surface finishes.

Presently the IWS merely determines whether a part is within tolerance or not. That data is displayed locally but is not stored. Eventually all data that is collected at the IWS, raw and processed, will be saved in the AMRF IMDAS (Integrated Manufacturing Data System), the distributed data system which provides common interfaces to the AMRF's user programs and underlying databases ([B.1] and [B.7]).

2. WORKSTATION EQUIPMENT

The Inspection Workstation is located in the temperature controlled room at the southwest corner of the AMRF shop floor. Figure 1 shows the location of the IWS within the AMRF. The physical layout of the IWS is shown in the schematic diagram in Figure 2.

A cart carrying a tray of parts to be inspected comes into the IWS along the track shown in the figure. The tray is automatically delivered to the tray transfer station that is located under the robot. (The robot is mounted on a gantry above the IWS.)

The actual inspection equipment at the IWS includes the coordinate measuring machine (CMM) and the surface roughness instrument (SRI). The CMM is used to inspect the dimensional tolerances of a part. The part is placed on the table of the CMM, and a force sensitive probe is directed by computer to touch the part at selected points to measure its surfaces. (Refer to the <u>Operations Manual for the Inspection Workstation</u> [A.8] for a more detailed description of the IWS equipment.)

The SRI is used to determine the surface roughness of a part. It performs this task by transmitting an infrared light beam onto the surface of a part and receiving the light scattered back by a row of twenty detectors. The scattering pattern detected is converted to a roughness average based on empirical determination.

The robot handles the parts while they are at the workstation. When a tray of parts arrives, the robot may transfer a part to the CMM or it may hold a part in front of the SRI. The automatic dial indicator (ADI) is used to help the robot to position the part at the correct distance from the SRI.

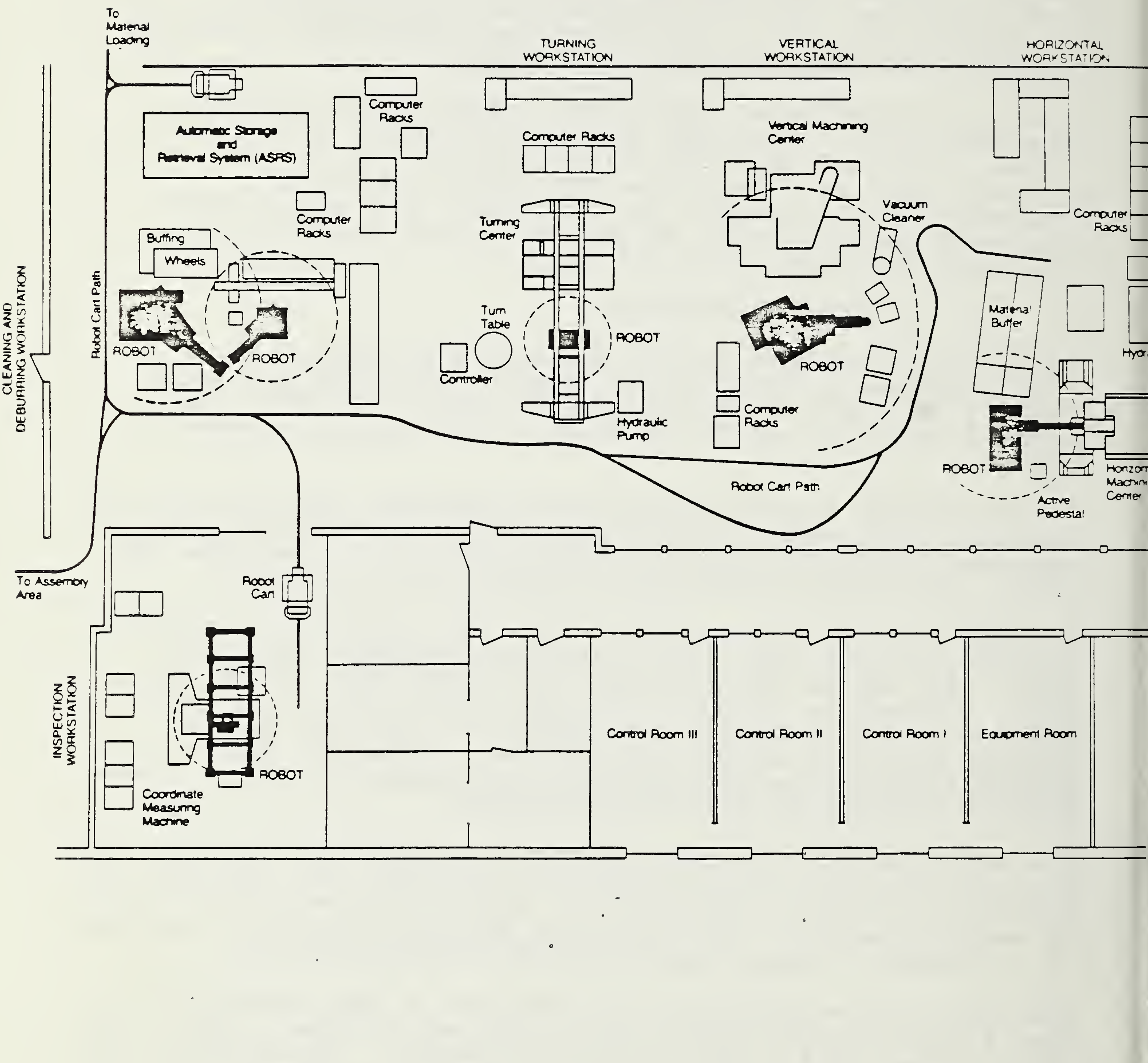

Figure 1. Floor Plan of the AMRF

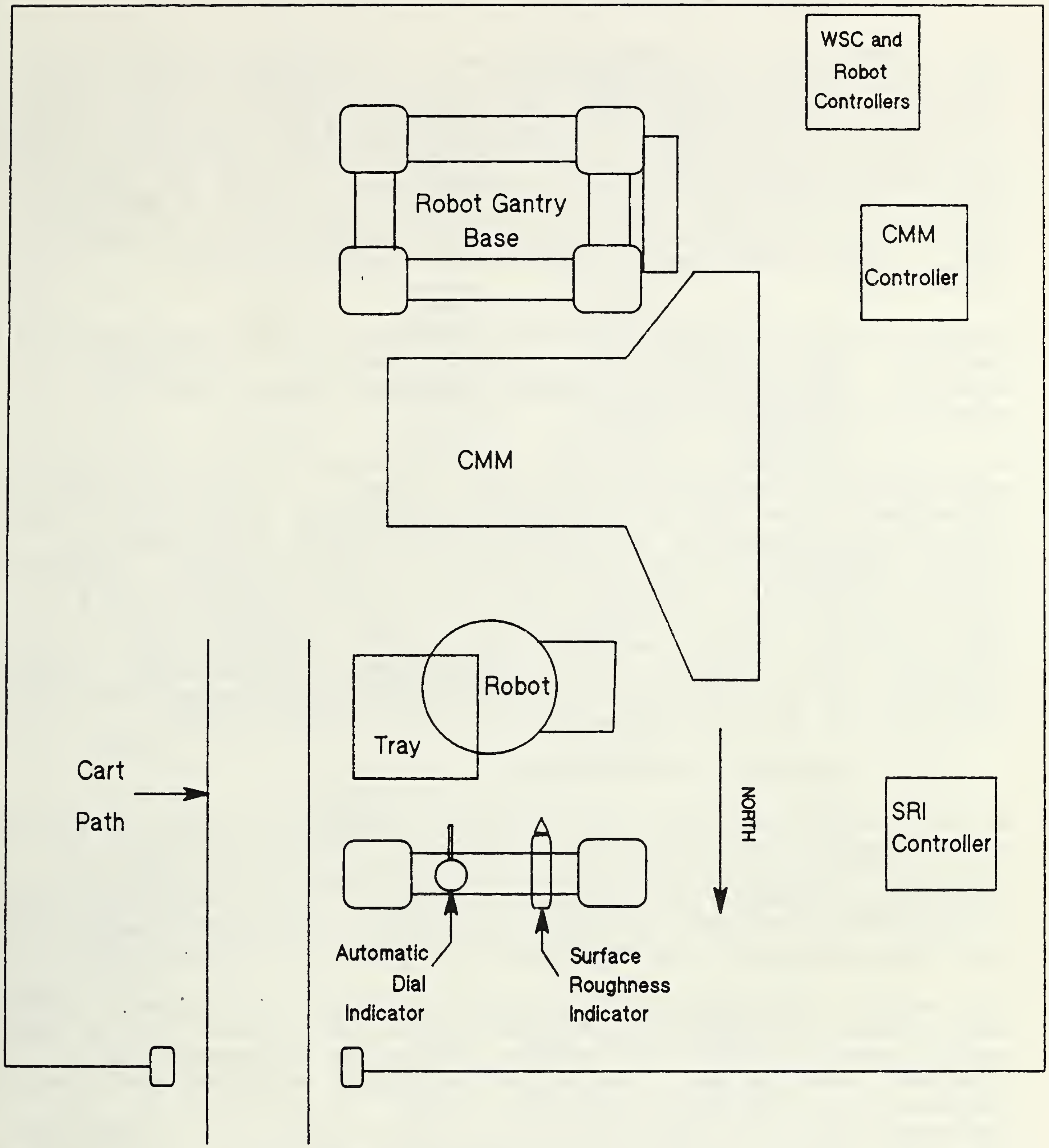

Figure 2. Floor Plan of the IWS

3. THE IWS CONTROLLERS

Each of the major equipment components of the IWS is controlled and monitored by a program running on a separate computer. The executing program along with the computer on which it is running is called a controller. The CMM, Robot, and SRI are controlled by the CMM Controller (CMMC), the Inspection Robot Controller (IRC), and the SRI Controller (SRIC), respectively. Since the ADI is used in concert with the SRI, it is also controlled by the SRI Controller. A fourth controller, the Workstation Controller (WSC), coordinates the activity of the three equipment controllers.

Refer to Figure 2 which shows the location of the controllers in the IWS. Note that the WSC and Robot Controller are shown together. These are two separate computer systems, but are mounted in the same cabinet.

All three equipment controllers are connected to the workstation controller via separate RS-232 serial ports. The WSC, in turn, is connected to the rest of the AMRF via a fourth serial port. The interfaces between the equipment and equipment controllers is varied, dependent on the requirements of the equipment. A future implementation will have all four controllers connected to a network interface unit (NIU), which in turn is connected to the AMRF Network. Figure 3a shows the current controller and equipment configuration, and 3b the future implementation. Note that the IWS has access to the AMRF Cell Controller and the IMDAS, although the physical connection to each is not direct.

4. LOGICAL CONFIGURATION

Aside from the physical configuration of the IWS, there is another way of looking at how the IWS components (controllers and equipment) are connected together. That approach concerns the routing of commands and statuses throughout the workstation. This is referred to as the logical configuration and is the manner in which the IWS should be considered to understand its design.

The logical configuration for the IWS is shown in Figure 4. The WSC receives commands from the AMRF Cell Controller. It, in turn, sends commands to the IRC and the CMMC. The SRIC gets its commands from the IRC. The actual pieces of equipment get their commands from the equipment controllers. In the current implementation, the WSC retrieves a part of the data it requires from the IMDAS. In the future, all four controllers will retrieve the data they require from the IMDAS, and this is also shown in the figure.

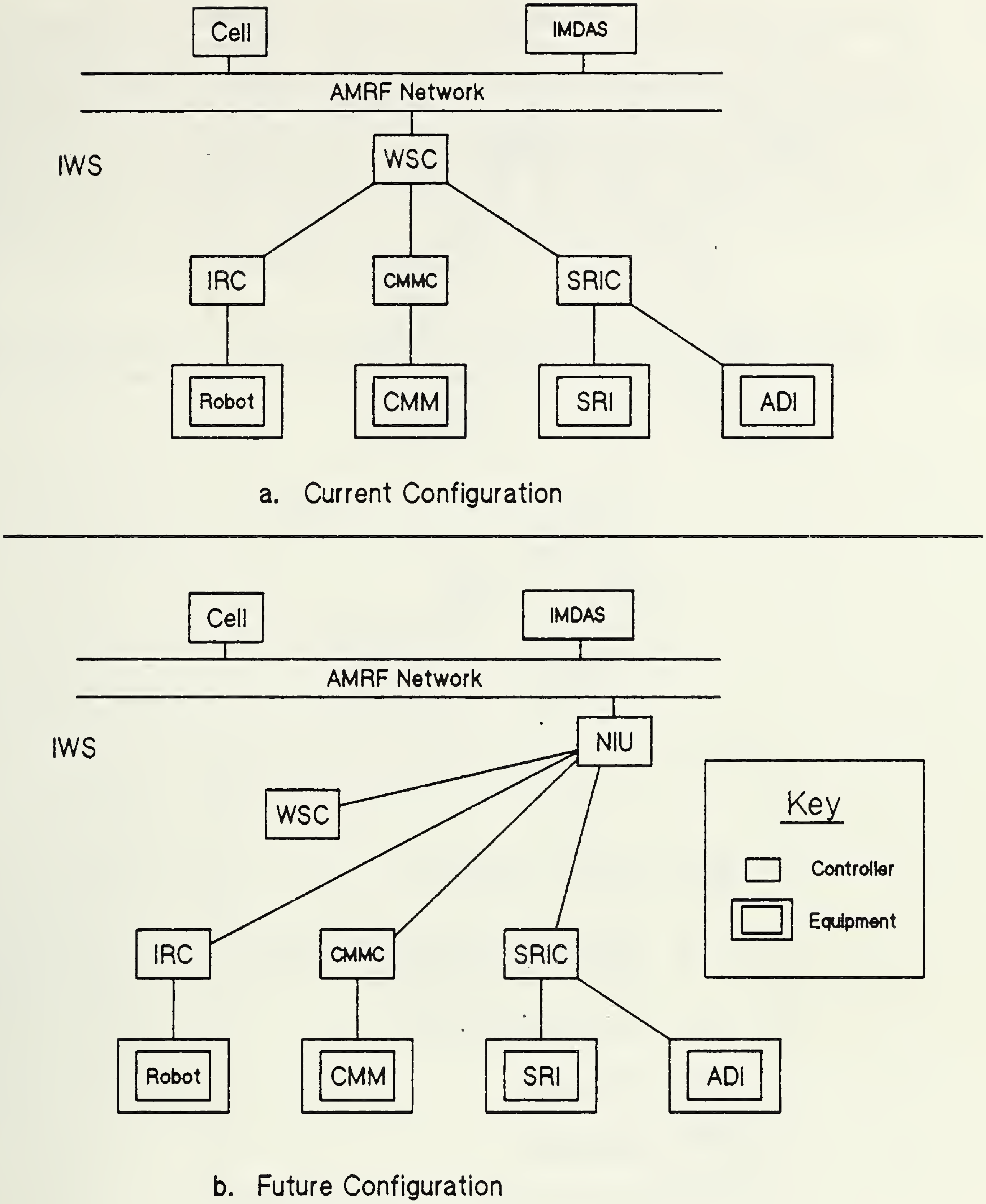

Figure 3. Controller and Equipment Configuration

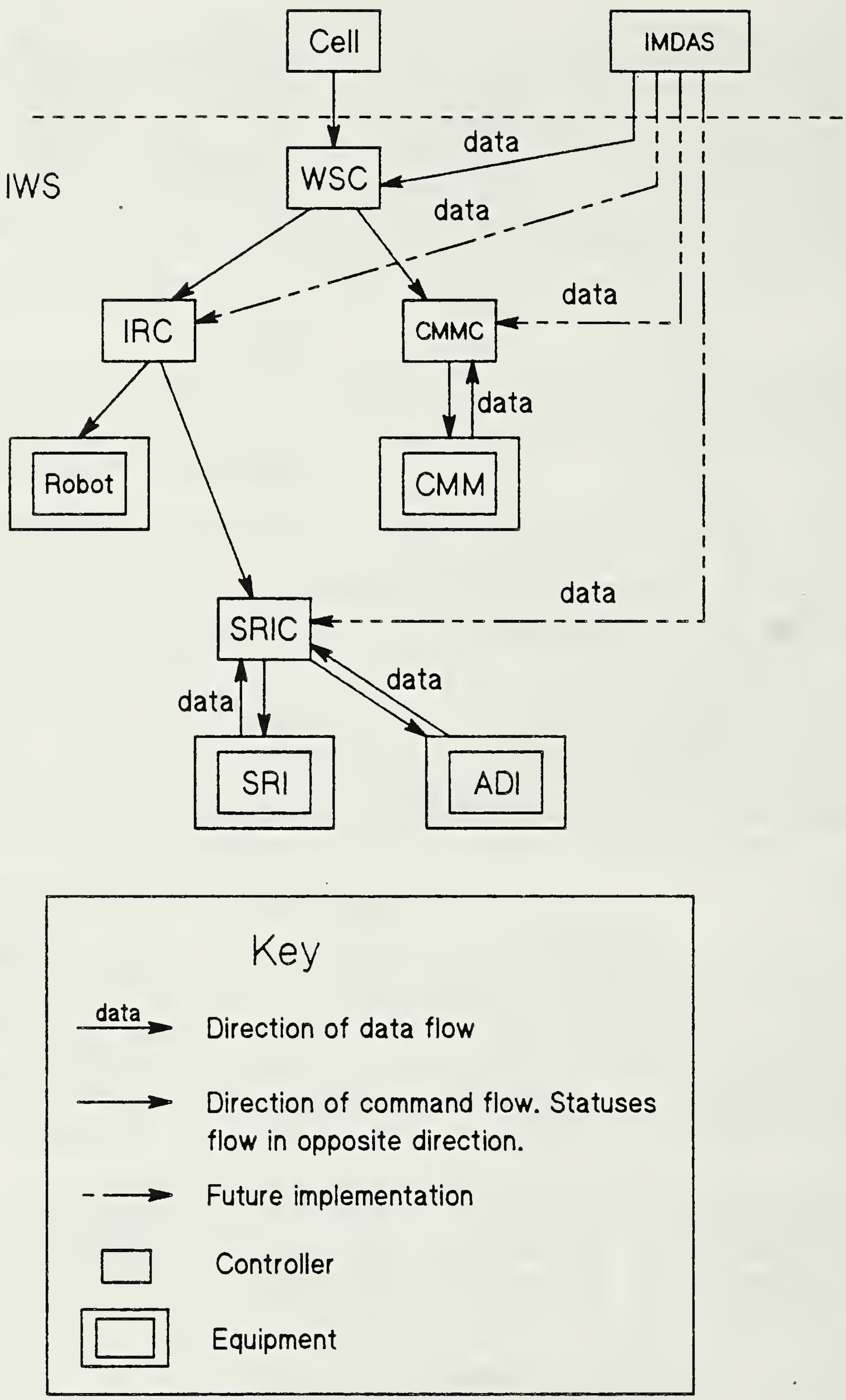

Figure 4. Logical Configuration of the IWS

5. THE EXECUTION CONTROL SYSTEM (ECS)

An identical computer program runs on all four controllers. This program is called the execution control system, or ECS. The controller programs referenced in section 3 are composed of software modules that are specific to a particular controller together with the ECS program. These modules (referred to as the controller modules) are loaded and executed by ECS and use the services and utilities provided by it. ECS takes care of the integration and details concerned with the AMRF, so that the controller modules can implement the main functions involved with the controller without regard to the AMRF.

ECS utilizes hierarchical task decomposition, data-driven control, state machines, common memory, and network communications. These are described further in the next three chapters.

III. DESIGN GUIDELINES

1. GOALS

The essential objectives for the IWS design included developing a general IWS architecture, integrating the IWS into the AMRF, and generalizing the implementation so that it would have usefulness beyond the IWS. Ultimately, the goal was to develop a generic method for implementing a workstation, so that any workstation in the AMRF could be built using this method.

1.1. General Architecture

The architecture of the IWS should be general enough, so that equipment made by other manufacturers could be easily substituted, the software could be transferred to another computer system, and the workstation itself could be easily reconfigured.

Ideally, the software architecture should be general enough to accommodate any AMRF workstation, so that by following designated procedures the generically specified workstation could be customized to become the Inspection Workstation.

1.2. AMRF Integration

The integration of the IWS into the AMRF needs to accommodate two interfaces--the AMRF Cell Controller interface and the IMDAS interface.

1.2.1. Cell Controller Interface

All AMRF workstations perform their main functions in the READY state. They are directed to this state by executing a sequence of transition commands issued by the Cell Controller, and returning the proper statuses to it. In like manner, they must shut down operations by honoring a similar shut down sequence of commands and statuses. This start up and shut down procedure is known as the UVA protocol and is described in detail in [B.2].

Once in the READY state, the Cell Controller issues work order commands to the workstation to direct it to perform its main functions. The command protocols (for both transition and work order commands) are fully specified and must be supported by all AMRF workstations. The formats for Cell Controller commands and statuses are fully documented in [B.5].

1.2.2. IMDAS Interface

All AMRF workstations must retrieve data from and store reports to the IMDAS. An IMDAS query language has been specified to allow data transfers in a generic manner. Each workstation can use this language and not concern itself with the details of how and where the data is stored in the AMRF. The details of this query language are provided in [B.1].

1.3. Standard Interfaces

In designing the IWS, all opportunities should be identified where generic interfaces or methods could be utilized. In interfacing the IWS to the AMRF, all AMRF interfaces that have been specified should be incorporated. Additionally, where opportunities for standardization avail themselves within the IWS itself, they should be noted and taken advantage of if possible. In particular, interfaces between controllers within the IWS should be standardized. Likewise, interfaces between controllers and equipment should be generalized if possible.

2. METHODS

To achieve the objectives set forth above, the IWS control structure should be carefully modularized and should be as data driven as possible.

2.1. Modularization and Task Decomposition

One of the key design principles utilized in the AMRF is hierarchical task decomposition. This concept is reflected in all levels of the IWS design, and will be described in the succeeding chapters.

The IWS, from its topmost level, is hierarchically organized. This is seen in Figure 4, which shows the logical configuration of the controllers and equipment in the IWS. The main function of the IWS is to inspect parts. Each IWS controller handles a separate aspect of this task, and these are hierarchically arranged as shown in this figure.

The software comprising each controller should be hierarchically decomposed as well. This makes it easier to substitute equipment from different manufacturers, to easily transfer the software to another computer, or to easily reconfigure the controller.

To design a controller in this fashion, the first thing that must be done is to define the main tasks that the controller must handle. These tasks are then decomposed into simpler tasks. The decomposition should be carefully designed, so that each task is clearly independent from the others and is in its proper position in the hierarchy.

2.2. Data-Driven Control

Each controller should be designed so that it is completely data driven (or at least as much as practical). The controller program is a fixed piece of code. The decisions it makes, the schedule of activities it directs, the actual parameters sent to the equipment are all determined by data. All data should be clearly identified and separated out of the control structure, so that the control structure is truly data independent.

To inspect a particular part, the data for that part should be retrieved by the controller directing the inspection. To coordinate the activity of the whole workstation, the WSC should retrieve the process planning data it requires as well.

IV. ARCHITECTURAL ELEMENTS OF THE IWS CONTROLLERS

1. STATE MACHINES

The basic building block of the IWS control software (as well as much of the AMRF control software) is the finite state machine (FSM). (Technically the IWS uses state machines, not finite state machines, but the abbreviation FSM will still be used.)

1.1. Description of a State Machine

All FSMs have a similar architecture. A diagram of the generic state machine is shown in Figure 5. The state machine can be viewed as a black box connected to the rest of the system by specified inputs and outputs. The inputs derive from three sources: outputs from other FSMs, data collected or processed at its own level (sensory data), and an FSM's localized view of the system (the world model). The black box processes these inputs and produces outputs, which are sent to other FSMs.

Internally, the FSM consists of three parts. A preprocessor converts the raw inputs to a form that is convenient for the next level of processing. That level consists of a number of rules, one of which will be executed. The one executed depends on the processed inputs as well as the current state of the FSM. Execution produces raw outputs, affects internal variables, and can switch the FSM to a new state. Finally, the postprocessor converts the raw outputs to a form that can be utilized by the other FSMs.

1.2. Combining State Machines to Form a Controller

A controller is created by stacking FSMs on top of one another. The top level FSM is in charge of executing the main functions that the controller performs. Those functions are decomposed into simpler tasks that the next level FSM performs, and so on. Each level of task decomposition is performed by an FSM.

Figure 6 shows an example of a three level controller. The main functions performed by the controller are done at Level 1. Each of these functions are decomposed into simpler ones that are executed at Level 2, and Level 2 functions are decomposed into simpler ones again at Level 3. Level 1 signals which functions in Level 2 should be performed by sending it commands, and Level 2 signals its progress back to Level 1 as status reports.

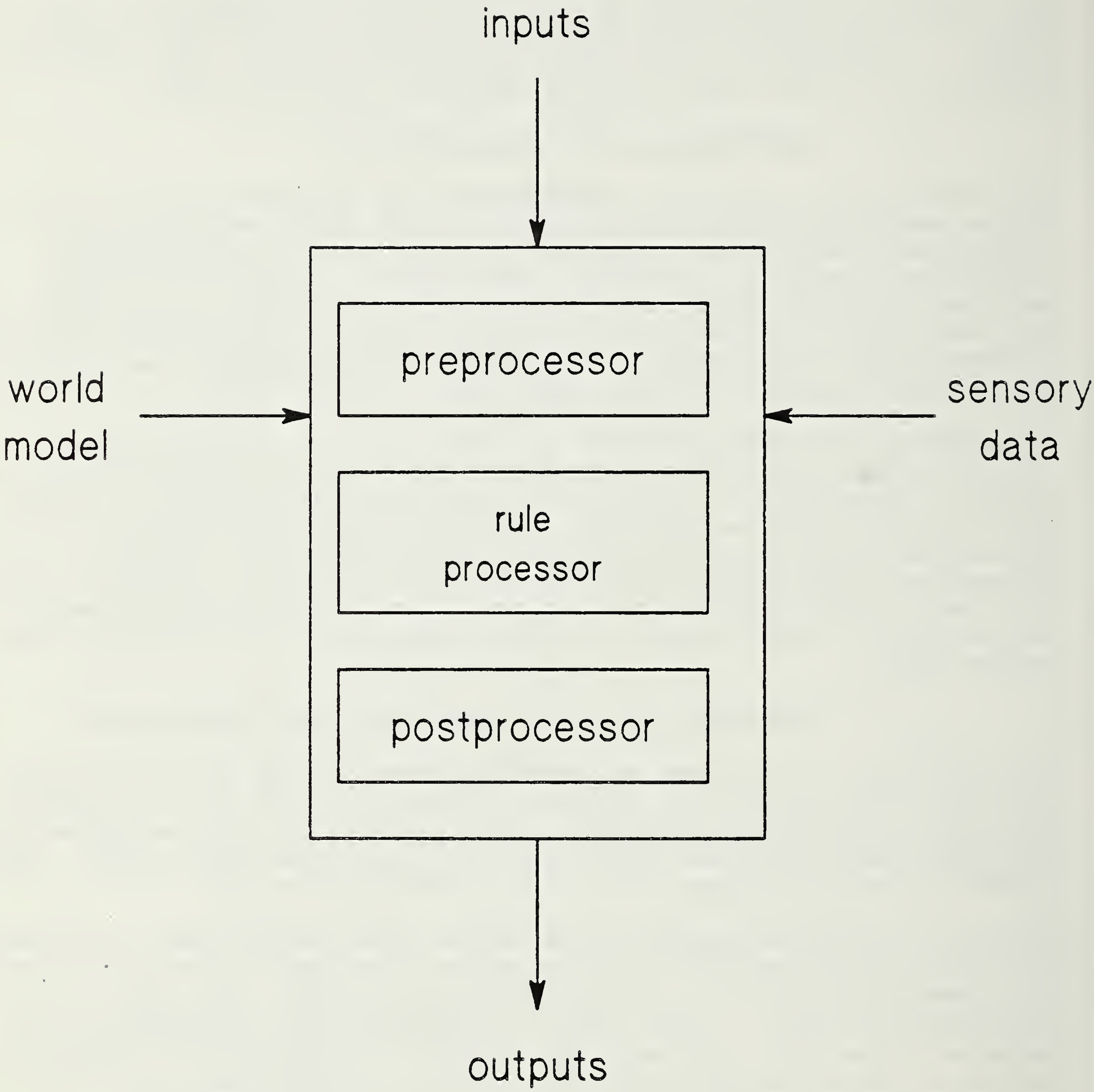

Figure 5. The Generic State Machine

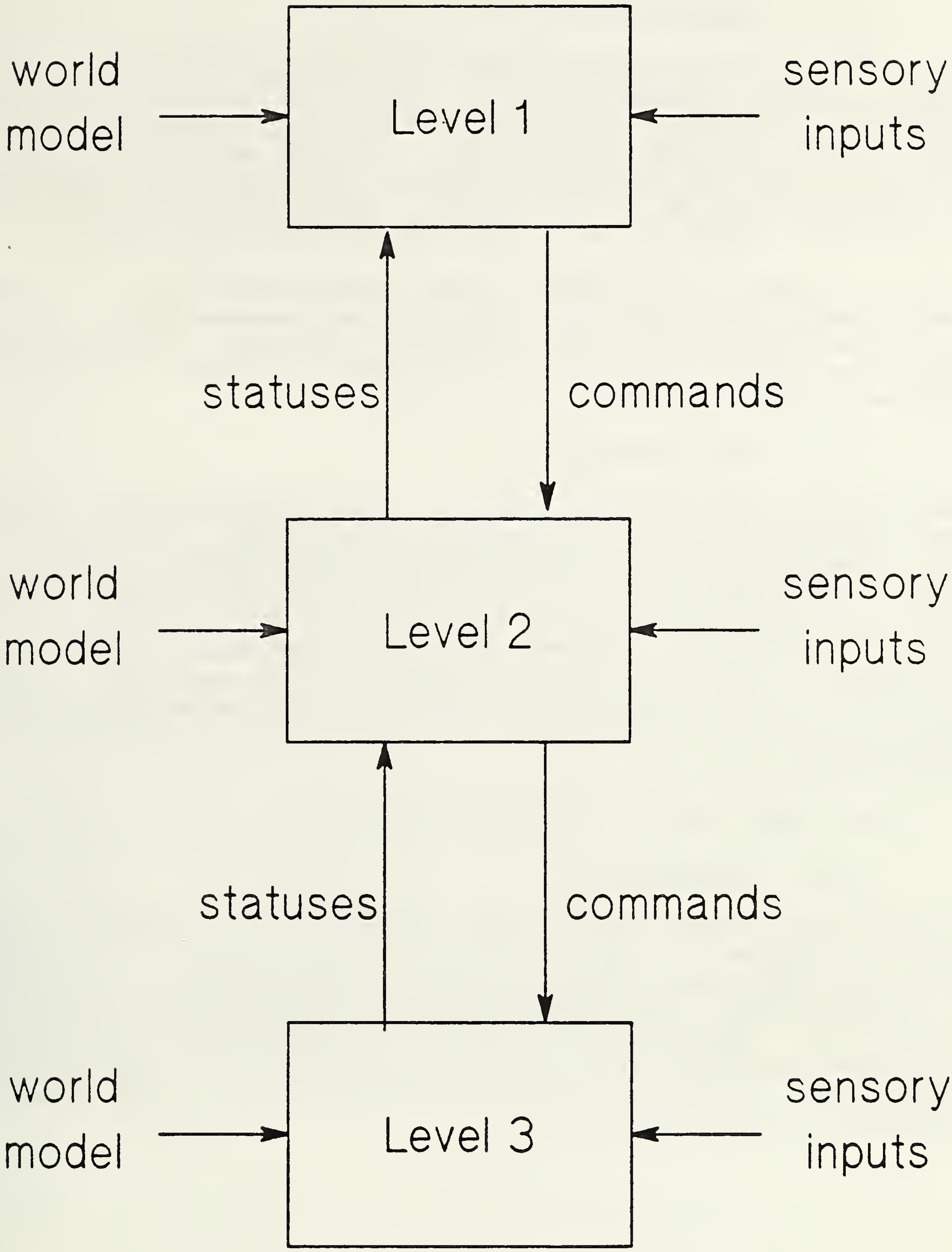

Figure 6. Three Level Controller

2. COMMUNICATIONS

2.1. Common Memory

All communications between FSMs are accomplished using common memory. This includes internal communications between FSMs residing within one controller and external communications between FSMs in two separate controllers. On initialization, each FSM declares which variables it needs to retrieve from common memory and which variables it will post. Any variable can only be posted by one FSM. However, any number of FSMs may retrieve it.

During execution the common memory system takes care of routing common memory variables to their proper destinations. The data transfers are totally transparent to the FSMs. Each FSM uses its local variables, and it is unimportant how those variables are updated. (See Figure 7.)

2.2. The Network

The network takes care of routing those common memory variables that must be transferred between controllers. The variables in this category are specified during system initialization. Again, none of the FSMs need to worry about how their local variables are updated. An input variable for an FSM may be produced by another FSM in the same controller, or it may come from another controller altogether. This is totally transparent to each FSM. Likewise, the common memory system does not need to know which variables are transmitted over the network. The network takes care of that. (See Figure 8.)

3. THE GENERIC CONTROLLER

3.1. Description

All IWS controllers have a task decomposition similar to the one shown in Figure 9. The top level FSM interfaces the controller to its supervisor controller (using the UVA protocol). The next FSM, the task manager, directs the main functions performed by the controller. Those main functions are decomposed into sub-tasks performed by successively lower level modules. Finally, the bottom level module interfaces the controller to the specific equipment it will supervise.

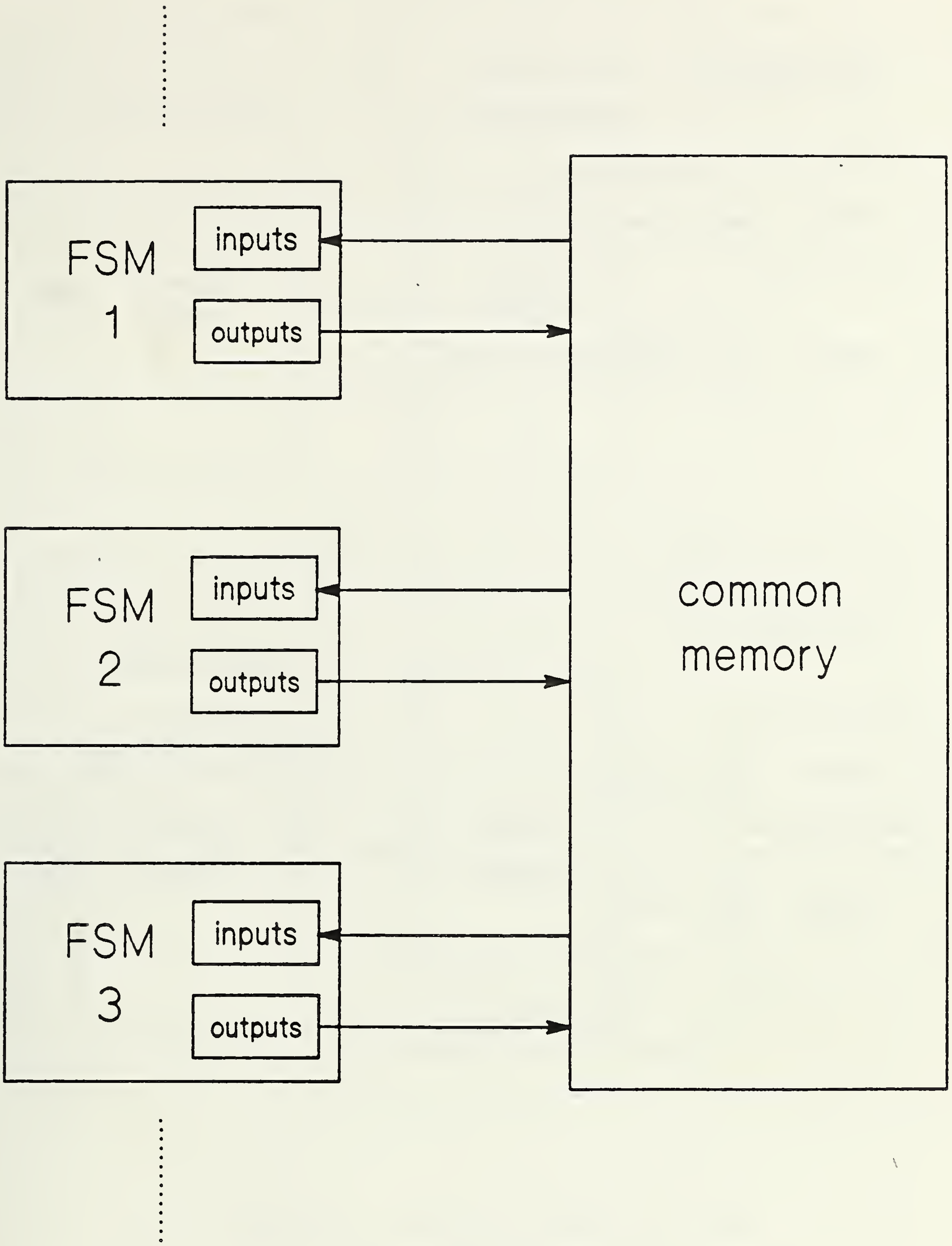

Figure 7. Data Transfer by Common Memory

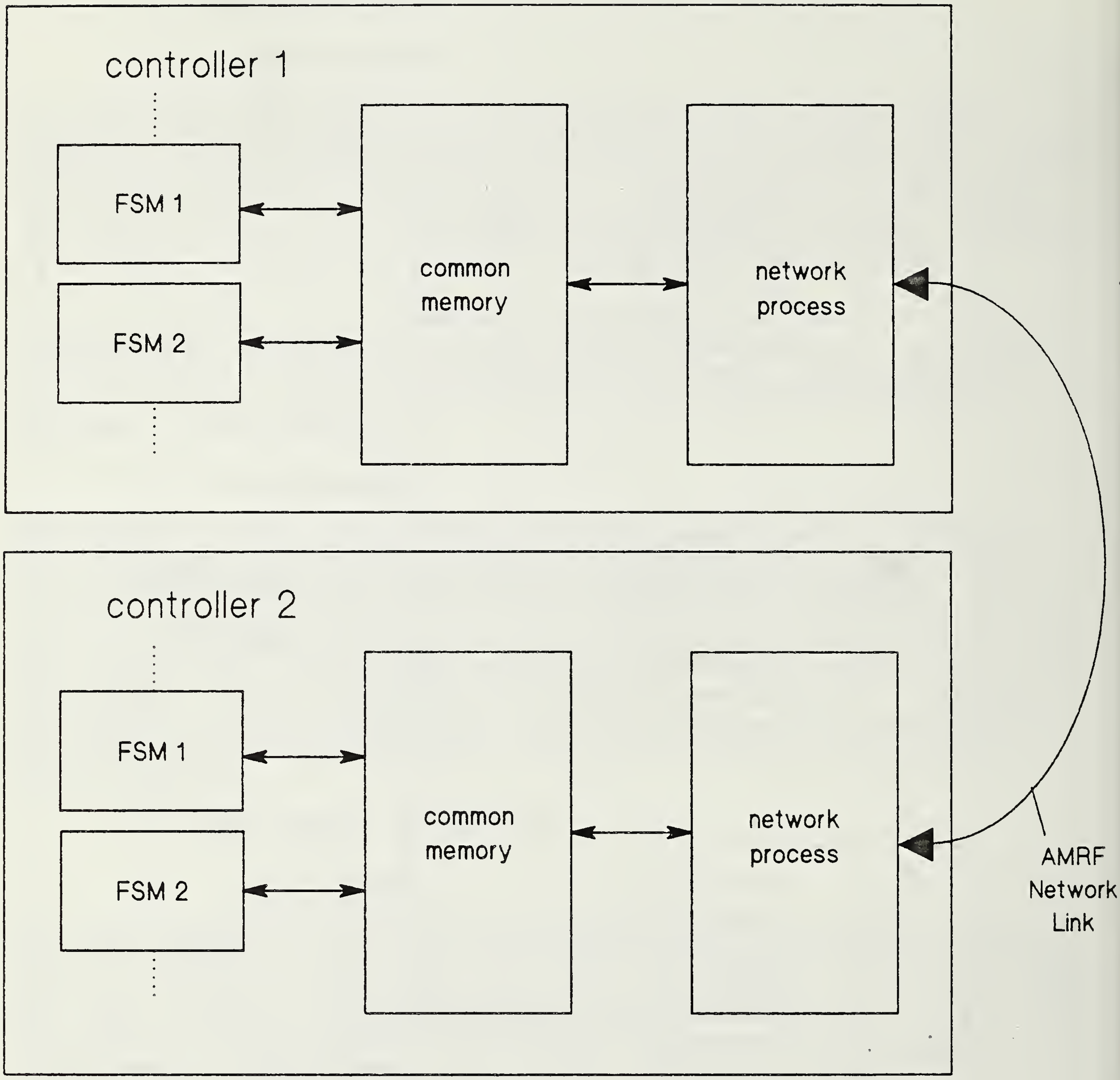

Figure 8. FSM Communications

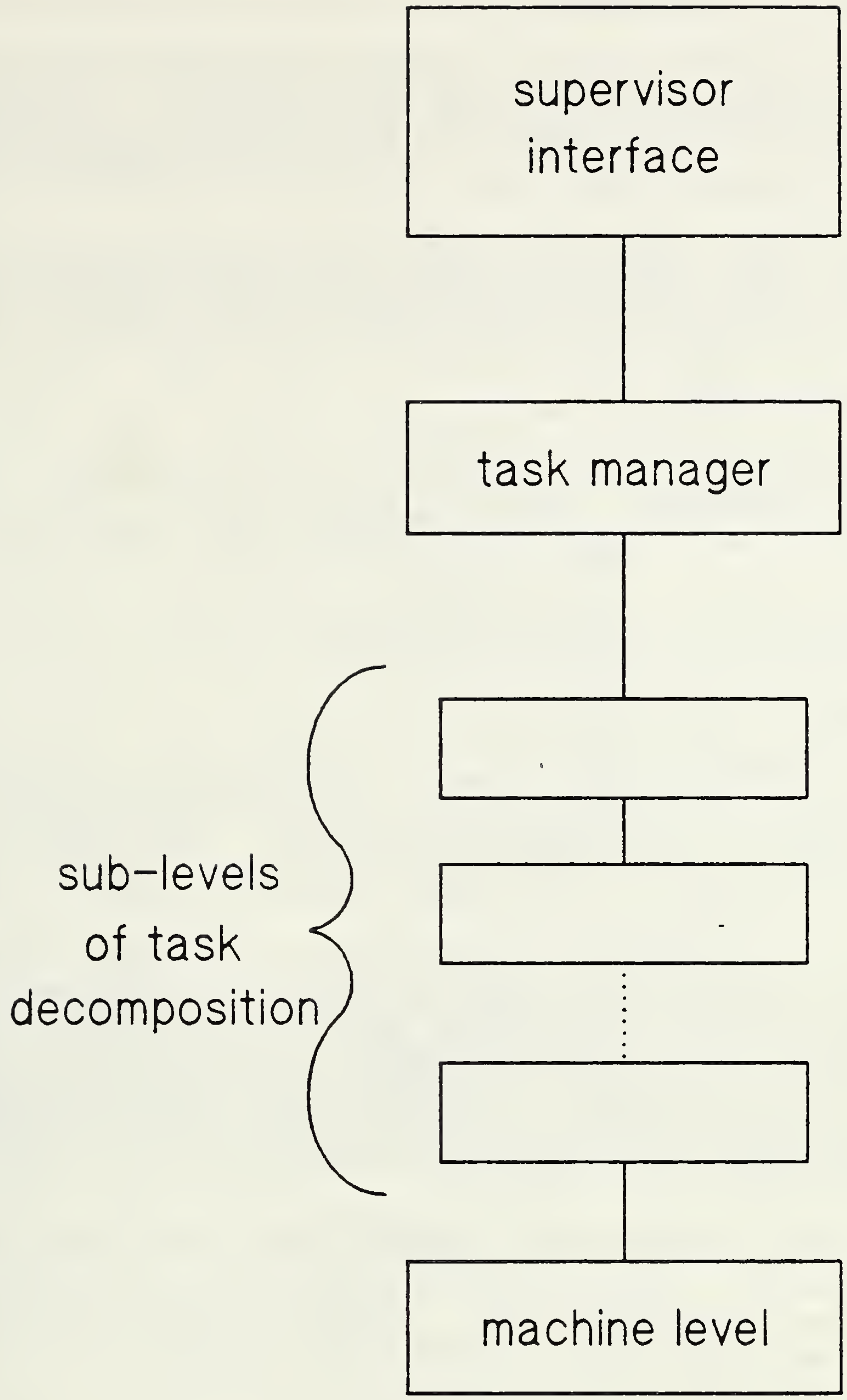

Figure 9. Control levels for the IWS Generic Controller

3.2. Device Interface

The controller's interface to the actual physical device should be clearly separated from the rest of the program. Preferably, the interface should be comprised of a set of procedures that are contained in the lowest level FSM, machine.

The machine level FSM should be the only module that is dependent on a specific device. All other modules direct their efforts to accomplishing the controller functions and are device independent.

3.3. Execution

During controller execution, the FSM's run in cycles. In each cycle, every FSM executes, one at a time. After the last FSM has run, common memory variables are transferred. Finally, the network transfers those common memory variables that are destined for another controller. This completes one cycle of the controller, and the next cycle can begin.

3.4. IMDAS Interface

All IWS controllers are data driven, and that data must originate in the IMDAS. The data server is a separate FSM that interfaces any controller FSM to the IMDAS. Currently, the only data retrieved from the IMDAS is the tray contents report, fetched by the workstation controller.

During controller execution, data is retrieved whenever an FSM needs it. Accessing data from the IMDAS continuously (or from any database system off-line from the IWS) would slow execution of the controller to a snail's pace. Instead, all the data required by a controller to perform one of its main functions is retrieved at one time by the task level module.

This large quantity of data is transferred to a local database, from which individual pieces can be retrieved as needed by any of the FSMs embodying the controller. This is done in a generic manner and will be explained in the next chapter.

V. THE EXECUTION CONTROL SYSTEM (ECS)

This chapter discusses the general ideas underlying the ECS program. For a detailed description of the implementation, consult the Implementation of the Execution Control System of the Inspection Workstation [A.2].

To understand the main idea behind the IWS software, it should be considered as three separate layers (Figure 10). The bottom layer is the computer's operating system, which directly accesses the computer's resources and makes them available to the next layer -- the ECS program. ECS runs on top of the operating system and implements the AMRF design principles--hierarchical task decomposition, data-driven control, state machines, common memory, and network communications. ECS serves as the interface between the computer's operating system and the controller program. The controller program runs in the top layer.

The relationship between the controller and ECS programs is analogous to that between an application program and the operating system. The application program utilizes the system's services, and consequently the computer's hardware. How that is done is the job of the operating system. In the same manner, the controller program uses the ECS services, and consequently is consistent with the AMRF principles. It is not necessary to program those principles directly into the controller program itself.

This approach frees the controller implementor to concentrate on the main functions performed by the controller, and not on details of the AMRF. Once an ECS is implemented on any computer, the task of writing a controller is greatly simplified. Ideally, the implementor would be free to select whatever language desired and whatever programming methods, as long as certain guidelines were followed (and the interface to ECS could be provided).

The services performed by ECS are explained in the following sections.

1. LOADING AND RUNNING STATE MACHINES

When ECS first begins its execution, it loads the FSMs that it will run. Some of the procedures used by these modules are imported from procedure modules that must also be loaded. All of these modules are executables, compiled from Pascal source code. Each module specifies which variables and procedures they can export to other modules and which modules they themselves need to import to access the latter module's variables and procedures.

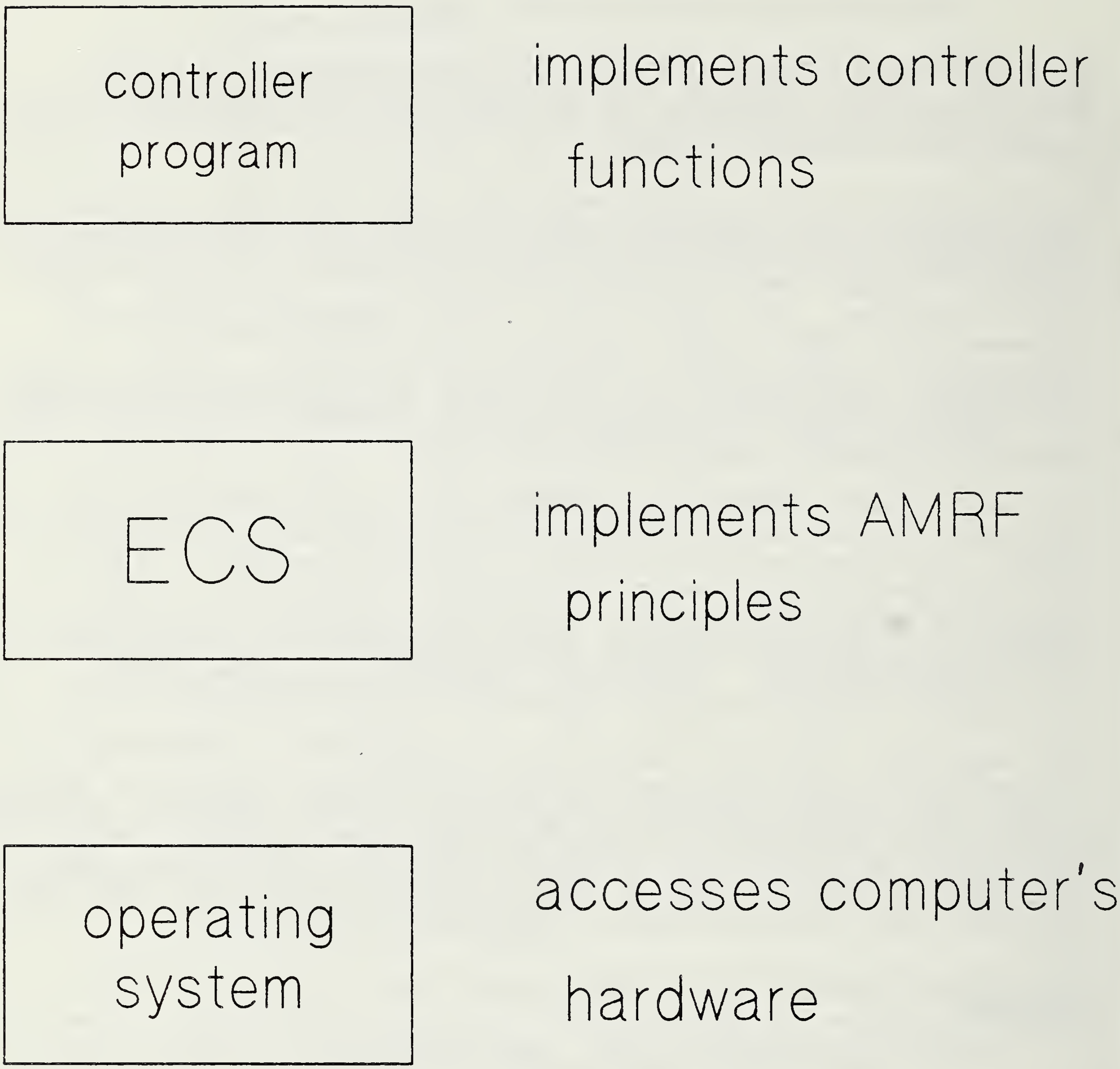

Figure 10. Three Layers of an IWS Controller

Once all modules have been loaded, the FSMs are executed one at a time, common memory variables and network variables are transferred, and the cycle repeats indefinitely (Figure 11). (Note that this implementation was chosen because the operating system on the HP computers used is single tasking.)

2. COMMUNICATING VIA COMMON MEMORY

Each FSM has a procedure, executed when it is loaded, that declares which of its variables will be written to common memory and which variables will be retrieved from it. As each variable is declared, space in common memory is allocated, pointers to its location in local memory and common memory are reserved, and the direction of its transfer is assigned.

The above information is stored in two lists of records--one for output variables (transferred from local to common memory) and one for input variables (with the data transfer reversed). Whenever common memory needs to be refreshed (once every FSM cycle), each list is scanned, and common memory variables are transferred accordingly. All output variables are transferred first, then all input variables.

3. THE NETWORK INTERFACE

The ECS program includes a module that takes care of transferring all network variables between controllers once every cycle. To common memory, a network variable looks just like any other variable.

When each controller begins its execution, a procedure from the network module is run that initializes the network and establishes data paths for all network variables. These variables and their intended destinations are defined by a script file that is read by the WSC after it begins its execution.

Thereafter, once every FSM cycle, a network procedure is run that transfers all network variables from the common memory of each controller to their proper destinations in the common memories of other controllers.

4. GENERAL ERROR RECOVERY

The ECS program includes a module that contains a few simple procedures that allow the controller implementor to specify the program's flow (or to exit the program) upon encountering an error. The detection of the error as well as the specifics of handling it are left up to the controller implementor. Currently,

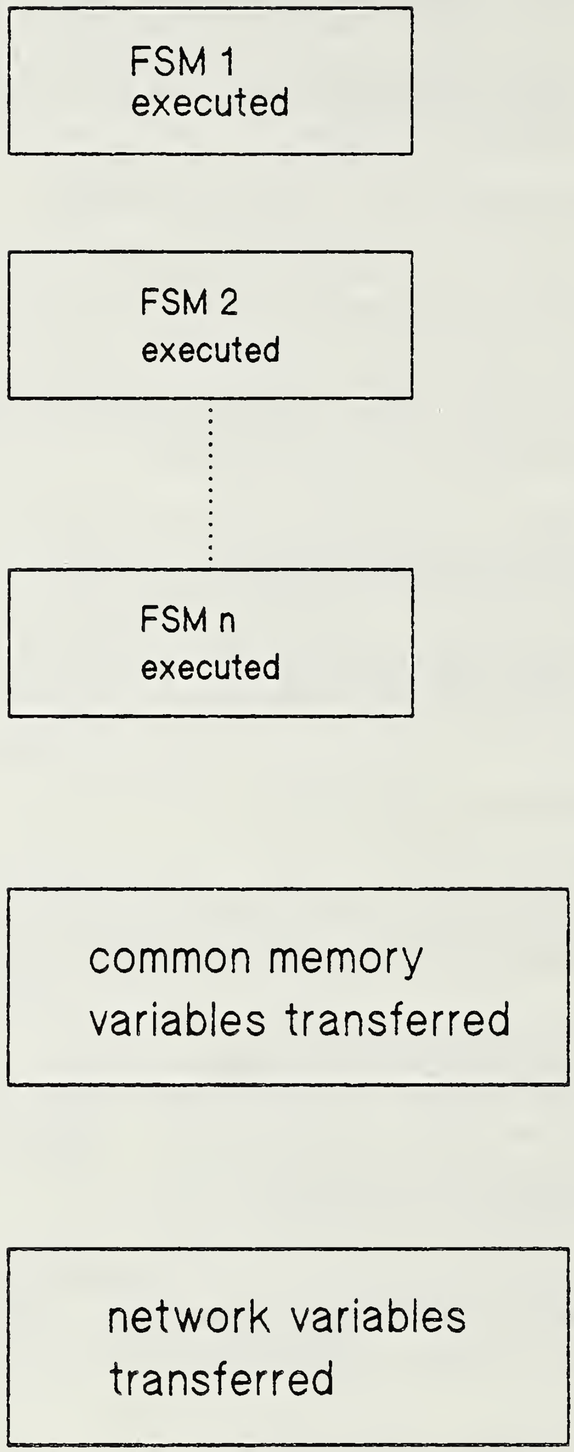

Figure 11. One Cycle of ECS

error recovery has not been built into the IWS. Anything that occurs that is unexpected by any of the controllers will crash it and also crash the entire workstation.

5. ACCESSING THE IMDAS

Each controller should be able to access the IMDAS. Currently the only transaction implemented by the IWS is the one in which the WSC retrieves the tray contents report. All other transactions require the block mode retrieval capability of the IMDAS, which has not yet been implemented.

A data server that accesses the IMDAS has been implemented as an FSM, and can be run on any of the IWS controllers. (It still must be integrated into the equipment controllers.) This module uses the full protocol so far specified for the IMDAS.

6. ACCESSING LOCAL DATA

As discussed in Chapter IV a local database system is used to allow each controller to access the data it needs as it needs it. This database is implemented as a flat file system. Each file represents a single relation that uses one or more keys to specify a data record and returns one or more pieces of data.

The interface to this local database is implemented as a general set of procedures contained in one module. This module is part of the ECS program. A separate module attaches each variable in a relation to its actual data type. This module is customized for each controller and is part of the controller program itself (not a part of the ECS program).

7. THE LOAD FILE

The load file is a data file specified by the controller implementor that tells the system what FSM modules (as well as the procedure modules they import) to load and execute. It also specifies the formats for all relations in the local database discussed in section 6 above, as well as the file names used for each relation.

8. FLEXIBILITY

The design of ECS provides a great deal of flexibility. Extra FSMs, used for monitoring or debugging purposes, can be loaded at run time just by changing the load file. For example, an FSM that reads and displays common memory variables could be added to a completed controller program, without having to change a single line of code of the controller.

The configuration of the IWS can be easily changed. Each FSM can communicate with another using common memory either directly within the same controller or over the network to another controller. This means that it would be very easy to combine two controllers into one, which could be convenient in some cases. (For example, it might be convenient to put both the SRIC and the IRC on the same computer, since the SRI and robot can be viewed as a single mechanism during an SRI inspection.) Alternatively, a very large controller could be split into two (or more) controllers, executing on separate computers.

Finally, once ECS is implemented on a different computer, the task of transferring a controller program to that computer is greatly simplified.

VI. DESIGN SPECIFICS FOR IWS CONTROLLERS

This chapter describes the four IWS controllers--the Workstation Controller (WSC), the CMM Controller (CMMC), the Inspection Robot Controller (IRC), and the SRI Controller (SRIC). For each controller, a brief description and task decomposition is presented. For further details, refer to the controller implementation documents listed in Appendix A.

1. WORKSTATION CONTROLLER

1.1. Brief Description

The WSC supervises the equipment controllers of the IWS, and incorporates the University of Virginia (UVA) model for system initialization, restart, and shut down.

As shown in Figure 4, the WSC is subordinate to the Cell Controller and manages the operation of the IRC and the CMMC.

The WSC receives commands from the Cell Controller. As stipulated in the AMRF architecture and UVA system model specifications, these commands are either transition or work order commands. The WSC receives the standard transition commands and issues the standard status responses. The work order commands supported by the WSC are RECEIVE TRAY, INSPECT LOT, and SHIP TRAY.

1.2. Task Decomposition

The hierarchical task decomposition for the WSC is shown in Figure 12. From highest to lowest level, those modules are the Workstation Manager (WSM), the Production Manager (PMGR), the Queue Manager (QMGR), and the Equipment Dispatchers.

The WSM serves as the communications administrator between the Cell Controller and the rest of the IWS controllers. Its functions include receiving and managing Cell Controller commands, reporting the IWS status back to the Cell Controller, and monitoring the operation of the PMGR.

The PMGR's primary responsibility is to coordinate the inspection process by generating the overall equipment task queue.

The function of the QMGR is to set up and supervise the operation of each of the equipment dispatcher modules.

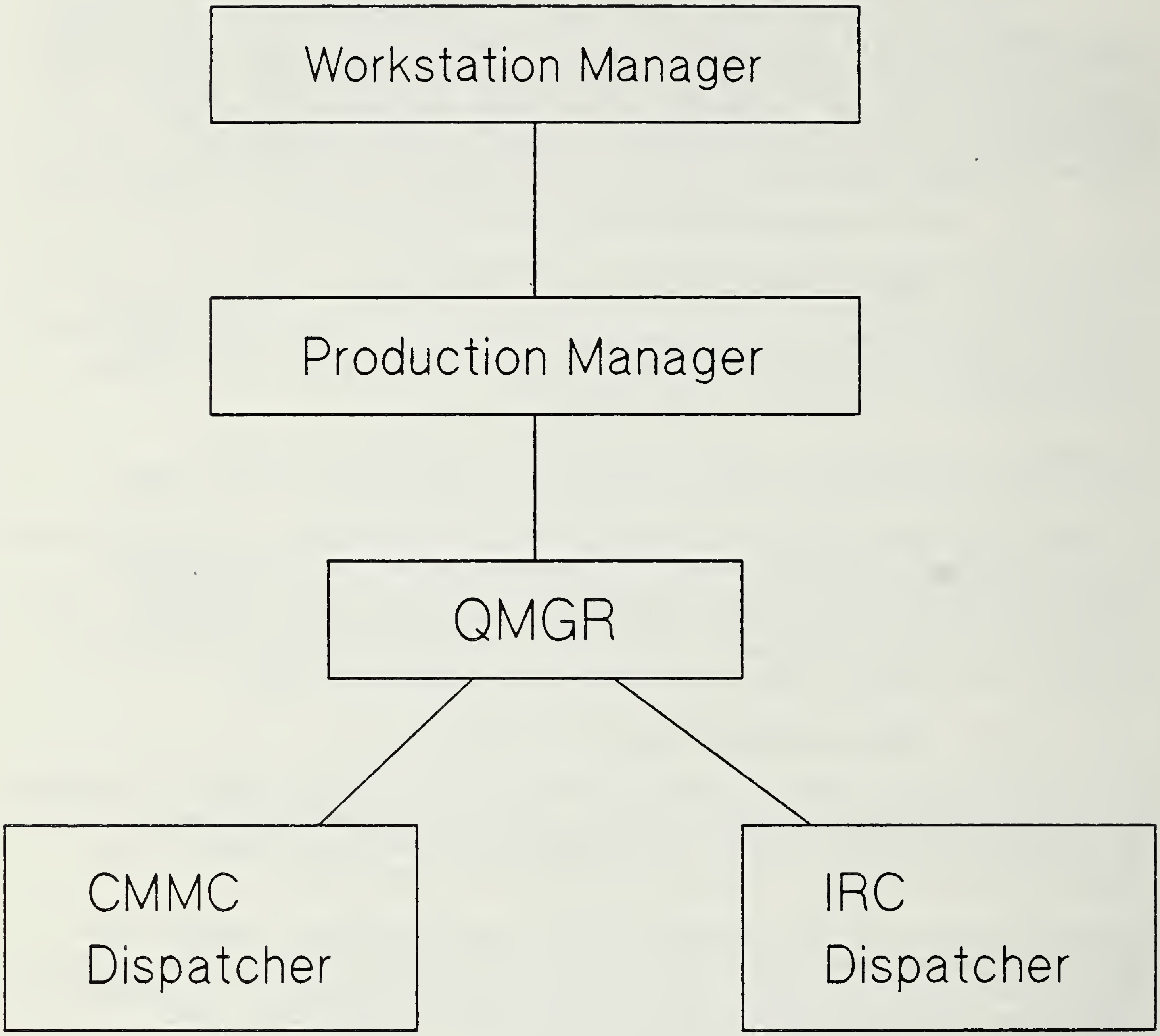

Figure 12. Control Levels for the Workstation Controller

There is one Dispatcher module for each equipment controller. Each Dispatcher is responsible for deciding the next equipment task in the list to be performed, sending back operation status, issuing commands to the equipment controllers, and monitoring the status of the equipment controllers.

2. CMM CONTROLLER

2.1. Brief Description

The CMM Controller is commanded by the Workstation Controller and is the supervisor of the CMM (Figure 4). (In a future implementation, CMMC will be able to access the IMDAS, as will all IWS controllers.)

Its main functions are to retrieve the data required to inspect a specified part and then to direct that inspection. In directing the inspection, the CMMC controls the CMM motion, analyzes the data returned from the CMM, reports the inspection results back to the AMRF (not currently implemented), and displays the results locally on the CMMC's screen.

2.2. Task Decomposition

The task decomposition for CMMC is shown in Figure 13. Listed in their order of hierarchical task decomposition (from highest to lowest), those modules are wsc_cmm, cmm_tol, tolerance, features, surfaces, points, and machine.

The top module, wsc_cmm, interfaces CMMC to WSC, and the bottom module, machine, provides the interface to the CMM. Cmm_tol directs the main tasks of CMMC, namely to retrieve the data to inspect a part and to supervise that inspection.

The intermediate modules (tolerances, features, surfaces, and points) perform the inspection. The module "tolerance" specifies the tolerances to inspect. A dimensional tolerance is based on geometric features of a part, and the module "features" subsequently measures the proper features. It does this by commanding its subordinate, "surfaces," to measure the surfaces that make up the features. Finally, the module "points" is directed to measure the proper points on the specified surfaces.

3. ROBOT CONTROLLER

3.1. Brief Description

As shown in Figure 4, the Inspection Robot Controller (IRC) is subordinate to WSC and is the supervisor of the robot.

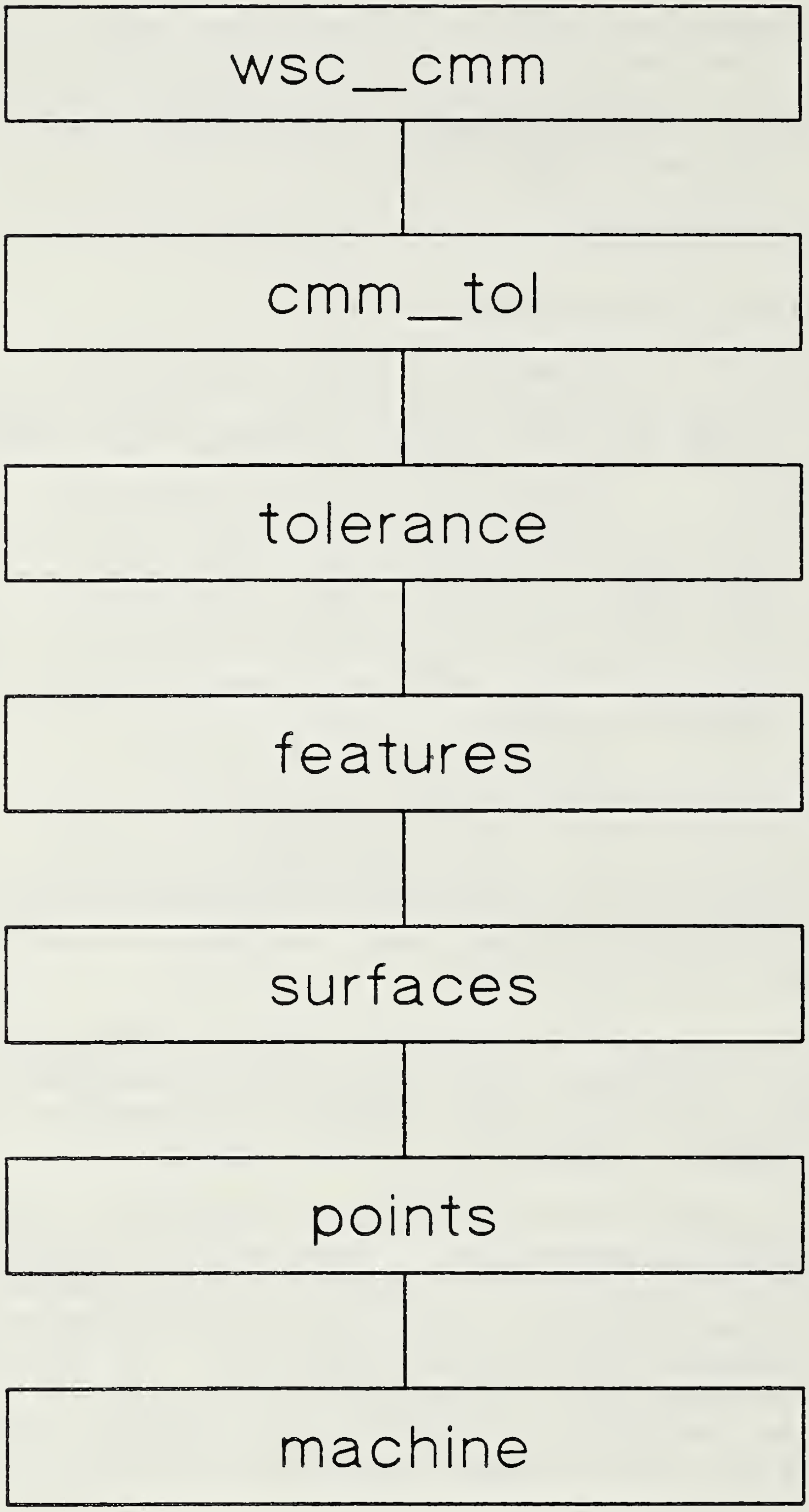

Figure 13. Control Levels for the CMM Controller

Additionally, IRC is the supervisor of SRIC. (In a future implementation, IRC will be able to access the IMDAS.)

The two primary duties of IRC are to control the robot and to supervise SRIC. In addition, before IRC can perform these main duties, it must retrieve the data that specifies the points the robot will move to throughout the IWS.

3.2. Task Decomposition

Figure 14 shows the task decomposition for IRC. The highest level task is wsc_irc. Subsequent tasks in the hierarchy (from highest to lowest) are task, e_move, and machine.

The wsc_irc module interfaces IRC to WSC, and the bottom level module, machine, interfaces IRC to the specific robot used in the IWS.

The module that directs the main functions performed by IRC is called task. The task level functions are subsequently decomposed in the e_move module (which stands for elementary move). The main purpose of e_move is to determine the path the robot must follow from its current location to where it must go to perform its next task.

4. SRI CONTROLLER

4.1. Brief Description

As shown in Figure 4, SRIC is subordinate to IRC and is the supervisor to the SRI and the Automatic Dial Indicator (ADI). (In a future implementation, the SRI Controller will be able to access the IMDAS.)

The main purpose of SRIC is to control those two pieces of equipment (the SRI and the ADI). SRIC also works in concert with IRC to direct the robot to position a part properly in front of the SRI, while the SRI is inspecting the surface finish of a part.

4.2. Task Decomposition

The state machine modules used to implement the SRIC are shown in Figure 15. Listed in their order of hierarchical task decomposition (from highest to lowest), those modules are irc_sri, task, surfaces, points, and, at the bottom, dial and sri.

The top level module, irc_sri, interfaces SRIC to IRC. At the bottom, the module dial interfaces SRIC to the ADI. Likewise, the module sri interfaces SRIC to the SRI. These two modules are directly analogous to the machine module in CMMC.

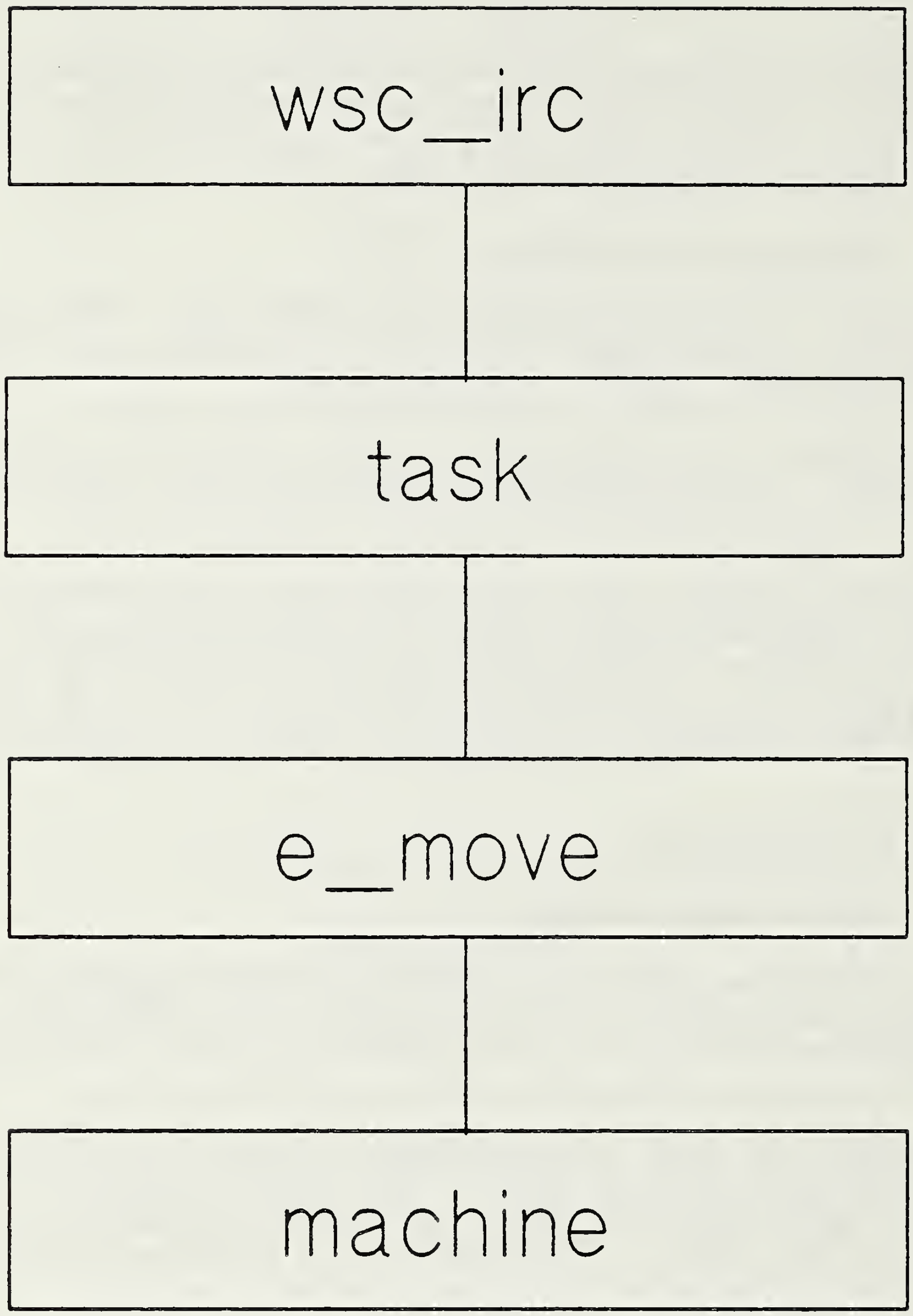

Figure 14. Control Levels for the Robot Controller

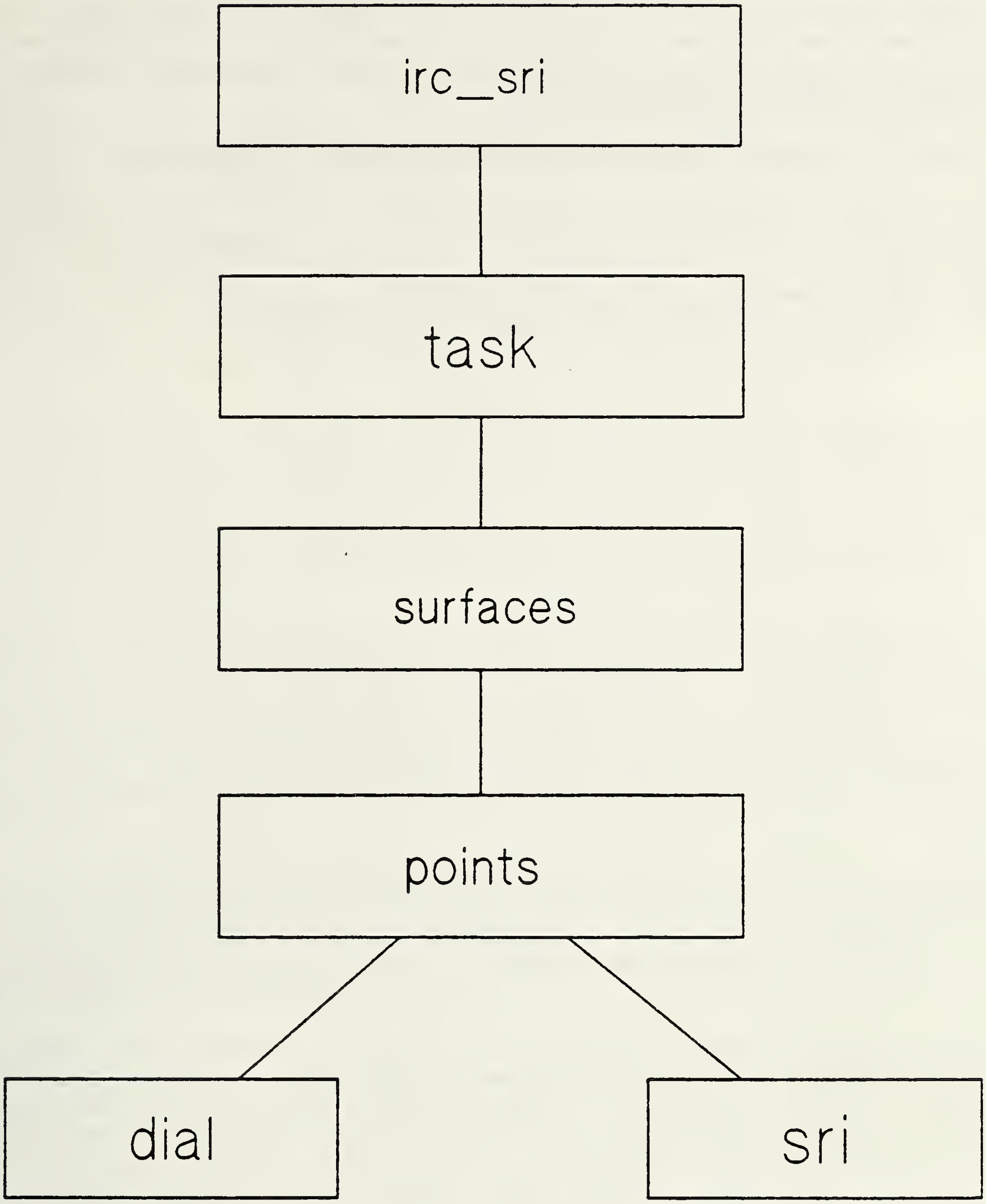

Figure 15. Control Levels for the SRI Controller

The module task performs the main functions of SRIC. This module orders its subordinate, surfaces, to measure a particular surface on the part, and in turn, commands the module points to measure specified points on that surface.

4.3. Why It Is Subordinate to Robot Controller

SRIC has been placed as the subordinate to IRC in the current IWS design. This is to allow a closely coupled operation of the two controllers, and the corresponding equipment they control, when the SRI is inspecting a surface's finish. At that time, the robot and SRI can be thought of as a single mechanism.

VII. OPERATING SCENARIO

This chapter presents a scenario of the IWS from start up through operation. The scenario traces the flow of commands and data throughout the workstation. It would be helpful to refer to Figure 4, showing the logical configuration of the IWS, while following the discussion below.

1. START UP

Initially the IWS operator starts up each of the four IWS controllers. The local network connecting them is not yet established, nor is the connection to the AMRF Network.

The AMRF operator starts a network program that communicates with the network module in the WSC. The latter in turn communicates with the equipment controllers and establishes the local IWS network and subsequently the network from the IWS to the AMRF. This network start up procedure is intrinsic in the network module in each of the four controllers and is transparent to them.

The AMRF Cell Controller can now communicate to the WSC over the network. The Cell Controller issues two types of commands--transition commands and work order commands. Transition commands are used to transfer the IWS from its initial SHUT DOWN state to the READY state, using a protocol based on the UVA model. In the READY state, the IWS can receive work order commands. These commands are used to direct each controller to perform its main functions.

The first command sent from the Cell Controller to the WSC is the transition command SYNC. The WSC uses this command to synchronize its execution with the Cell Controller. (In the current implementation the WSC does not send SYNC commands to its subordinate controllers.)

Next, the Cell Controller sends the IWS the WARM STARTUP transition command. The WSC subsequently sends a WARM STARTUP command to both the IRC and the CMMC. Since the IRC is the SRIC's supervisor, it sends a WARM STARTUP command to the SRIC. The SRIC starts up and reports DONE back to the IRC. Both the IRC and the CMMC starts up and each reports DONE back to the WSC. After the WSC performs its start up procedures and receives status reports of DONE from both the IRC and the CMMC, it reports DONE back to the Cell Controller. The IWS is now in the READY state and can receive work order commands from the Cell Controller.

2. OPERATION

Parts manufactured in the AMRF are sent to the Inspection Workstation (IWS) for dimensional tolerance and surface roughness measurements. The parts are delivered to the IWS by the Material Handling System (MHS) in a tray on an Automated Guided Vehicle (AGV). Once the AGV is on its way to the IWS, the Cell Controller issues a RECEIVE TRAY work order command to the WSC. The WSC uses the information in the work order command to retrieve the tray contents report from the IMDAS. (Currently, this report is not used by the IWS.)

After the MHS delivers the tray of parts to the IWS, and the latter has completed the RECEIVE TRAY work order command, the Cell Controller sends the IWS the work order command INSPECT LOT which directs the IWS to inspect the parts in the tray with a specific process plan. This plan, which the WSC uses to coordinate its subordinate controllers to inspect the particular tray of parts, is called the operation sheet. The WSC retrieves the operation sheet from the IMDAS (currently being read in from a local data file residing in the WSC). The WSC uses the operation sheet to generate a list of tasks for each of the controllers it supervises--the CMMC and the IRC.

Subsequently, according to the established plan, the WSC issues the appropriate commands to the CMMC to perform a dimensional tolerance inspection, and to the IRC to either perform an inspection with the Surface Roughness Instrument Controller (SRIC) or to transfer a part.

In order to inspect a part, the CMMC requires an inspection plan; therefore, before sending the command to inspect a part, the WSC issues the work order command LOAD DATA to the CMMC to load the data for the part. The data includes part data (specifying the geometry and topology of the part) as well as an inspection plan. The inspection plan specifies which part features are to be measured and the tolerances imposed on those features. (This is the equipment level process plan.) Currently, all of the part data and inspection data resides locally on the CMMC.

While the CMMC is loading data, the WSC sends the work order command to the IRC to TRANSFER the part from its current location (the tray, initially) to the CMM, where it will be inspected.

After the part has been transferred to the CMM, the WSC sends the INSPECT PART work order command to the CMMC. A CMM part inspection consists of reading the inspection plan to find out what type of tolerance measurement is to be performed and what part feature or features are to be inspected, determining which surfaces pertain to those features and which points to inspect on those surfaces, and lastly probing those points in order to calculate the actual dimensions of the part. When the CMMC completes the inspection, it reports a status of DONE back to the WSC.

Subsequently the WSC issues a TRANSFER work order command to the IRC to return the part to the tray. If there are any CMM inspections left to perform on the remaining parts in the tray, the WSC repeats the above procedure on each of those parts in turn.

The SRI inspection is a bit more complicated since it involves both the robot and the SRI. While the CMMC is busy inspecting a part, the robot is free to inspect a part with the SRI, concurrently with the CMM inspection.

The WSC first sends the work order command PECK to the IRC to find the precise position of the part's surface from the robot's gripper. (This measurement is necessary to precisely position the surface to be measured the proper distance away from the SRI.) Upon receiving the PECK work order command, IRC directs the robot to pick up the part, take it over to the Automatic Dial Indicator (ADI), and repeatedly move towards it a specified distance (peck) until the ADI gets a non-zero reading.

Like the CMMC, the SRIC needs an inspection plan in order to perform its operation on the particular part. WSC sends the work order command LOAD SRI DATA to the IRC, from which it is relayed to the SRIC. The SRIC then loads up the data required. (As is the case with the other controllers, all data currently resides on the local controller.)

After the necessary data is loaded, the WSC issues the work order command INSPECT WITH SRI to the IRC to begin the inspection. The robot then takes the part over to the SRI, positions it the proper distance away from it, and sends the work order command INSPECT to the SRIC.

At this point, the SRIC takes over the surface inspection. The IRC monitors the SRIC's progress. Whenever the SRIC requests a change of position or orientation of the surface in front of the SRI, the IRC directs the robot to honor that request. When the surface inspection has been completed, the SRIC reports a status of DONE back to the IRC, which in turn, reports DONE back to the WSC. The robot is now free to be commanded again by the WSC.

Consequently, the WSC issues the work order command TRANSFER to return the part just inspected by the SRI to the tray.

When all of the inspection procedures in the operation sheet are completed, the WSC signals back to the Cell Controller that it is DONE with the INSPECT LOT command.

When the Cell Controller decides to have the MHS take the tray away, it sends the WSC the work order command to SHIP TRAY which directs the WSC to release it. This allows the MHS to pick up the tray and deliver it to its next destination.

This completes an inspection of an entire tray of parts at the IWS. When a new tray of parts is ready to be inspected, the whole procedure herein described is repeated.

APPENDICES

A. IWS DOCUMENTATION LIST

1. H. T. Moncarz, Architecture and Principles of the Inspection Workstation, NBSIR 88-3802, June 9, 1988.

2. H. T. Moncarz, Implementation of the Execution Control System of the Inspection Workstation, NBSIR 88-3787, May 19, 1988.

3. H. T. Moncarz and T. H. Hopp, Implementation of the CMM Controller, to be published as an NBSIR, 1988.

4. H. T. Moncarz and T. V. Vorburger, Implementation of the SRI Controller, NBSIR 88-3794, June 2, 1988.

5. H. T. Moncarz and B. Borchardt, Implementation of the Inspection Robot Controller, NBSIR 88-3772, April 21, 1988.

6. S. A. Osella, Implementation of the Workstation Controller, to be published as an NBSIR, 1988.

7. J. Zimmerman, Inventory of Equipment in the Inspection Workstation, NBSIR 88-3775, May 5, 1988.

8. H. T. Moncarz, S. A. Osella, B. Borchardt, and R. Veale, Operations Manual for the Inspection Workstation, NBSIR 88-3766, April 21, 1988.

9. J. Zimmerman, Recommended Technical Specifications for Procurement of Commercially Available Systems for the Inspection Workstation, NBSIR 88-3779, 1988.

B. REFERENCES

1. C. Furlani, et al., The Integrated Manufacturing Data Administration System, to be published as an NBSIR, 1988.

2. D. R. O'Halloran and P. F. Reynolds, Jr., "A Model for AMRF Initialization, Restart, Reconfiguration, and Shutdown", NBS/GCR 88/546.

3. T. H. Hopp, AMRF Database Report Format: Part Model, NBSIR 87-3672.

4. T. H. Hopp and K. C. Lau, "A Hierarchical Model-Based Control System for Inspection," Automated Manufacturing, ASTM STP 862, L. B. Gardner, Ed., American Society for Testing and Materials, Philadelphia, 1985, pp. 169-187.

5. C. R. McLean, Control Systems Command Feedback Protocol Document, to be published as an NBSIR.

6. A. J. Wavering and J. C. Fiala, The Real-Time Control System of the Horizontal Workstation Robot, NBSIR 88-3692.

7. D. Libes and E. Barkmeyer, "The integrated manufacturing data administration system (IMDAS)--an overview", International Journal of Computer Integrated Manufacturing, Vol. 1, No. 1, pp. 44-49.

C. GLOSSARY (and abbreviations)

ADI Abbreviation for the Automatic Dial Indicator.

automatic dial indicator
Instrument used to measure the distance that a spring mounted stem is depressed.

common memory system
Manages communications between state machines.

CMM Abbreviation for the Coordinate Measuring Machine.

CMMC Abbreviation for the CMM Controller.

controller
Supervises the operation of a mechanism, another controller, or both.

coordinate measuring machine
Machine used to measure the dimensions of a part.

data server
Software module that interfaces the controller it resides on to the data it requires.

ECS Abbreviation for the execution control system.

execution control system
Computer program that runs on each controller computer and implements the AMRF design principles. This program loads and executes those modules which determine which controller is actually being run.

FSM Abbreviation for finite state machine. Strictly speaking, the software control modules used in the IWS are state machines, not finite state machines. However, for convenience, the abbreviation FSM is kept.

inspection workstation
AMRF workstation that inspects parts for dimensional tolerance and surface finish.

IRC Abbreviation for the Inspection Robot Controller.

IWS Abbreviation for the Inspection Workstation.

load file
Data file that specifies what state machines ECS should load and execute.

logical architecture
Specifies the direction of commands and statuses between controllers and between controllers and equipment.

network
The connections (both hardware and software) that connects the IWS controllers together and to the rest of the AMRF. Local network refers to the former only.

network interface unit
This device, connected to each controller in the IWS and to the AMRF, provides the network link (both hardware and software) to the IWS.

NIU Abbreviation for network interface unit.

physical architecture
Specifies the physical connections among the controllers and equipment.

SRI Abbreviation for the Surface Roughness Instrument.

SRIC Abbreviation for the SRI Controller.

state machine
Software control unit with outputs dependent on inputs to it plus its internal state. This is the building block for the IWS control software.

surface roughness instrument
Machine that measures the optical scattering off the surface of a part that can be correlated with its surface roughness.

transition commands
Commands used to transfer the IWS to a new state (specified by the UVA protocol).

UVA Protocol
Model, proposed by research group from the University of Virginia and adopted by the AMRF, that specifies the start up and shut down sequence for the AMRF as a whole as well as every controller within the AMRF.

work order commands
A command accepted by a controller when it is in ready state, and used to perform one of its main functions.

WSC Abbreviation for the Workstation Controller.

READER COMMENT FORM

ARCHITECTURE AND PRINCIPLES OF THE INSPECTION WORKSTATION

This document is one in a series of publications which document research done at the National Bureau of Standards' Automated Manufacturing Research Facility from 1981 through March, 1987.

You may use this form to comment on the technical content or organization of this document or to contribute suggested editorial changes.

Comments: ____

____

____

____

____

____

____

____

____

If you wish a reply, give your name, company, and complete mailing

address: ____

____

____

____

What is your occupation? ________________________________

NOTE: This form may not be used to order additional copies of this document or other documents in the series. Copies of AMRF documents are available from NTIS.

Please mail your comments to:

AMRF Program Manager
National Bureau of Standards
Building 220, Room B-111
Gaithersburg, MD 20899

FORM NBS-114A (REV. 11-84)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 88-3802	2. Performing Organ. Report No.	3. Publication Date JUNE 1988

4. TITLE AND SUBTITLE

"Architecture and Principles of the Inspection Workstation"

5. AUTHOR(S)

Howard T. Moncarz, Machine Intelligence Group

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
U.S. DEPARTMENT OF COMMERCE
GAITHERSBURG, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

National Bureau of Standards
Gaithersburg, Maryland 20899

10. SUPPLEMENTARY NOTES

[ ] Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This document describes the Inspection Workstation (IWS) of the AMRF, in particular its software control architecture. The IWS uses a coordinate measuring machine for dimensional metrology, an optical roughness gage for surface finish inspection, and a robot for part handling within the workstation. Each of these three pieces of equipment is supervised by a separate controller, and a fourth controller coordinates these equipment controllers. The workstation coordination and all inspection procedures are specified entirely by data. In addition, the IWS is integrated into the AMRF control hierarchy, product data base, communications network, and material handling system. The software control architecture described herein allows this level of data-driven control and AMRF integration.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

AMRF; data-driven control; dimensional tolerance; Inspection Workstation, surface roughness; workstation architecture

13. AVAILABILITY

[X] Unlimited
[ ] For Official Distribution. Do Not Release to NTIS
[ ] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402.
[X] Order From National Technical Information Service (NTIS), Springfield, VA 22161

14. NO. OF PRINTED PAGES

54

15. Price

$13.95

USCOMM-DC 85-6006