SANDIA REPORT

SAND87—1563 • UC—28
Unlimited Release
Printed December 1988

32-2168396
cy 1

The Structure and Function of the PBFA-II Control System

8232-2//068396
00000001

Steven Y. Goldsmith

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND87-1563

THE STRUCTURE AND FUNCTION OF THE PBFA-II CONTROL SYSTEM

December 1988

Steven Y. Goldsmith
Digital Systems Division 7545
Sandia National Laboratories
Albuquerque, NM 87185-5800

ABSTRACT

This report identifies major hardware and software components comprising the Particle Beam Fusion Accelerator II (PBFA-II) control system. Both the system architecture and the theory of operation are discussed. It is intended to be a general introduction for engineers involved in the maintenance and extension of the control system. The companion to this document, "System Manager's Technical Guide for PBFA-II Control/Monitor System", SAND87-1564, contains the detailed documentation on the system software as well as procedures for configuration management. The material covered in this document is prerequisite to SAND87-1564.

AUTHOR'S NOTE

This document provides an overview of the PBFA-II C/M system software structure. The system is extensive, comprising over 250,000 lines of code. No single document can adequately provide a detailed description of the design. The documentation philosophy of the PBFA-II project is to provide a Computer-Aided Software Engineering (CASE) tool to serve as the primary documentation for the maintenance phase. This documentation is provided through the TAGS[1] system, in the form of approximately 3,000 pages of data flow diagrams, data descriptions, process descriptions, and comment pages. Throughout this document are references to the relevant TAGS documents that provide more detailed information. I suggest that anyone interested in learning about the C/M system read this document first, then become familiar with TAGS through SAND-1564 and the TAGS manuals, and finally return to this document and use it to guide a browsing session at the TAGS terminal.

[1]TAGS is a requested trademark of Teledyne-Brown Engineering, Huntsville, AL.

CONTENTS

		Page
1.0	INTRODUCTION	5
2.0	BACKGROUND	6
	2.1 Design Philosophy	6
	2.2 General System Requirements	7
3.0	COMPUTER SYSTEMS	8
	3.1 Design Considerations	8
	3.2 The Computer Control Network	10
	3.3 The Integrated Control and Display System	10
	3.4 The Operations Console	12
	3.5 The Local Control Units	12
	3.6 The Timing and Firing System	12
4.0	SOFTWARE ARCHITECTURE	17
	4.1 Design Considerations	17
	4.2 The Host-Satellite Interface Subsystem	19
	4.3 The Operator-Machine Interface	25
	4.4 The System Data Bases	27
	4.5 The Control Action Tasks	37
	4.6 The Transaction Processors	38
	4.7 The Remote Manual Capability	39
	4.8 The Unit Actions Capability	41
	4.9 The Integrated Control Capability	43
	4.10 The Automatic Staging Capability	45

FIGURES

1.	PBFA-II Control System	9
2.	Control Network Topology	11
3.	Integrated Control and Display Components	13
4.	Operations Console Components	14
5.	Typical Local Control Unit	15
6.	Timing and Firing System	16
7.	Layered Capabilities	20
8.	Event Transaction Handling by the HSI	22
9.	Control Transaction Handling by the HSI	24
10.	Command and Display Transaction Handling by the OMI	26
11.	Points Data Base System	29
12.	Machine Elements Data Base System	31
13.	Shot Parameter Configuration Subsystem	33
14.	Point Data Recording Subsystem	36
15.	Remote Manual Capability	40
16.	Unit Action Capability	42
17.	Integrated Control Capability	44

1.0 INTRODUCTION

The Particle Beam Fusion Accelerator II (PBFA-II) facility is unique in the world of superpower machines and places unique demands on a computer-based control system. PBFA-II is comprised of a large number of elements which are in most cases specialized equipment designed and constructed solely for use in the facility. In addition to being unique, the elements vary in their characteristics and function. They differ in the number and mix of I/O points (analog or digital), the complexity of their control sequences, the bandwidth of their transfer functions, and in the level of control required (direct digital control or supervisory control). The computer system required to control such a diverse set of elements must have the flexibility to provide a wide variety of custom control procedures. The system must also accommodate the different sensor scan rates and response time requirements.

The equipment comprising PBFA-II have characteristics that resemble those found in process plants, nuclear reactors, laboratory instruments, security systems, and power-conditioning systems. Consequently, the control system must have the same capabilities found in several classes of computer-based control systems that have evolved to handle these applications. Examples of these systems are: industrial process control systems, SCADA systems for power grid control, batch manufacturing sequencers, and laboratory automation systems.

Commercially-available control systems do not meet all of these requirements. To control the PBFA-II facility requires the synthesis of features from several of these classes of control systems, and some unique features as well. Such a custom control system was designed and constructed under the direction of the Digital Systems Division, 7545.

To maintain the PBFA-II facility and conduct experiments requires a large force of operations engineers. The operations staff must perform a broad range of tasks, using the control system, from diagnosing a single sensor failure to conducting a complex experiment. Consequently, the control system must operate in several modes, often simultaneously, to effectively support this broad range of activities. In each mode the tasks must be carefully partitioned between human and computer, and an interface provided between the two that transmits the proper information in a timely manner.

The research aspects of the PBFA-II project also place constraints on the control system design. Some of the elements comprising the accelerator themselves are the subject of research. Because the control system must interface to all elements, changes in element designs accumulate as changes in control system criteria. To effectively participate in the PBFA-II project under these circumstances requires that the control system project be conducted as a research and development project. The system design must accommodate changes in scope, function, and performance in response to newly discovered information, sometimes on a weekly basis. Isolation of changes

within the system design is crucial to the success of the project. This research aspect of the project, together with resource limitations, constrain the final system a greater degree than does the technical challenges of accelerator control.

2.0 BACKGROUND

2.1 Design Philosophy

Because the PBFA-II project was conducted in a fast-track mode, the deployability of the control system is critical to meeting the milestones set by the project office. Consequently, the system relies heavily on vendor-supplied hardware and software from standard product lines to facilitate the deployment of the system. The design effort is centered on effective utilization of these standard products. The most desirable situation is to have a single vendor supply all computer hardware and software components, thereby leveraging the project schedule with the integration efforts of the vendor. The PBFA-II control system is based on Hewlett-Packard's A-Series technical computer systems. The following products are heavily utilized: the HP 2250 Measurement and Control System and the Control/1000 product; the DS/1000 Distributed System product; the IMAGE/1000 Data Base product; and the RTE-A Real-Time Operating System.

The flexibility required of the control system due to the nature of the project and the diversity of machine elements can only be achieved by a software-intensive design philosophy; a philosophy which dictates that hardware-based control actions are reserved for functions with the most demanding timing requirements. The development cycle for custom hardware is more time consuming and costly than for software. In addition, software is a far more tractable medium than hardware, a fact that is critical when faced with the incomplete and changing system criteria encountered on the PBFA-II project.

Software maintainability is a concern when changes to the system occur frequently. To insure the quality of the software is adequate to support maintenance activities, a set of quality standards are enforced. To this end, the entire software system is written in highly-structured FORTRAN 77.

The FORTRAN language package available for the HP A-Series computers is well integrated with the operating system and the FORTRAN compiler is highly optimized. No other languages are allowed for processes executing on the control system network. The bulk of the system is specified using Structured Analysis and Structured Design techniques. In addition, every program and subroutine module is required to have a standard documentation header describing the purpose and characteristics of the module, and must conform to rigid documentation standards.

2.2 General System Requirements

The general requirements for the PBFA-II control system design are called out in the PBFA-II Control Monitor System Plan.[1] Of the system goals outlined in the plan, the following have direct bearing on the system architecture: (1) minimize the number of operators required; (2) provide consistent and repeatable operating procedures; (3) support installation and checkout of accelerator equipment during the facility integration phase; (4) provide for the acquisition and playback of equipment performance data; and (5) provide a control system that is reliable.

To minimize the number of operators needed to operate the facility, the control system must have the following features: (1) a sophisticated, centralized operator-machine interface system that allows on-demand access to many status and control panels; (2) a set of programs that automate many of the time-consuming manual control procedures; and (3) a broad scope of control whereby the majority machine elements are interfaced to the control system. For a small number of operators to effectively operate the entire PBFA-II facility, all status information and command points must be centralized. If status and command points are not centralized, the coordination of the facility requires more operators, better communications between operators, and more time. Centralization of information necessitates interfacing to nearly all machine elements to insure access to interdependent data among machine elements.

At the central operations console, operators must be able to choose which of the many possible status and command panels are required to conduct the tasks at hand. The ease with which operators can switch between panels is important to allow expansion of the number of tasks a single operator can handle simultaneously. Equally important is the nature of automatic control programs and how the control programs interface to the operator. When conducting multiple tasks simultaneously, the operator should be required to initiate only the control sequence, specifying the target state parameters, and to verify the sequence was initiated. The system must notify the operator when the sequence is complete, the new state has been reached, or when an abnormal event occurs that may prevent completion. Thus, the operator is freed from the task of continuously monitoring the progress of the sequence. This, in turn, reduces the amount of information the operator must process to the minimum required for the task.

The automation of control sequences for individual machine elements lends a degree of consistency and repeatability to facility operations. Errors due to the uncoordinated operation of many interacting machine elements are still possible, however. To insure a high degree of consistency and repeatability requires the automation of major tasks involving integrated control of multiple elements. Programs that perform these tasks must contain a large amount of

knowledge on standard operating procedures and the modes of interaction between machine elements. Another important requirement for automated operation is a method for storing and retrieving standard sets of operating parameters. A facility with many parameters such as PBFA-II must have such methods to allow experiments to be repeated accurately.

During the installation phase, and during subsequent routine maintenance operations, individual control and status points are used to validate the interfacing and calibration of equipment. Manual operations may also be necessary to determine the correct operating procedure before automating that procedure. To support these activities, the control system must have a manual mode of operation available from the operations console whereby individual control points may be exercised and status readings verified. Historical records of equipment status are needed to evaluate performance, diagnose malfunctions, and develop better control procedures. A recording and archiving system is required that stores state versus time records for both status and control points. Historical records are also necessary for archiving the results of a shot.

Because the control system is the central point for all facility operations, it must be reliable and configured for fault tolerance. The availability of the control system must be high also, requiring a small Mean-Time-to-Repair. Without these attributes, a single fault in the control system can prevent any major operation from taking place.

3.0 COMPUTER SYSTEMS

3.1 Design Considerations

The fundamental structure of the control system is based on the distributed processor architecture common to nearly all large computer-based control systems. A distributed system provides task concurrency, the capability for incremental growth, and enhanced reliability to the control system.

There are three classes of processors in the PBFA-II control network. A central processor handles integrated control functions and operator displays. Actual generation of display graphics is performed by a pair of dedicated processors to increase the responsiveness of the operator actions at the central console. Control of machine elements is handled by the third class of processors.

Figure 1 shows the structure of the control system, including the allocation of local control processors to specific machine elements. Allocation is based primarily on leveling the volume of sensor and actuator data on each local control node. The exception is the safety control unit which is dedicated solely to performing safety-related actions.

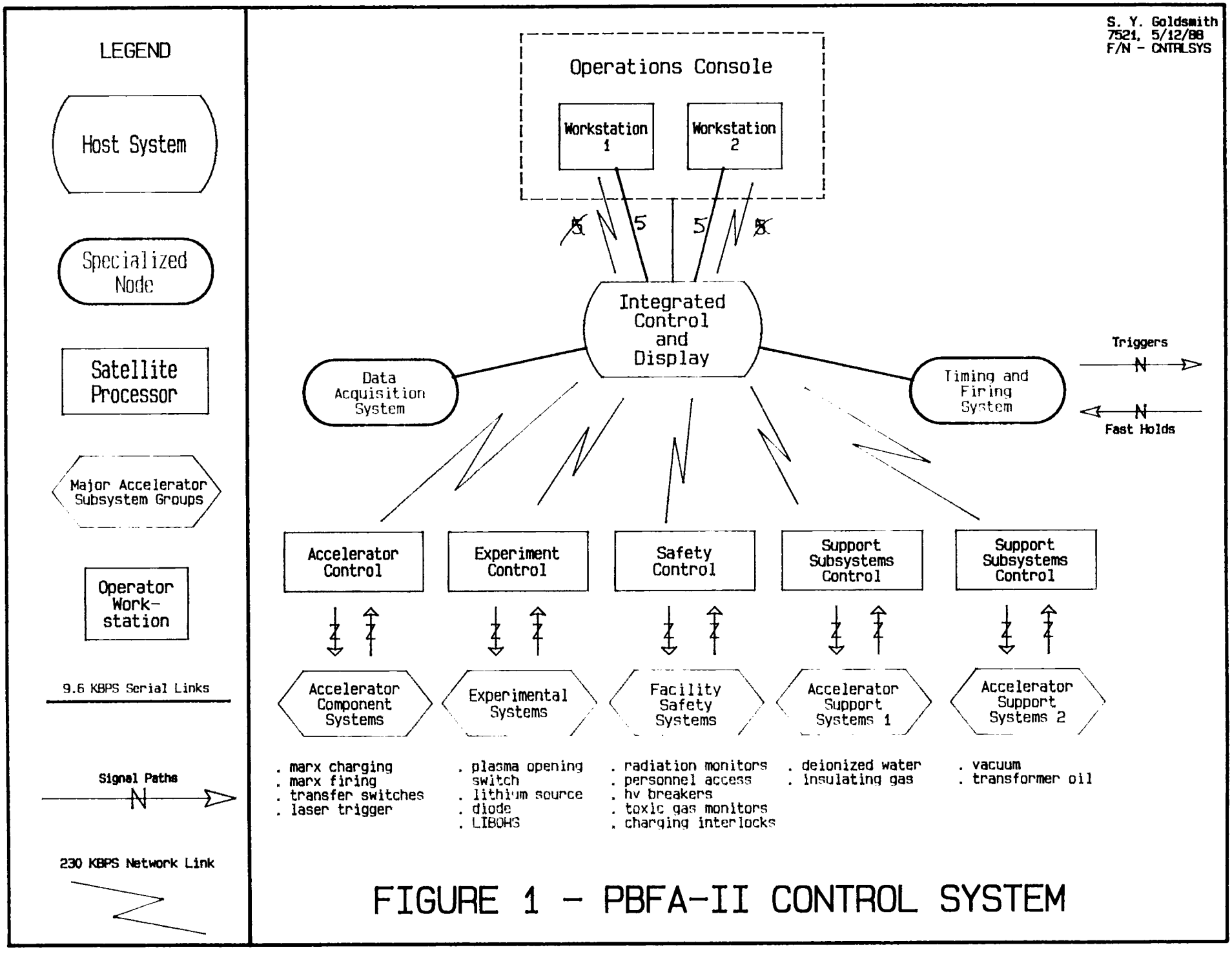

FIGURE 1 - PBFA-II CONTROL SYSTEM

Processors are interconnected via point-to-point links. Point-to-point connections were chosen to allow some Local Control Units to gain access to the central node on a priority basis without bus-contention delays. Cable routing constraints, primarily EMP and radiation, and installation costs prohibited a ring network from being deployed in the facility.

Early in the project, the primary environmental concern was the severe electromagnetic pulse threat produced by the accelerator. To reduce the pickup in signal leads, processors were to be distributed to every major subsystem site and housed in shielded enclosures. Communications between local control processors and the central system was to be accomplished via fiber-optic links. Five months into the design phase, a higher baseline operating voltage for the accelerator was adopted causing a potential threat to local processors from Bremsstrahlung radiation.[2] Local control processors had to be relocated to the shielded control room which housed the central computer system. The challenge became fitting the local control processors in the already crowded screen room and routing signal cables from sites throughout the facility into the screen room. Cables to and from the local control processors are routed through two central interface panels that heavily filter all signals. These panels allow penetration of the shield room while maintaining the electrical integrity of the shield room.

In order to insure that the control system would be reliable enough to conduct experiments, redundancy is built into the component with the poorest availability and highest criticality: the operations console. The console was partitioned into two independent workstations. These workstations are parallel-redundant beginning at the central computer system I/O bus. In addition to redundancy, computer equipment with high reliability is used to implement the system.[3]

3.2 The Computer Control Network

Control of the PBFA-II facility is accomplished by a distributed computer network consisting of nine computer systems: five Local Control Units (LCU), an Integrated Control and Display System (ICDS), two Console Graphics Processors (CGP), and a development system. The computer systems are interconnected in a star configuration with the ICD system as the central node, as shown in Figure 2. Communications between computers is performed over synchronous serial links. The links transmit and receive at a speed of 256 kilobaud under the HDLC protocol.

3.3 The Integrated Control and Display System

The ICD system is based on a Hewlett-Packard A-900 processor that performs three million instructions per second. The processor is configured with three Mbytes of battery-backed,

S. Y. Goldsmith
7521. 2/9/88
F/N - NETWKTOP

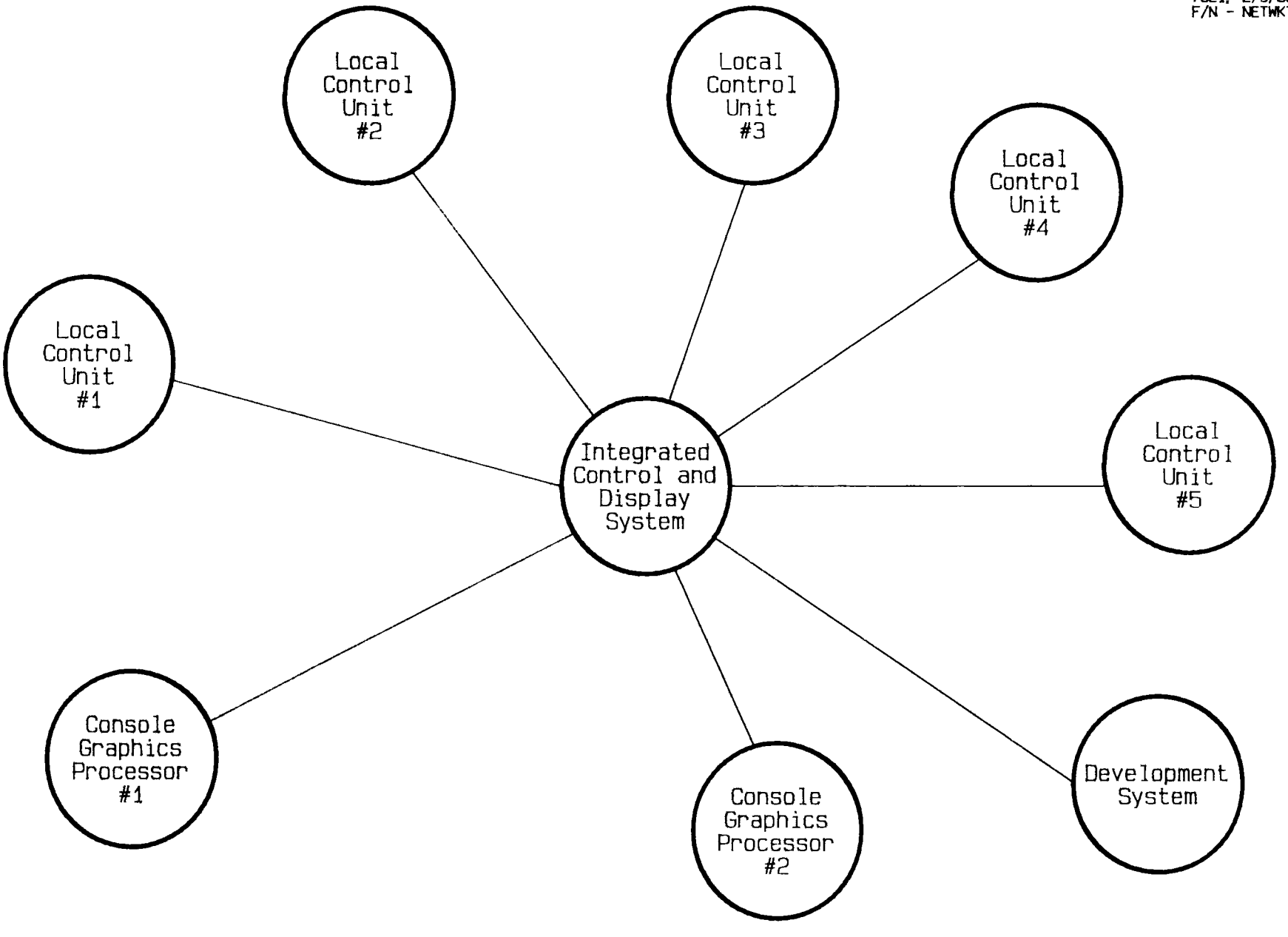

FIGURE 2 - CONTROL NETWORK TOPOLOGY

error-correcting memory. A 132 Mbyte, fixed disk provides non-volatile storage for system software and the various data bases. The ICD system components are shown in Figure 3.

Peripheral devices interface with the ICD system through two eight-channel, asynchronous serial multiplexors. Each multiplexor has an aggregate throughput of 76.8 kilobaud. The allocation of multiplexor ports to peripheral devices is as follows: one port for each of the seven touch-panel input devices on the Operations Console; one port for each of the two operator terminals on the Operations Console; one port each for the timing and firing unit, and the data acquisition system; two ports for the Shot Data Reduction station, one port for the graphics terminal and one port for the laser printer [DOC:ICDS SEC:SBD].

3.4 The Operations Console

The Operations Console, shown in Figure 4, consists of two independent workstations. Workstation A is comprised of four graphical interface units, a Console Graphics Processor, and an operator terminal. Workstation B is comprised of three graphical interface units, a Console Graphics Processor, and an operator terminal. The workstations are interfaced to the ICD system through separate serial multiplexors. The Console Graphics Processors are HP A-600+ computers capable of 1 MIPS. The processors are configured with 1 Mbyte of battery-backed memory. Each CGP has a 15 Mbyte fixed disk that stores system files and the picture generation programs. Video cards in the Console Graphics Processors drive Conrac color monitors over RS-373 RGB interfaces [DOC:CDCS SEC:SBD].

3.5 The Local Control Units

The Local Control Units are based on the A-600+ processor and the HP 2250 Measurement and Control Subsystem. Each LCU is configured with 1 Mbyte of battery-backed memory.

The Local Control Units are memory-based and rely on the ICD system to download the software system at boot-up time. Figure 5 shows the basic components of a typical LCU [DOC:ACMP1-5 SEC:SBD].

3.6 The Timing and Firing System

Triggering the accelerator requires a series of pulses synchronized in the nanosecond regime. The task of generating the pulses is allocated to a specialized hardware system based on the CAMAC standard as shown in Figure 6. This system is comprised of a series of interconnected, programmable delay generators and trigger amplifiers. The delay settings are programmed over a serial link by the ICD system. A timing sequence may be triggered upon command from the ICD system or by a command from the operator's manual, Fire/Hold Interface.

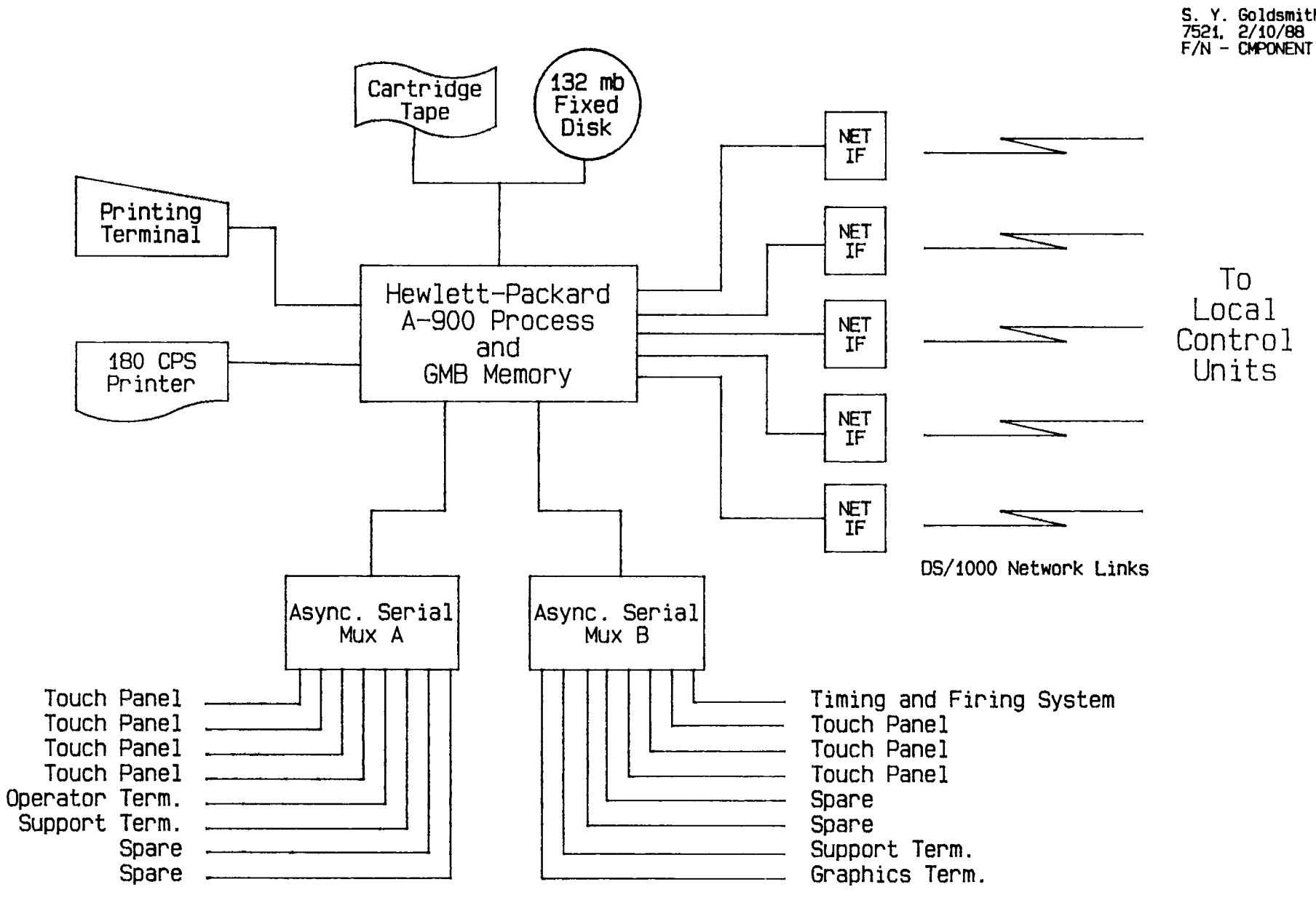

FIGURE 3 - INTEGRATED CONTROL AND DISPLAY COMPONENTS

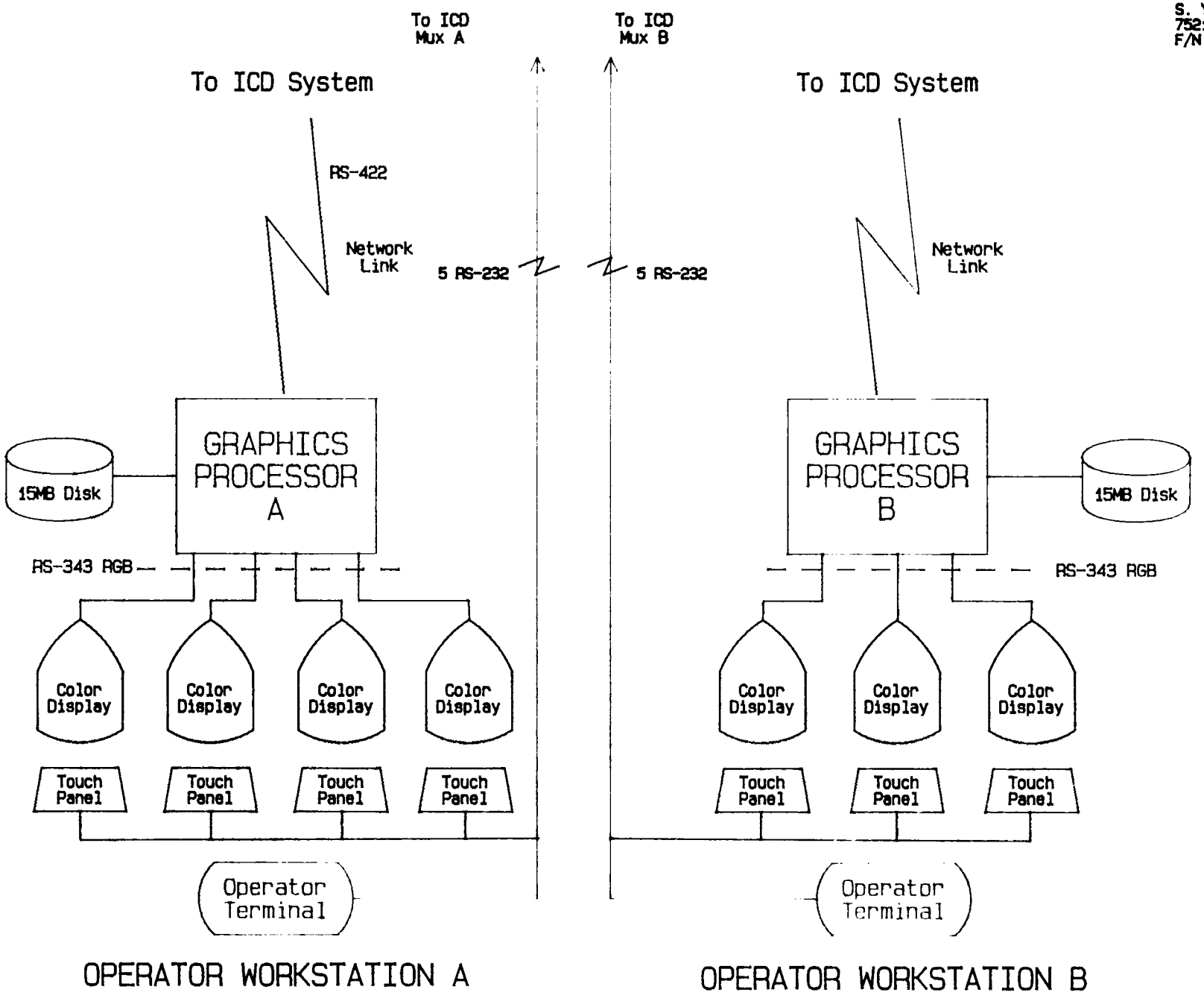

FIGURE 4 - OPERATIONS CONSOLE COMPONENTS

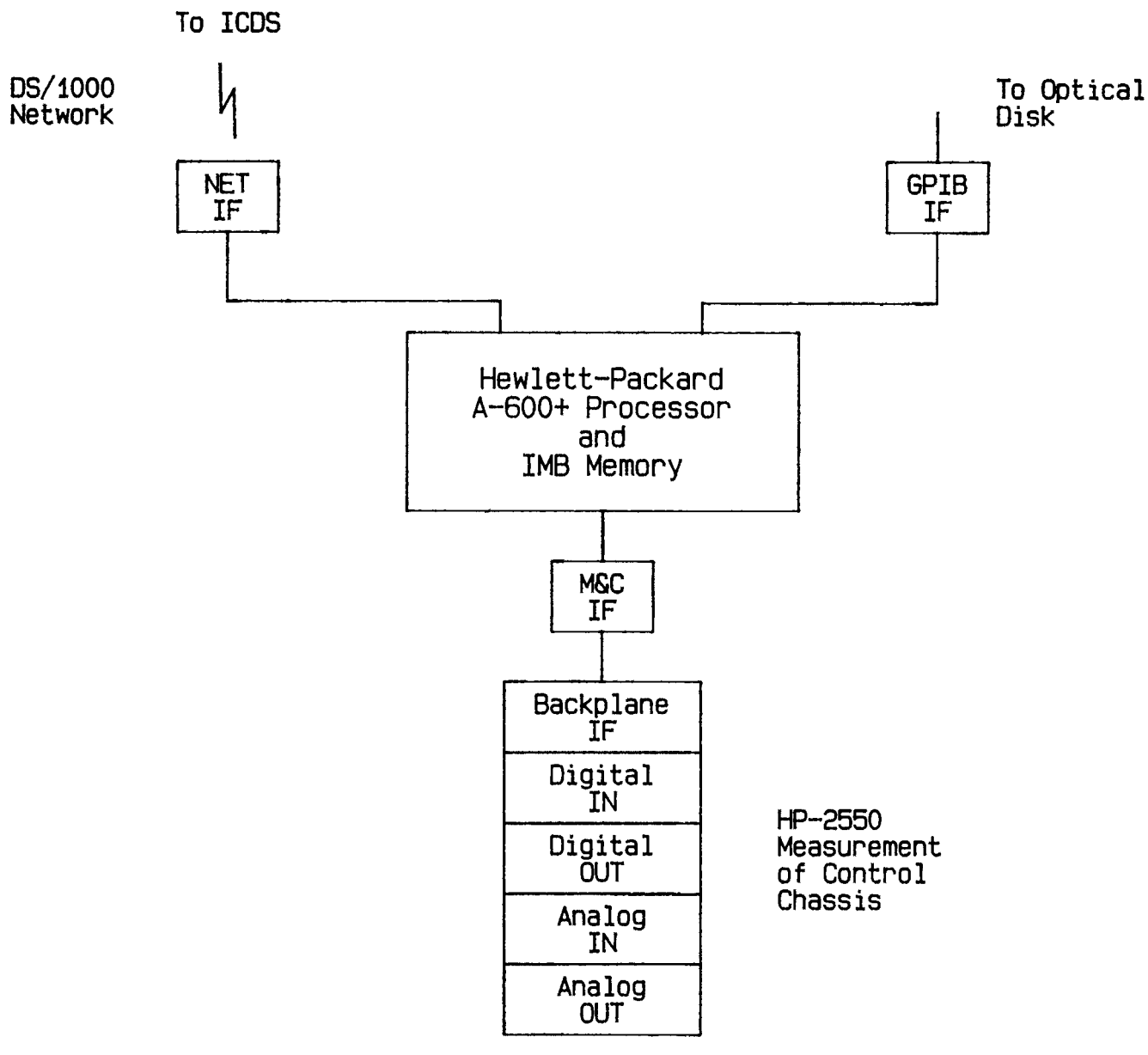

FIGURE 5 - TYPICAL LOCAL CONTROL UNIT

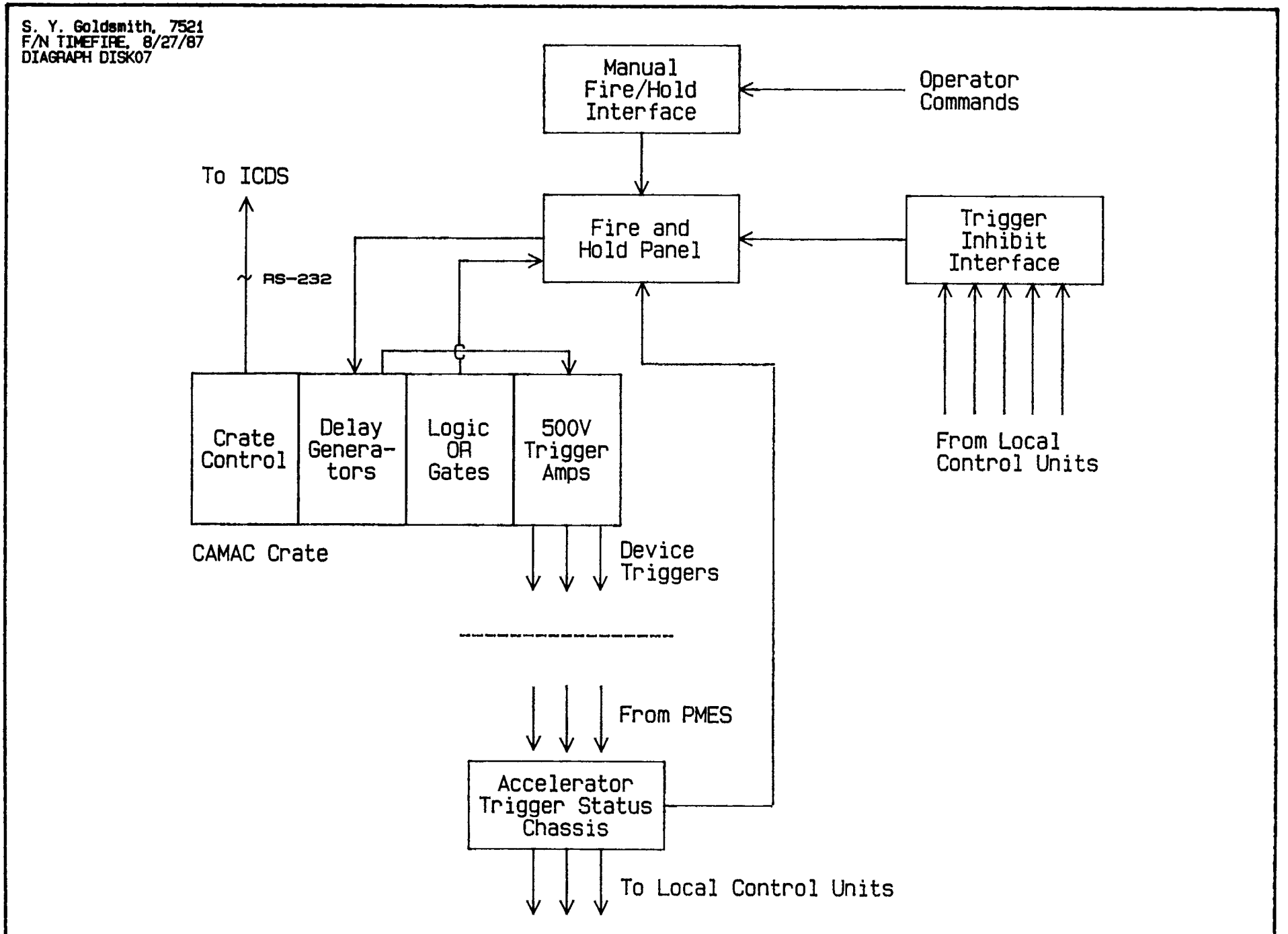

FIGURE 6 - TIMING AND FIRING SYSTEM

Emergency conditions sensed by the Local Control Units or sensed by a special chassis will cause an inhibit signal to be generated. The purpose of this signal is to prevent the initial trigger to the timing delay generators from being issued. Each Local Control Unit has a dedicated binary output that may be set and reset by any process executing on the LCU. When conditions sensed by control software indicate the accelerator cannot be fired in a safe manner, the inhibit signal is set. This signal is sensed by the Trigger Inhibit Interface, which notifies the Fire and Hold Panel of the signal state. The Fire and Hold Panel latches the inhibit signal and opens a switch in series with the primary trigger, thereby, preventing accelerator firing.

The Timing and Firing system provides signals indicating the firing status of the various capacitor banks and high voltage devices. These signals are routed to the Local Control Units for use by the Control Action Task software in determining the state of the accelerator. These signals are sensed by the Performance Monitoring and Evaluation System (PMES) and routed to the Accelerator Trigger Status Chassis for interfacing to the LCUs. This information allows control processes to evaluate the success of the shot and to take appropriate post-shot actions if necessary.

4.0 SOFTWARE ARCHITECTURE

4.1 Design Considerations

Understanding the design of the PBFA-II software system begins with the recognition that control system criteria were not available at the onset of the design phase; the designs of most machine elements were taking place in parallel with the control system design effort. Only general statements about the purpose of machine elements and estimates of equipment characteristics could be made initially. Because the accelerator is a research machine, criteria changed during both the development and operational phases. A software architecture that accommodates incremental criteria development and a phased implementation of control capability is critical to a successful project.

The software architecture, developed in response to the incomplete and changing controls criteria, isolates knowledge of equipment characteristics, operating sequences and states, and element interactions. Two primary sites where changes are likely to occur are identified within the software structure: a set of modules solely responsible for controlling the machine elements; and a set of modules responsible either for interfacing the control modules to the facility operators or for performing tasks involving the integrated control of several machine elements. The task of controlling machine elements is delegated to a single class of program modules that execute on the LCU called Control Action Tasks (CAT) [DOC:ACMP1 SEC:SBD, DOC:ACMP1 SEC:IORTD-3]. More specifically, Control Action Tasks

perform the following functions: scanning sensor inputs, setting control outputs, reporting input and output states to the ICD system, detecting and reporting alarm conditions, and carrying out standard control sequences in response to commands from the ICD system. Integrated control and operator interface functions are delegated to a class of program modules that execute on the ICD system called Transaction Processors (TP) [DOC:ICDS SEC:SBD]. Transaction Processors are the software agents that handle the four major classes of messages in the system: messages indicating a change in the facility state; messages specifying a change to be made in the facility state; messages from operators specifying a command; and messages to operators indicating the facility status. Each instance of a message represents a transaction request. Decision elements within a Transaction Processor detect a message and process it, sometimes initiating one or more transactions as a result.

A support layer is designed around the two classes of processes that provides three basic services: (1) communication of the current status of machine elements from Control Action Tasks to Transaction Processors; (2) display of status information and acquisition of operator commands; and, (3) storage and maintenance of facility configuration information and historical performance records. A set of distinguished interfaces exist between the support services and the applications modules that prescribe a standard for the exchange of information. The interfaces are implemented with a library of subroutines that create an effective barrier between applications modules and the system processes comprising the support layer. The intent of the support layer is to facilitate the design and phased integration of multiple Control Action Tasks and Transaction Processors.

The service layer and applications module classes are partitioned into five software subsystems: (1) communications between CATs and TPs are performed by the Host-Satellite Interface (HSI) subsystem [DOC:HSI SEC:SBD]; (2) operator interface tasks are performed by the Operator-Machine Interface (OMI) subsystem [DOC:OMI SEC:SBD]; (3) storage and maintenance of facility configuration records, historical performance records, and machine element parameter records are performed by the System Data Bases (SDB) [DOC:SDB SEC:SBD]; (4) providing a uniform structure for Control Action Tasks is performed by to the Control Action Task subsystem; and (5) Transaction Processors perform all integrated functions at the ICD. Each of these subsystems is discussed in subsequent sections.

While Control Action Tasks, Transaction Processors, and the support layer are necessary components of the system, they do not provide any of the major capabilities required to conduct operations. Additional higher-level software structures constructed of these basic components are implemented to provide this functionality. Four modes of

control are required. Figure 7 shows the multilevel approach to implementing the four control modes in which each level uses the services of the layer below it. The layers vary along a scale that apportions decision-making functions between the operators and the computer systems. At the lowest level is the Remote Manual Mode, which allows actuation of individual control points from the operations console. Point trending and status readout functions are also provided at this level. The next layer, the Unit Actions layer, provides automation of standard control sequences for individual machine elements. Each Unit Action produces a specific state in a machine element upon receipt of an operator command. Sequencing multiple machine elements in unison using Unit Action commands is performed by the Integrated Control layer. Integrated Control processes automate major tasks and take into account interactions among many machine element. The final layer, Automatic Staging, sequences entire shot procedures using Integrated Control functions and Unit Actions. This level is the most sophisticated, requiring both off-line planning processes and on-line coordination processes. Each level of control is discussed in subsequent sections.

4.2 The Host-Satellite Interface Subsystem

The Host-Satellite Interface (HSI) [DOC:HSI SEC:SBD] subsystem provides the communications service between Control Action Tasks and Transaction Processors. Specifically, the HSI is responsible for the following tasks: (1) providing a channel for communicating accelerator events to Transaction Processors [DOC:AFHSI SEC:SBD]; (2) providing a channel for communicating commands to Control Action Tasks from Transaction Processors [DOC:EFHSI SEC:SBD]; (3) providing a message buffering capability to accommodate multiple message streams from the LCUs and dynamic message loads arising during various stages of operation; and (4) providing a means to route specific messages or classes of messages to the particular Transaction Processors that require them. The HSI uses the DS/1000 Network System software to perform communications and the RTE-A Operating Systems' Class I/O facility for message queueing.

The HSI handles two classes of transactions: the Event Transaction and the Control Transaction. An event occurs when a significant change in the status of a machine element is detected by a Control Action Task. Events fall into one of six categories: Action, Alarm, Milestone, Group, Fault, and a general purpose category called Status. Six distinct message types, collectively called Satellite Reports [DOC:CMS SEC:IOPT-16, DSDIO-16], are used for communicating the event occurrences. These are: 1) Action Reports, 2) Alarm Reports, 3) Milestone Reports, 4) Group Reports, 5) Fault Reports, and 6) Status Reports. The information content of a Satellite Report is determined by the report type and the fields containing specific details of the event. Action Reports indicate either a change in the major

S. Y. Goldsmith
7521. 2/10/88
F/N - ARCHSYG

PBFA-II
Control/Monitor System

Layered Capability Architecture

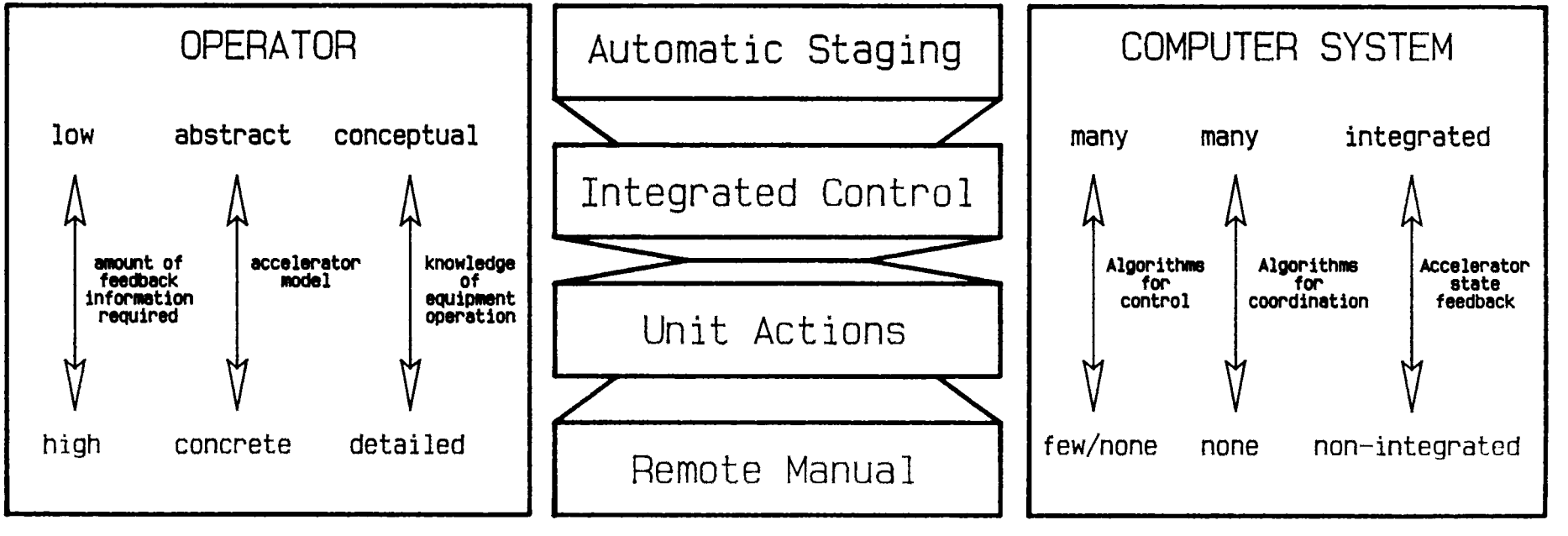

FIGURE 7 - LAYERED CAPABILITIES

state of a machine element or a change in the Primary Operating Mode of a machine element. Alarm Reports indicate that a serious condition or inadmissible state is present in a machine element. Milestone Reports indicate that a requested state or goal has been achieved. Group Reports indicate a change in the state of one or more sensor points belonging to a particular group of sensors. Fault Reports contain information on machine element malfunctions of a non-critical nature. Status Reports have a multitude of uses: everything not covered by the other five report types.

Event Transactions are usually initiated by Control Action Tasks. They may also be initiated by Transaction Processors. Figure 8 shows the processes and data structures involved in handling Event Transaction communications. To initiate an Event Transaction, a CAT calls one of six subroutines corresponding to the type of event. The details of the event are passed to the module through the formal parameters. The subroutine formats the report and sends it to a central event queue (EVTQ) that buffers reports from all Control Action Tasks executing on the LCU. An Event Link Master process [DOC:ACMPI SEC:IORTD-5] monitors the EVTQ to determine if any reports are present. If reports are detected in the queue, the Event Link Master reads the entire contents of the EVTQ and creates a single transmission packet containing a header and all reports. This packet is transmitted over the DS/1000 network to the ICD system. At the ICD system a dedicated process monitors each LCU link for the arrival of a transmission packet. These processes, called the Event Link Monitors [DOC:AFHSI SEC:IORTD1-6], send the received transmission packets to a central queue (ERSQ) where reports from all LCUs are merged into a single stream. A process called the Event Report Server [DOC:AFHSI SEC:IORTD-7] monitors the ERSQ for the presence of reports. The Event Report Server services the ERSQ according to a FIFO discipline. The task of the Event Report Server is to route the report at the head of the queue to Transaction Processors that use the report in decision processes. An event-to-process mapping table is maintained that identifies all Transaction Processors requiring a particular report [DOC:AFHSI SEC:IORTD-8, DOC:PBFAII SEC:PPT-4002]. The entries in the table depend on which Transaction Processors are active at the time. When a Transaction Processor becomes active, usually at bootup or upon operator request, it places the address of a private mailbox in the table entries for the reports it requires. When the Transaction Processor is commanded to become inactive, it removes the address from entries in the table. The mapping and unmapping task is performed by the Event Mapping Service process [DOC:AFHSI SEC:IORTD-9]. Recall that a report is uniquely identified by a general type and a specific code. This type and code is used to index the list of Transaction Processors that must receive the report. The Event Server distributes the current report to the private mailboxes of all Transaction Processors listed in the table for the

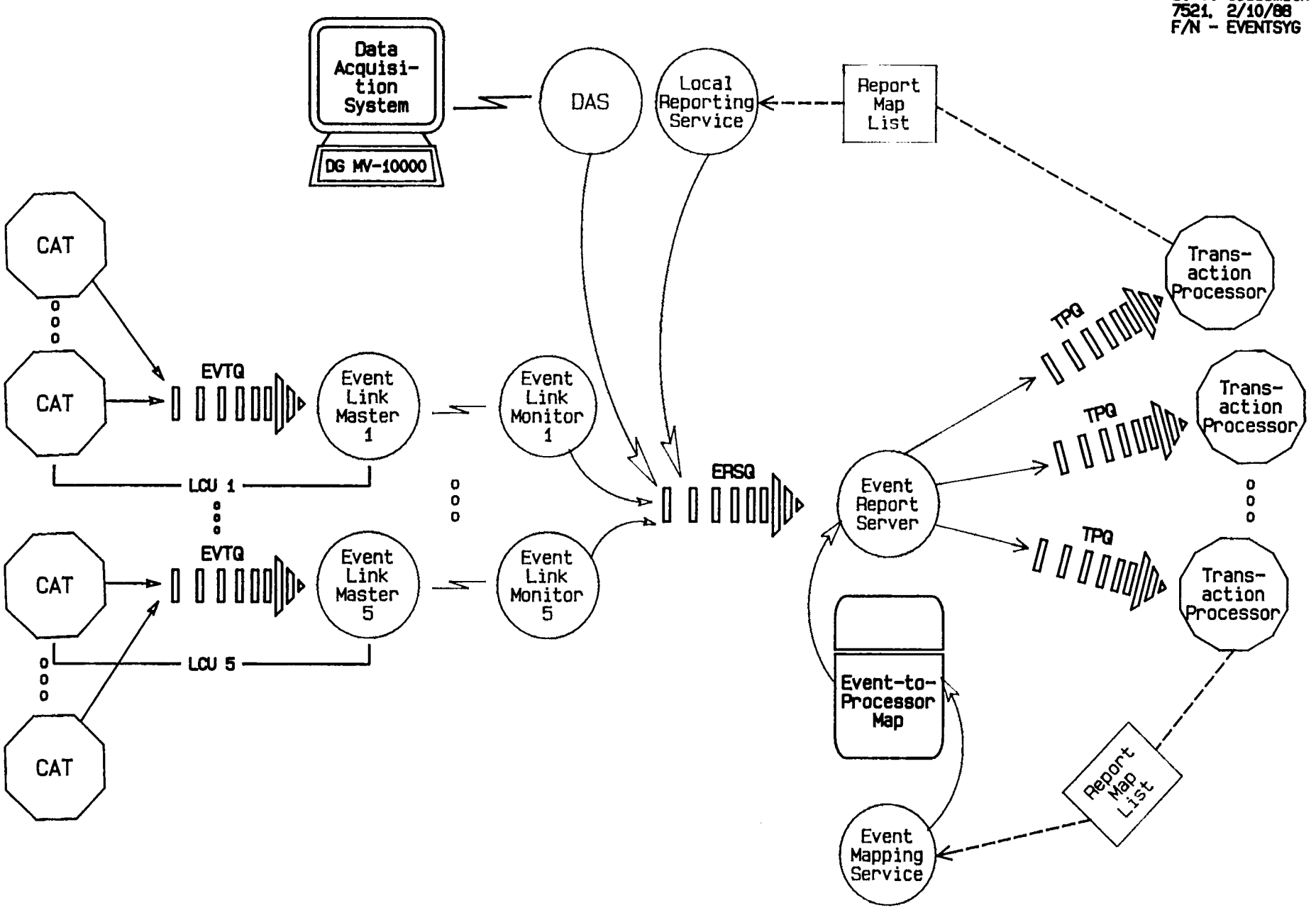

FIGURE 8 - EVENT TRANSACTION HANDLING BY THE HSI

report. When the Transaction Processors read their mailboxes and process the information in the report, the Event Transaction is complete.

Transaction Processors may also initiate Event Transactions. A typical application for TP-initiated reports are in integrated alarm monitoring processes. An integrated Alarm Report, identical in form and function to local alarm reports, may be issued by a Transaction Processor for use by other Transaction Processors, possibly in an operator display or automatic shutdown application. The Event Transaction service can provide a uniform methodology for TP-to-TP communications as well as CAT-to-TP communications [DOC:AFHSI SEC:IORTD-12]. The Transaction Processor reporting task is handled by the Local Reporting Service process. To initiate an Event Transaction, the TP calls one of six subroutines corresponding to the type of report in a manner analogous to CAT-initiated Event Transactions. The subroutine formats the report and schedules the Local Reporting Service process on the ICD system. The Local Reporting Service enters the report into the ERSQ, where it is merged with the main report stream and handled like any other report.

When Event Transactions must be handled from systems other than an LCU, a custom Event Link Monitor is designed. There is currently a custom monitor to interface the Data Acquisition System to the ICD system. The process monitors an asynchronous serial link for reports from the DAS, formats them, and merges them with the main report stream.

The second type of transaction handled by the HSI is the Control Transaction. Control Transactions occur when a Transaction Processor commands a state change to take place in some machine element, or otherwise attempts to communicate with a machine element. There are two types of control commands; the Unit Action Command and the Remote Manual Command. Figure 9 shows the processes and data structures involved in handling Control Transactions. To initiate a Control Transaction, a Transaction Processor calls a subroutine and passes the command parameters through the formal parameter set. The formal parameters include the name of the machine element and the desired values for machine element state variables. The subroutine formats the command and decodes the machine element name to obtain a network address for the CAT assigned to the machine element. The network address consists of the node number of the target LCU and the task name of the CAT. The command is issued by initiating a remote schedule of the CAT through the DS/1000 Remote Executive (REXEC) [DOC:EFHSI SEC:SBD]. When the CAT process executes as a result of this scheduling event, it retrieves the command buffer and processes it, completing the Control Transaction.

S. Y. Goldsmith, 7521
F/N CTHSI, 2/10/88
DIAGRAPH DISK07

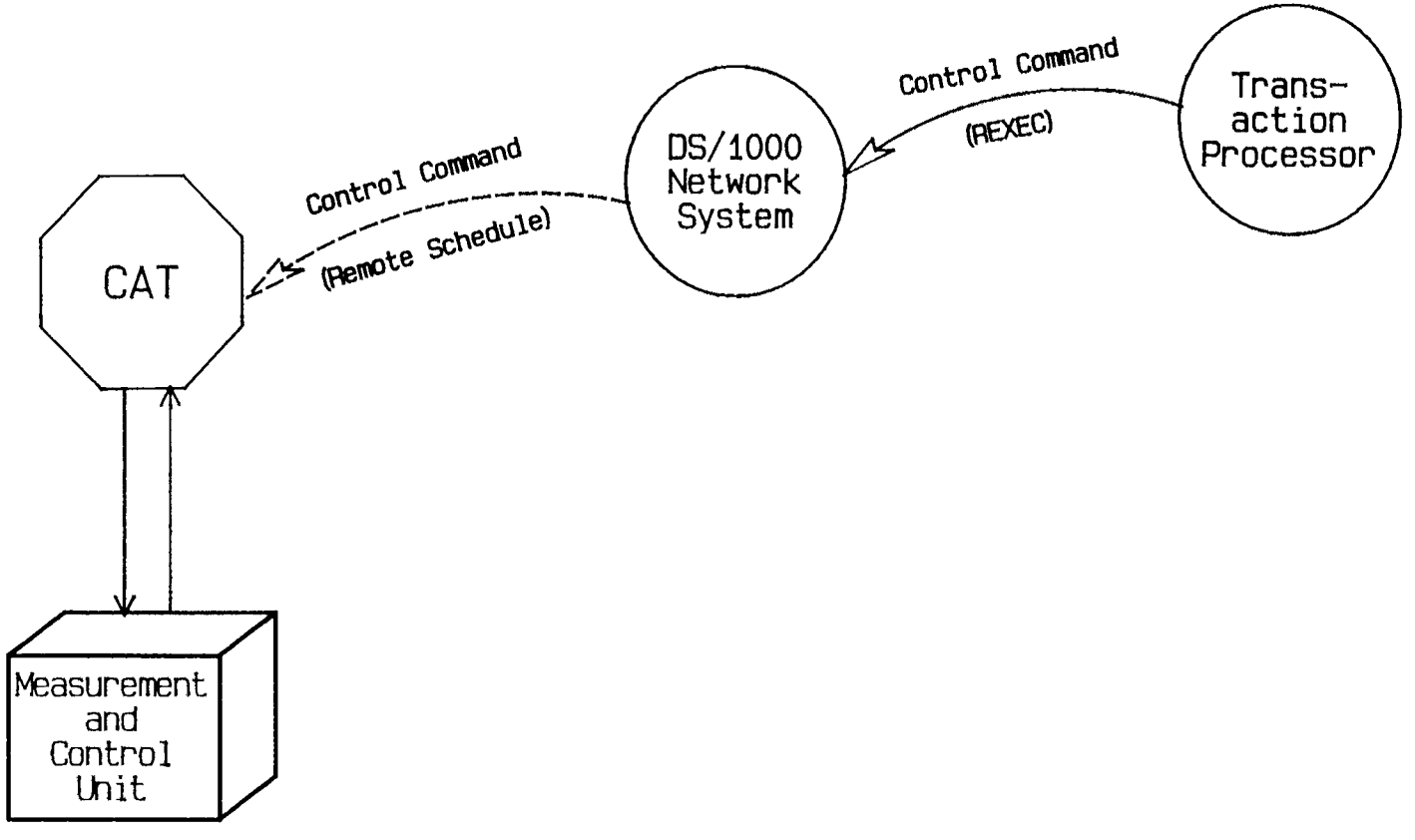

FIGURE 9 - CONTROL TRANSACTION HANDLING BY THE HSI

4.3 The Operator-Machine Interface

The Operator-Machine Interface handles all communications between the facility operators and the control system. The OMI has both an off-line and an on-line function. First, in the off-line role, the OMI subsystem provides a set of software tools to facilitate the development and integration of operator display panels. Second, in the on-line role, the OMI accepts and decodes commands from the operators and displays status and control information to the operators.

The OMI subsystem handles two classes of transactions: Command Transactions and Display Transactions [DOC:OMI SEC:SBD]. A command occurs whenever an operator touches an active area on a touch-sensitive display panel. Figure 10 shows the processes and data structures involved in handling Command Transactions. A touch-panel input device generates a simple message indicating the Cartesian coordinates of the touched area. This message is sent to the ICD system over an asychronous serial communications link. A single process at the ICD system called the Touch Panel Monitor monitors links from the seven touch input devices. When the Touch Panel Monitor [DOC:OMI SEC:IORTD-2] receives a touch message, it adds a Logical Unit Number (LUN) to the message that uniquely identifies which TP was the source of the message. This message, called a Touch Report, is sent to a central queue, GPMQ, and merged with the stream of pending Touch Reports. The GPMQ holds the input stream to the primary OMI process, the Graphics Panel Manager (GPM) [DOC:OMI SEC:IORTD-1].

The Graphics Panel Manager's primary function is to decode a sequence of Touch Reports into a command code. A strict grammar is enforced by the GPM, effectively leading operators through the sequence of touches comprising a command by providing visual feedback on the possible options after each touch. This function is provided by a complex state machine and input language grammar described in SAND86-0643. For the purposes of this report, it is sufficient to assume a Touch Report uniquely specifies a Command Transaction.

The Graphics Panel Manager reads the Touch Report at the head of GPMQ and decodes the report information. The X and Y coordinates, together with the LUN specify an index into a table containing the Command Transaction code. This table, called the Logical Button Table, holds the Command Transaction codes for the seven currently visible displays. The code is sent in a message, the Command Report, to the mailbox of the Transaction Processor handling the particular display. The Command Transaction is complete when the Transaction Processor performs the specified action.

Every display panel is managed by a dedicated Transaction Processor called a Display Monitor [DOC:OMI SEC:IORTD-3]. The function of the Display Monitor is to implement the operator commands provided by the display panel and to keep

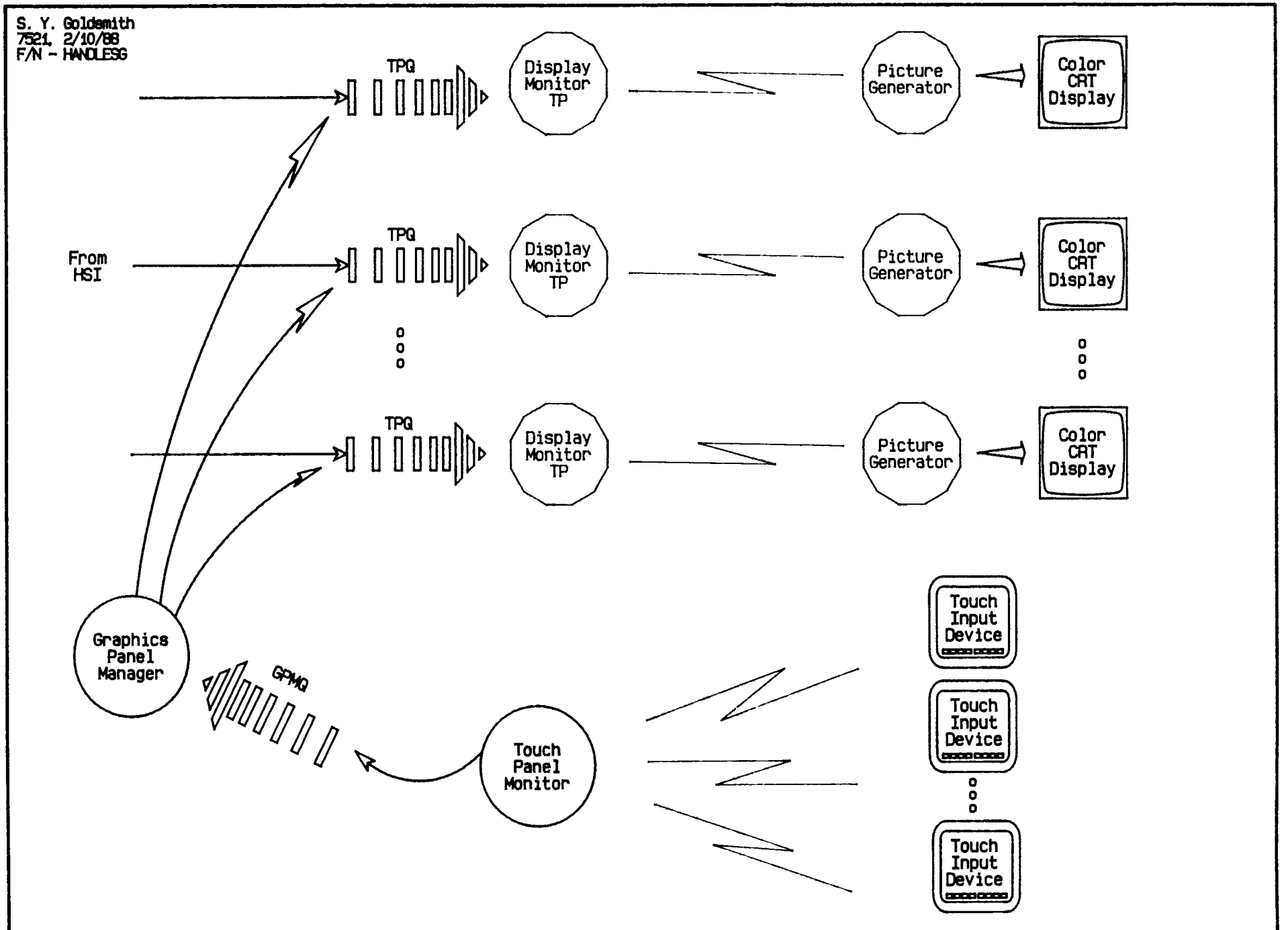

FIGURE 10 - COMMAND AND DISPLAY TRANSACTION HANDLING BY THE OMI

the picture updated with the proper status information. Facility status information is obtained by the Display Monitor through the Host-Satellite Interface Subsystem, via the specific Satellite Reports mapped to it in the Event-to-Process Map Table. Command Reports and Satellite Reports are placed in the same mailbox. The Display Monitor is a custom process designed specifically for each display panel. It contains display specific information such as button-to-command mappings, the particular picture sub-elements making up the picture, and the required Satellite Reports needed to maintain a current picture.

The task of drawing and redrawing the actual picture is allocated to a co-process of the Display Monitor called the Picture Generator. Picture Generators execute on the Console Graphics Processors (CGP) [DOC:OMI SEC: IORTD-4]. The function of the Picture Generator is to process Display Transactions initiated at the ICD system and sent over the DS/1000 network to the CGP. A Display Transaction may be initiated by either the Display Monitor or by the Graphics Panel Manager. Only transactions involving a change in button displays are initiated by the GPM. The purpose of these changes is to implement the proper grammar and lead the operator through the proper command options. Display Monitors initiate changes to the various status display elements upon receiving the information in a Satellite Report. When the PG receives a Display Transaction from a Display Monitor, it deletes the old information and makes the necessary changes to the visible picture thereby completing the Display Transaction.

4.4 The System Data Bases

The System Data Bases (SDB) provide records maintenance services for two types of information: (1) information describing the current configuration of the facility, and (2) the results of performance tests and historical records of experiments. Four data base subsystems comprise the SDB subsystem. Two data bases contain the facility configuration: the POINTS data base, a disk-resident data base that contains records describing the current sensor and actuator point inventory [DOC:OLDBS SEC:IORTD-1]; and, the MACH_ELEMENTS data base, a disk-resident data base that contains records describing the machine elements currently installed, the Control Action Tasks that control them, and the parameters constraining their operation [DOC:OLDBS SEC:IORTD-2]. The third data base, the SHOT data base, contains records describing the performance of machine elements during experiments and the operating points specified for the experiment [DOC:OLDBS SEC:IORTD-7]. A fourth data base, HGTAB, is a mixture of records from the other three data bases that are used in on-line operations by decision and display processes executing on the ICD system [DOC:RTDBS SEC:IORTD-6].

The POINTS data base is implemented with the HP IMAGE/1000 sequential data base system. Figure 11 shows the basic schema for POINTS. There are three master data sets: (1) the Facility Point Name master set; (2) the POINT ADDRESS master set; and (3) the NODE master set. The master data sets are linked to a single detail set, the POINT ATTRIBUTES detail set.

Every sensor or actuator point in the control system is described by a POINT ATTRIBUTES detail record. The detail record contains the following items: the point-type code, the point name and global identifier code, the point descriptor tag, the engineering units code, the conversion coefficients and gain (analog-type points only), and the node number of the LCU interfaced to the point.

Each point has two unique keys associated with it. The point name is a 10-character string formally called the Facility Point Name (FPN). The FPN is used in all references made by operators to identify the point. These references occur in data base maintenance operations, on-line status displays, on-line control operations, and historical trend readouts. The second key, called the Global Point Number (GPN), is used by system processes to access point attributes records directly during on-line operations. Either key will uniquely specify the point, but the GPN allows direct indexing into the HGTAB data records describing the point to obtain faster handling of large numbers of point transactions. An additional 24-character string called the POINT DESCRIPTOR TAG allows a more verbose description of the point for use in operator displays, historical trend printouts, and point inventory listings.

The disk-resident point attributes records are accessed through the three master data sets. The FPN master data set allows access to the Point Attributes Detail set with an FPN key. The FPN is hashed into the GPN when accessing the record. The GPN represents the actual record number for the detail set in the IMAGE/1000 data base. FPN master data set records are created when the point is first entered into the system. Point attributes records (and FPN master records) are created, deleted, and modified by a single interactive program, the Point Management program (PMM). Through this program, operators maintain the point inventory for the facility. Before a new point can be added to the POINTS data base, the point address of the point must be in the POINT ADDRESS master data set. A configuration process for each LCU is performed prior to defining points and specifying their attributes. This process performs three steps: 1) it interrogates the I/O multiplexor on the LCU and determines the configuration in terms of board type and slot number; 2) it constructs a record consisting of point address and type and enters the record into the POINT ADDRESS master data set; and 3) it enters the board type and slot array into the NODE master data set. The POINT ADDRESS

S. Y. Goldsmith, 7521
F/N - PTSDBSYS, 2/10/88
DIAGRAPH DISK07

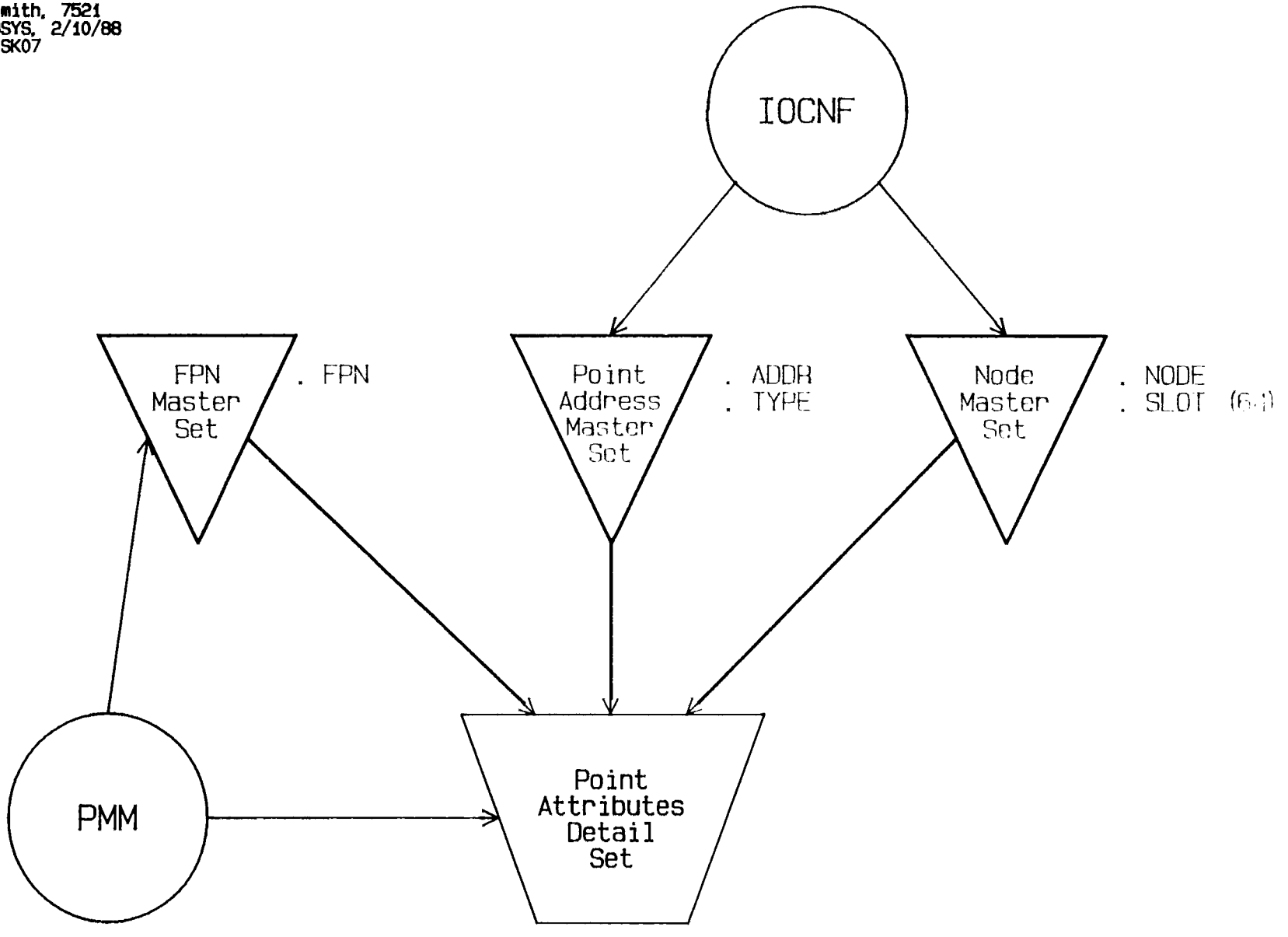

FIGURE 11 - POINTS DATA BASE SYSTEM

Master specifies the pool of points available for assignment and prevents the definition of a point at an illegal or non-existent address. The NODE master data set allows access to all points residing on an LCU through a NODE key. This is useful in producing inventory listings of defined points. As mentioned previously, it also stores the hardware configuration of the I/O multiplexor for the LCU.

A fourth master set, the GROUP master set, is also in the POINTS data base although it is not linked directly to the POINT ATTRIBUTES detail set. The GROUP master set is used to manage the point group feature of the Remote Manual Mode capability. The data set contains a 32-character field to hold the Group Name, a Group Number field, and a Group File field. A group is a collection of points specified by a Group File listing the Facility Point Name of each point. The group has two unique keys associated with it: the Group Name and the Group Number. The Group Number is the actual record number of the entry in the GROUP master set. It is derived from the Group Name through the IMAGE/1000 hashing algorithm. The Group File is created using the standard ASCII editor. The first record contains the Group Name. The remainder of the file is the list of FPNs comprising the group. When the file has been created it is submitted to the Group Builder process. Group Builder is a user-interactive process that allows operators to manage the creation and deletion of point groups. When invoked in the create mode, the Group Builder process enters a group record into the GROUP master set. The Group Number is assigned at this time. When invoked in the delete mode, the Group Builder process removes the record for the group from the GROUP master set. The point groups serve as the basis for many important features of the Remote Manual Mode capability and are discussed in more detail in Section 4.7.

The MACH_ELEMENTS data base system is comprised of a hierarchical data base and several processes as shown in Figure 12. The top-level directory, called MACH_ELEMENTS, holds an Element Master File and subdirectory files for every machine element in the system.

The Element Master File [DOC:RTDBS SEC:PPD-703] records the current facility configuration. It contains a record for each machine element indicating the machine element name, the node number of the LCU that interfaces it, the task name of a CAT, and a code that determines whether the machine element is active in the current configuration. During the initialization process for an LCU the Element Master File is accessed. A field for each machine element record indicating the node number of the LCU is interrogated. The records for all machine elements controlled by that LCU (node) are read. For every machine element whose Initialize Code is set, the LCU initialization process initializes and starts the corresponding Control Action Task residing in memory on the LCU. By setting or resetting the Initialize Code and restarting both the LCU and the HGTAB initialization process, a machine element may be added or

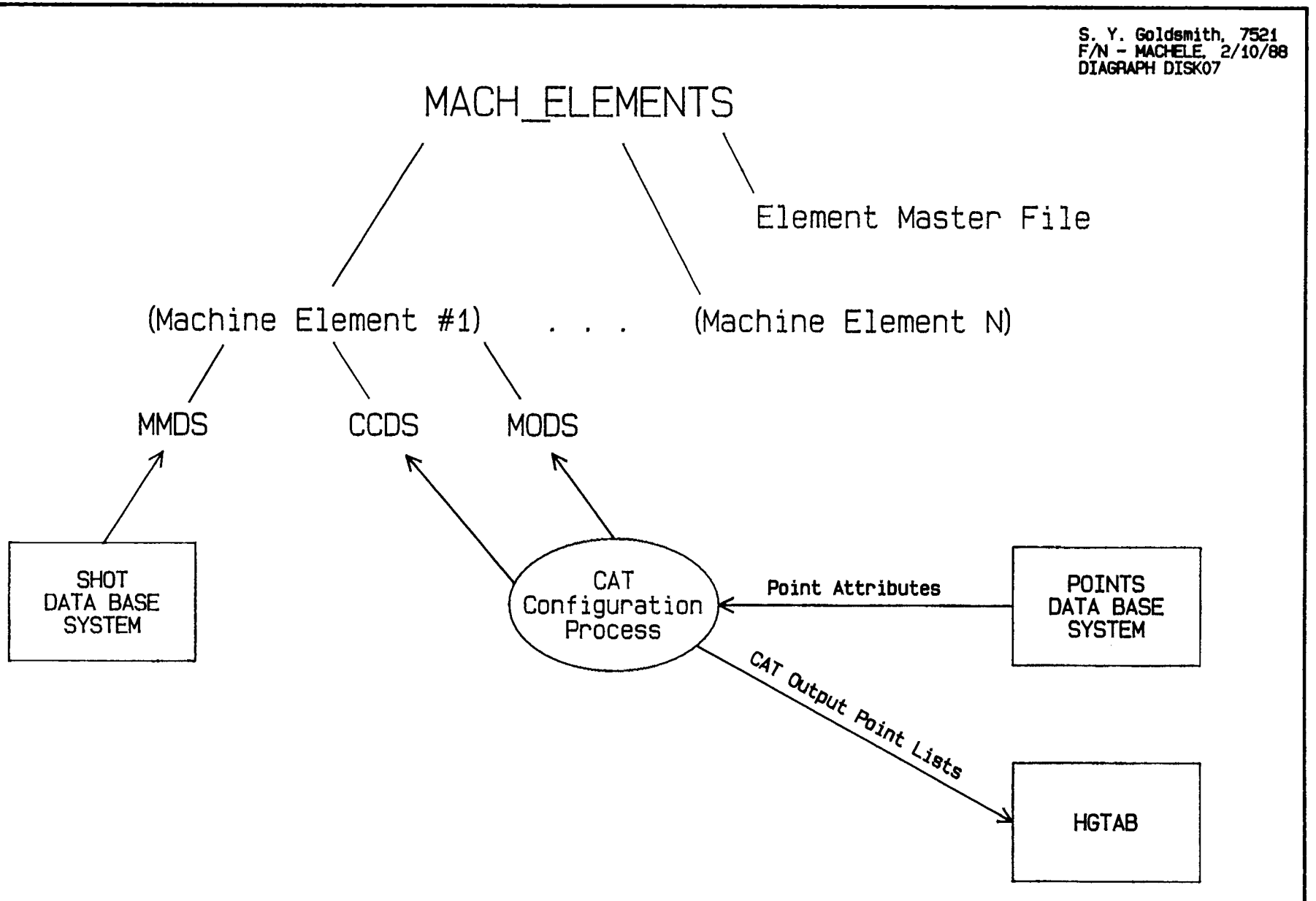

FIGURE 12 - MACHINE ELEMENTS DATA BASE SYSTEM

removed from the configuration. The machine element is uniquely identified by two keys: (1) the Machine Element Name and the Global Action Number (GAN). The GAN is the record number of the Machine Element File entry. The Machine Element Name is used by operators and system engineers to reference the machine element in user-interactive processes. The Global Action Number is used in on-line Control and Event Transactions to directly index information on the machine element rapidly.

Every machine element has a private directory that contains four files: 1) the Machine Model Data Set (MMDS); 2) the CAT Configuration Data Set (CCDS); 3) the Manual Output Data Set (MODS); and 4) an SOSI Configuration File. The CCDS and MODS files are used in conjunction with the Unit Action Mode capability and the Remote Manual Mode capability, respectively. The MMDS file and the SOSI Configuration File are used by Shot Parameter Configuration System (SPCS) described later in this section.

The Machine Model Data Set contains a parameter set describing the operational characteristics of the machine element. The parameters are the modeling constants of the implicit model used by the Control Action Task to control the machine element properly. Typical parameters are tolerances on analog settings, delay values, charging rates, and physical constants. The MMDS parameters do not change from shot to shot unless the machine element performance characteristics change.

The CAT Configuration Data Set [DOC:RTDBS SEC:PPD-704] contains records that provide information on the sensor and actuator points used in the Control Action Task. The file is derived from records in the POINTS data base by the CAT Configuration Process. When the CAT is started from a dormant state, it undergoes an initialization process whereby the CCDS file is read from MACH_ELEMENTS on the ICD system and passed to the CAT. The CAT configures its I/O tables with the information.

The Manual Output Point Data Set contains records describing the system addresses for all output points used by a CAT. The sole function of the file is to configure a portion of the HGTAB data base that indicates the physical addresses of output points for use in Remote Manual operations. The table decodes a Global Point Number to a Local Point Number and Control Action Task name, which comprises an address for the point.

The Shot Parameter Configuration Subsystem (SPCS) [DOC:SPSS SEC:SBD] is that part of the shot data base responsible for maintaining sets of parameter values for various types of shots. The processes and data structures comprising the SPCS are shown in Figure 13. To specify a shot, an operator initiates an interactive session with the SPCS subsystem through a support terminal. An interface process, the SOCI, provides menu-driven procedures for

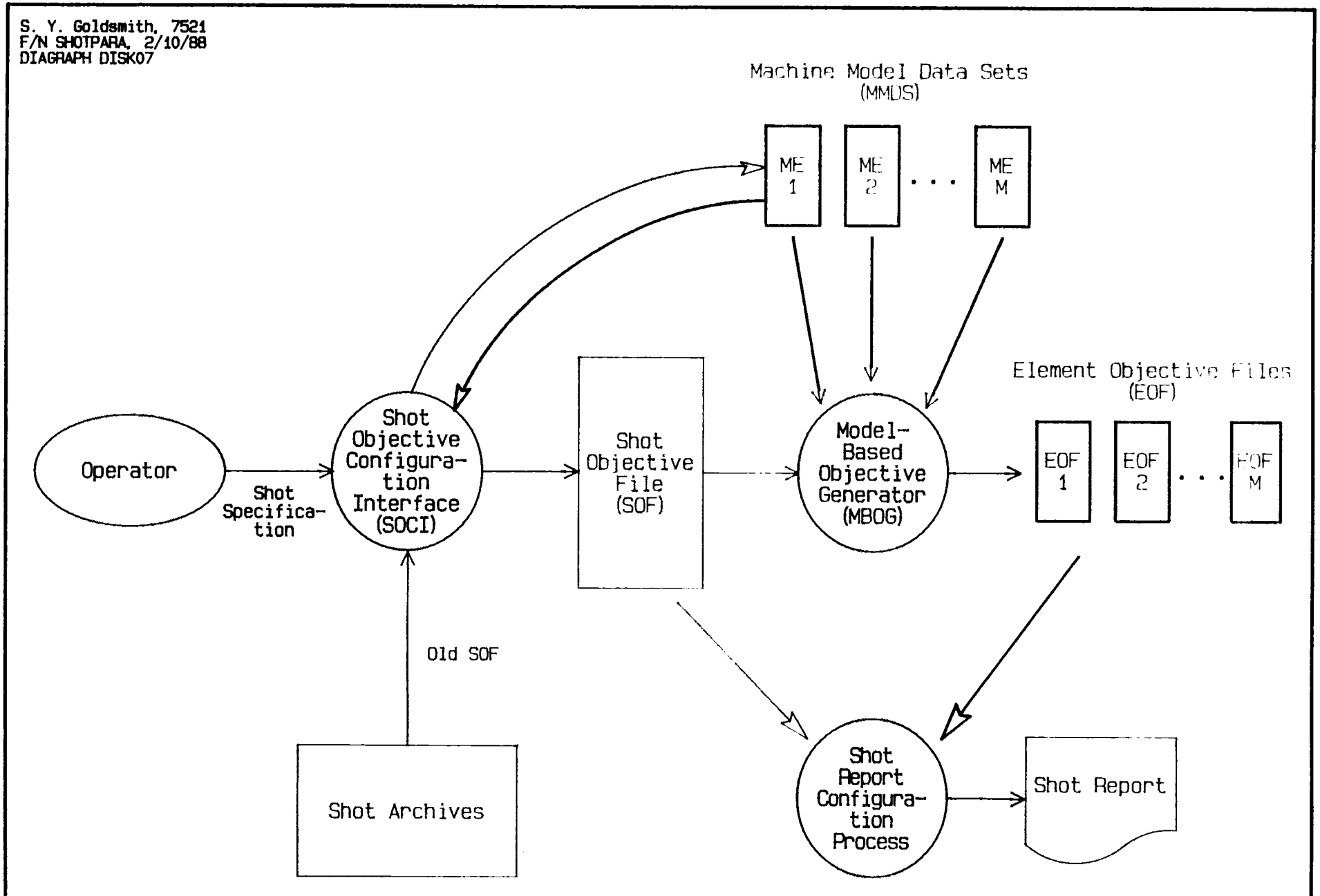

FIGURE 13 - SHOT PARAMETER CONFIGURATION SUBSYSTEM

specifying a shot [DOC:SPSS SEC:IORTD-2]. The operator indicates one of six standard shot types and is then prompted for parameters relevant to that shot type. The shot types currently implemented are: 1) Marx Trigger Generator Test; 2) Laser Trigger Test; 3) Marx Dummy Load Test; 4) Gas Switch Dummy Load Test; 5) Applied-B-Diode Test; and 6) Blade Load Test. The shot-type parameters and header information, consisting of operator name, shot number, shot description test, are stored in the Shot Objective File (SOF). The SOF represents a high-level description of the goals of the shot. The SOF file is then submitted to a process that produces a more detailed specification of the shot in terms of parameter settings for each machine element. The process, called the Model-Based Objective Generator (MBOG), uses information from the SOF and from the Machine Model Data Sets to produce a complete and consistent set of parameters and model constants for each machine element [DOC:SPSS SEC:SBD]. The MBOG is not implemented with full capability at this writing because the functions performed by the MBOG requires expert-level knowledge of accelerator subsystems and operations. Instead, the operators specify all parameters for the machine elements involved in the shot through the SOCI. The MBOG simply reads the MMDS files and appends the model constants to the target parameters for each machine element. The Element Objective File (EOF) is created for each machine element by the MBOG and contains all constants and parameters for the element. A complex procedure may require several Element Objective Files, representing the sequence of states the machine element must pass through in the course of the shot. The entire set of Element Objective Files constitute a precise specification of the accelerator state at various stages of the shot.

Because PBFA-II is an experimental facility, it is necessary to control the parameter settings carefully to insure repeatable and consistent results. When configuring a shot through the SOCI, the operator may specify a previously-conducted shot be used as the source of parameters for the pending shot. The Shot Objective File corresponding to the specified shot number is read into the SOCI and used as a template for the current shot. Because the shot specification process occurs off-line, several shots may be specified during one interface session. The SPCS will queue the shots in the order entered, allowing several shots to be taken in succession.

The Shot Objective Configuration Interface has a privileged mode that allows viewing and updating of the Machine Model Data Sets for each machine element. The SOCI is the sole process that may update the MMDS files. The MMDS files are changed only when the model constants no longer reflect the actual performance of the machine element. MMDS files are updated when calibration and test procedures indicate a change in operating characteristics.

Data on machine element performance taken during experiments is archived in the SHOT data base [DOC:RTDBS SEC:IORTD-7]. The SHOT data base is a hierarchical data base implemented with the HP FMP file manager. It consists of a series of directory structures, one for each shot taken on the facility. The directory structures are created by the SPCS when the shot parameters are specified. Shots are identified by the shot number encoded as the last four characters of the global directory name. The shot global directory contains one or more Event Files and the Shot Objective File (SOF) discussed previously.

Event Files provide the means to organize a shot into a set of discrete sequential events. The motivation for this structure is to allow multiple, unsuccessful attempts at taking the shot to be recorded separately for ease of analysis later on. Each aborted attempt that failed to achieve the shot objectives is recorded under a distinct Event Reference Number (ERN). Events are identified by the ERN encoded as the last three characters of the subdirectory name. The ERN directory contains subdirectory files for every machine element that participated in the shot. In the machine element's directory are three files: (1) the Element Objective File; (2) the Machine Model Data Set; and (3) the Performance Record Data Set. The Element Objective File and MMDS files were discussed previously. Their role as part of the shot archive is to record the intended parameter settings. The actual settings attained by the machine element during a shot are recorded in the Performance Record Data Sets. Every sensor and actuator point that changed state during the shot is represented by a file containing a time record of the point values. The files are identified by the FPN of the point and the extent ".PRDS".

Point trend data is recorded by a subsystem of the shot data base called the Point Data Recording subsystem (PDR) [DOC:ARC00 SEC:SBD]. The process and data structures comprising the PDR subsystem are shown in Figure 14. A Control Action Task in the recording mode sends a copy of the current point state vector to a central queue (RECQ) on each scan. The queue is used by all CATs executing on the Local Control Unit. A single process, the Point Recording Process, reads each point state buffer from the queue and stores it in a central memory file [DOC:RTREC-01 SEC:PPD-1]. The memory file is organized into records, one for each Control Action Task. Each record actually operates as a circular file, overwriting the oldest data buffers when full.

A low priority archiving process periodically removes buffers from the central partition and sends them to the ICD system via the DS/1000 network [DOC:RTREC-01 SEC:PPD-2]. These buffers become part of the Performance Record Data Set (PRDS) for the current shot. Management of the PRDS files is performed by special monitor processes called Master Archival Processes (MAP) [DOC:ARC00 SEC:SBD]. The Master

S. Y. Goldsmith, 7521
F/N - RECSUBSY, 2/10/88
DIAGRAPH DISK07

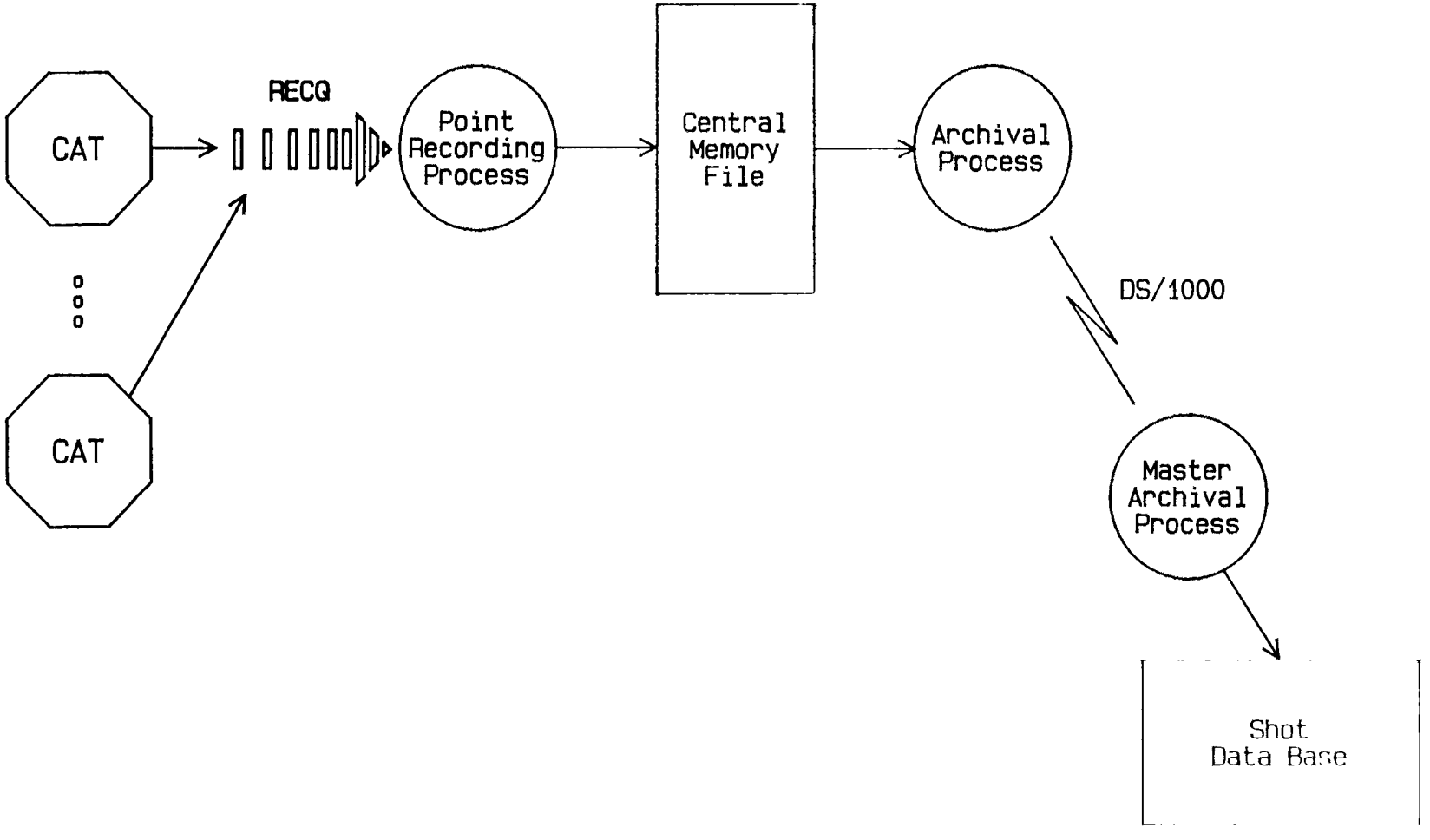

FIGURE 14 - POINT DATA RECORDING SUBSYSTEM

Archival Processes receive point buffers from the archival processes on the LCUs over the DS/1000 network. The MAP breaks the buffer into individual point states and records it in the proper PRDS under the proper Machine Element directory.

Portions of the system data bases required to perform transactions in real-time are stored in a memory-resident data base, HGTAB. HGTAB records reside in a global memory partition. When the ICD system is booted up, a process reads selected records from the disk-resident data bases and copies them into the HGTAB records. In addition to the static records, HGTAB also contains records that hold the current state of each machine element. The state is described in terms of the point values, alarm status, and the state of machine elements. Information contained in the Satellite Reports is stored and updated in real time by various reporting services. The reporting services consist of the HSI subsystem and a Transaction Processor for each Satellite Report type. The TP handling a report type maps into all report codes in the Event-to-Process map table in the HSI and, consequently, receives all instances of that report type. The reports are logged in the memory-resident tables for use by other TPs in decision or display tasks (see Figures 16 and 17).

4.5 The Control Action Tasks

The control of individual machine elements is accomplished by programs called Control Action Tasks (CAT). A generic program skeleton provides the framework for each custom CAT [DOC:ACMPI SEC:PPD-303]. The skeleton code performs a set of standard services: mode switching, the major loop cycle, process point scanning and output, communications with Transaction Processors, recording of process point states, decoding and dispatching of generic commands, and program initialization.

A machine element may be controlled in one of three Primary Operating Modes: 1) Manual Mode; 2) Local Mode; and 3) Automatic Mode. A fourth mode, Off-Line Mode, shuts down the machine element control system altogether. In Manual Mode, the CAT accepts commands from a Transaction Processor to change the states of individual control points. The CAT simply outputs the requested values without utilizing knowledge of the machine element. The Local Mode allows monitoring the state of the machine element, but disables all control points. Monitoring is accomplished through the Group Reporting mechanism discussed in Section 4.7. Actual control is accomplished by operators through the hardware-based manual control chassis for the machine element. The Automatic Mode accepts high-level commands specific to the particular machine element being controlled. These commands are processed by a custom-control subroutine developed specifically for the machine element. In the Off-Line Mode, all control points are disabled and the CAT is set dormant waiting for initialization.

Control Action Tasks operate on a major loop cycle. The cycle is implemented by time-listing the program in the dispatch list of the RTE-A operating system. A separate entry point is used for each control mode. The basic set of operations performed on each cycle is as follows: 1) check for commands from a Transaction Processor; 2) scan all input points; 3) record all input states; 4) perform custom-control calculations; and 5) set all new output states. Step 4 is performed only in the Automatic Mode.

The structure of the Control Action Task allows efficient phasing of control capabilities without requiring substantial recoding of the program. Local and Manual Mode control do not incorporate any knowledge of correct operating sequences and common control actions. In these modes, all that is required to achieve effective manual control is the Point Attributes records and MACH_ELEMENT data base entries. The point description tables in the CAT are derived directly from the System data base records during initialization of the CAT. Manual mode is used in conjunction with the Remote Manual Mode capability.

In the Automatic Mode, control sequences are implemented in a single subroutine that implements a Finite State Machine (FSM) model of the machine element operation. The FSM provides the framework for all Automatic Mode commands. The FSM will reject illegal commands, detect alarm states, perform control sequences while monitoring, and communicate important states of the ICD system through the HSI. Each FSM is a custom-design based on the characteristics and desired operating states of the machine element.

4.6 The Transaction Processors

Transaction Processors are programs that perform integrated control and display functions. Transaction Processors also handle generic system tasks such as real-time data base update and data archiving. Any program that handles one or more of the four major classes of transactions is considered a Transaction Processor.

There is very little generic structure to a Transaction Processor. Each TP is usually a custom application. TPs perform different roles in the different software layers. There are two general characteristics of Transaction Processors, however: an interface to the HSI and an interface to the HGTAB data base. By interfacing to the HSI, a TP can monitor operator commands as well as messages from individual Control Action Tasks. Because the TP is in suspension while awaiting messages, it provides an effective real-time response mechanism for high-level conditions, without impacting the average CPU utilization. Access to the HGTAB data base allows a TP to initialize itself with the current state of the accelerator when scheduled to begin executing. Examples of Transaction Processors functions are discussed in the following sections.

4.7 The Remote Manual Capability

The Remote Manual Capability allows operators to interact with individual machine elements on a point-by-point basis. The Remote Manual Control mode essentially duplicates the control functions available through the local hardware chassis for the machine element at the operations. This capability is important for phased implementation of machine element control capabilities. During the installation and checkout phase, machine elements are controlled manually to verify wiring and operational features. Only when testing is complete and operating sequences have been established though Remote Manual means, can automation of the operating sequences begin. Remote Manual is also useful for troubleshooting machine elements. Remote manual operation, in conjunction with the point recording system, enables characterization experiments to be performed that provide a detailed picture of element operation.

The Remote Manual Capability is implemented with the standard support services. A display panel is designed for each machine element or each manual control task (a task may involve many machine elements). This involves writing the two custom programs required by the OMI: the Display Monitor and the Picture Generator. With the exception of the CAT, the remainder of the processes are generic. Figure 15 shows the processes and data structures comprising the Remote Manual Capability. There are two primary transactions that the Remote Manual (RM) system is concerned with: the RM Command and the Group Report. To initiate an RM action, the operator sets each output point on the display screen to the desired value. The entire batch of point values is then sent to the Control Action Task, which validates the request and sets the proper output point values. The RM command is acknowledged with an Action Report to the initiating display monitor.

The Group Report Transaction occurs when a point in a particular group changes state. The assignment of points to groups is done at the SDB by naming the group and listing all FPNs in a file as discussed previously in Section 4.3. The cohesion of a group is arbitrary and may be based on machine element, task, or some special-purpose monitoring requirement. The display monitor displays the state of every point in the group. Changes in state are reported on an exception basis by the Control Action Task when a change is detected in a point value. The point value, along with its Group ID number, is sent in a Group Report to the HSI. All Group Reports are intercepted by a Transaction Processor called the Point Value Data Base Monitor (PVDBM). The PVDBM maintains a table of current point values for all points in the system in the HGTAB data base partition.

Every Group has a unique identification number called the Group Flag. The Group Flag is sent to the HSI in the Group

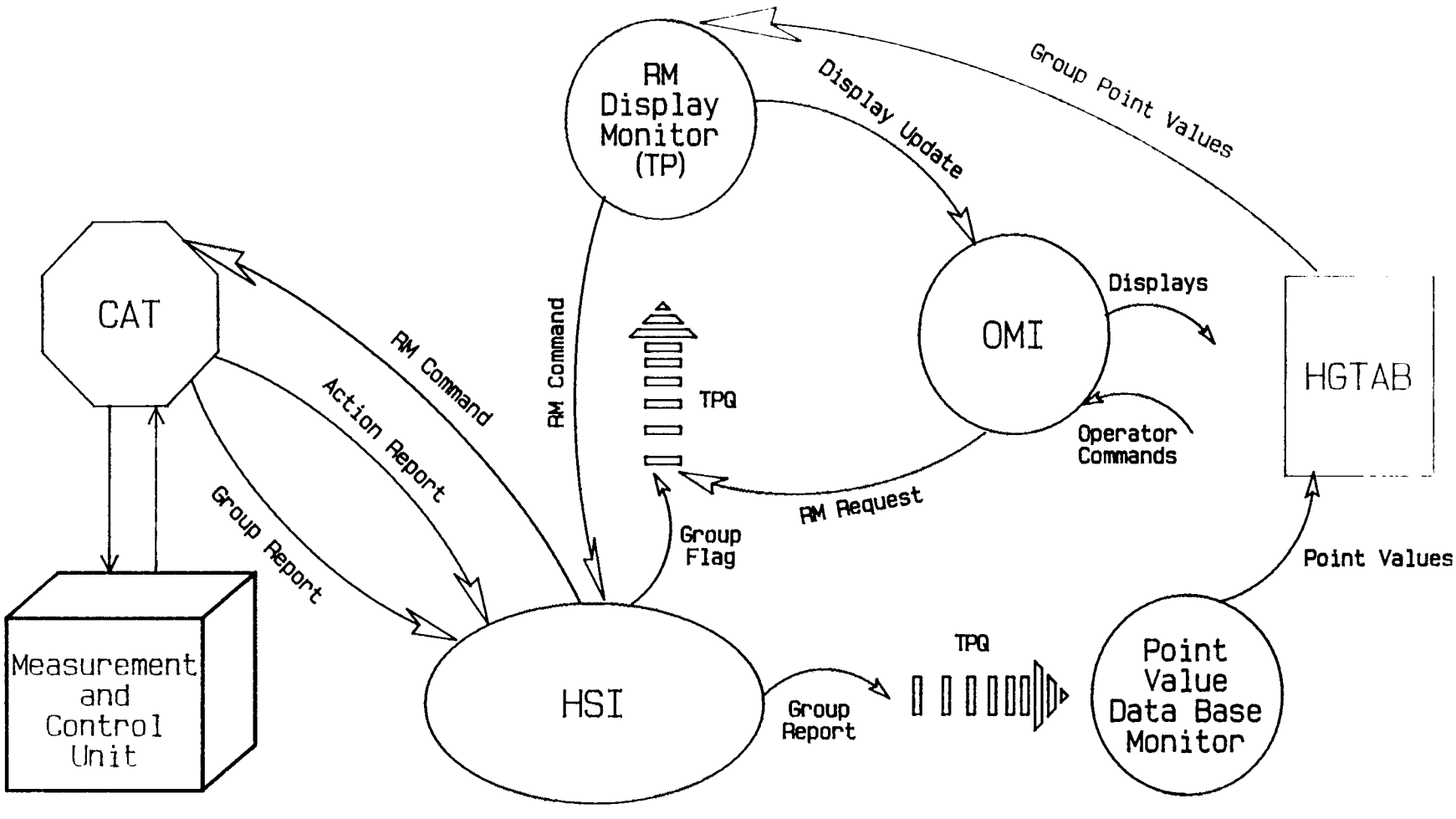

FIGURE 15 - REMOTE MANUAL CAPABILITY

Report to indicate that a point in the Group has changed its state. Any Transaction Processor that is mapped to the Group Report will receive the Group Flag, causing the Transaction Processor to check the point value tables in HGTAB for the latest states of all points in the Group.

4.8 The Unit Actions Capability

The Unit Actions Capability provides high-level control of machine elements from the Operations Console. A Unit Action is an entire control sequence that is carried out on a machine element in response to a single operator command. Each machine element has several possible Unit Actions. Each Unit Action is designed to produce a specific state in the machine element. Through the use of Unit Actions, operators may conduct shot procedures involving many machine elements, without the cognitive burden of point-by-point control and monitoring in the Remote Manual mode.

The Unit Action Capability is based on the Automatic Mode of operation in the CAT system. The custom-control kernel in the CAT implements a Finite State Machine that responds to Unit Action commands to produce the desired state. The processes comprising the Unit Action Capability is shown in Figure 16.

The Unit Action system is concerned with four types of transactions: the Unit Action command, the Action Report, the Alarm Report, and the Milestone Report. To issue a Unit Action command, the operator invokes the command at a display panel. The command is sent to the Display Monitor where it is decoded and sent to the HSI. The HSI transmits the command to the proper CAT, which then validates and executes the command. When the commanded sequence has been initiated, the CAT sends an Action Report to acknowledge the command. The Action Report is received by the HSI and routed to the proper Transaction Monitors, including the Display Monitor that initiated the command. The Control Action Tasks will perform the requested Unit Action sequence provided the command is feasible in the current machine element state. Once the sequence is initiated, the operator receives information on the status of the procedure through Milestone Reports and Alarm Reports. Milestone Reports indicate the machine element has reached some significant intermediate state. Alarm Reports indicate some inadmissible state or malfunction that prevents the machine element from reaching the requested state. These reports are sent to the Display Monitor where the information is formatted and displayed. Current Milestone states and Alarms are also maintained in the HGTAB data base by on-line monitors. This allows Display Monitors to be removed from visible status and restarted again with current information. This capability is crucial to concurrent monitoring of multiple tasks by the operators. When a CAT has successfully completed a Unit Action sequence, it issues an

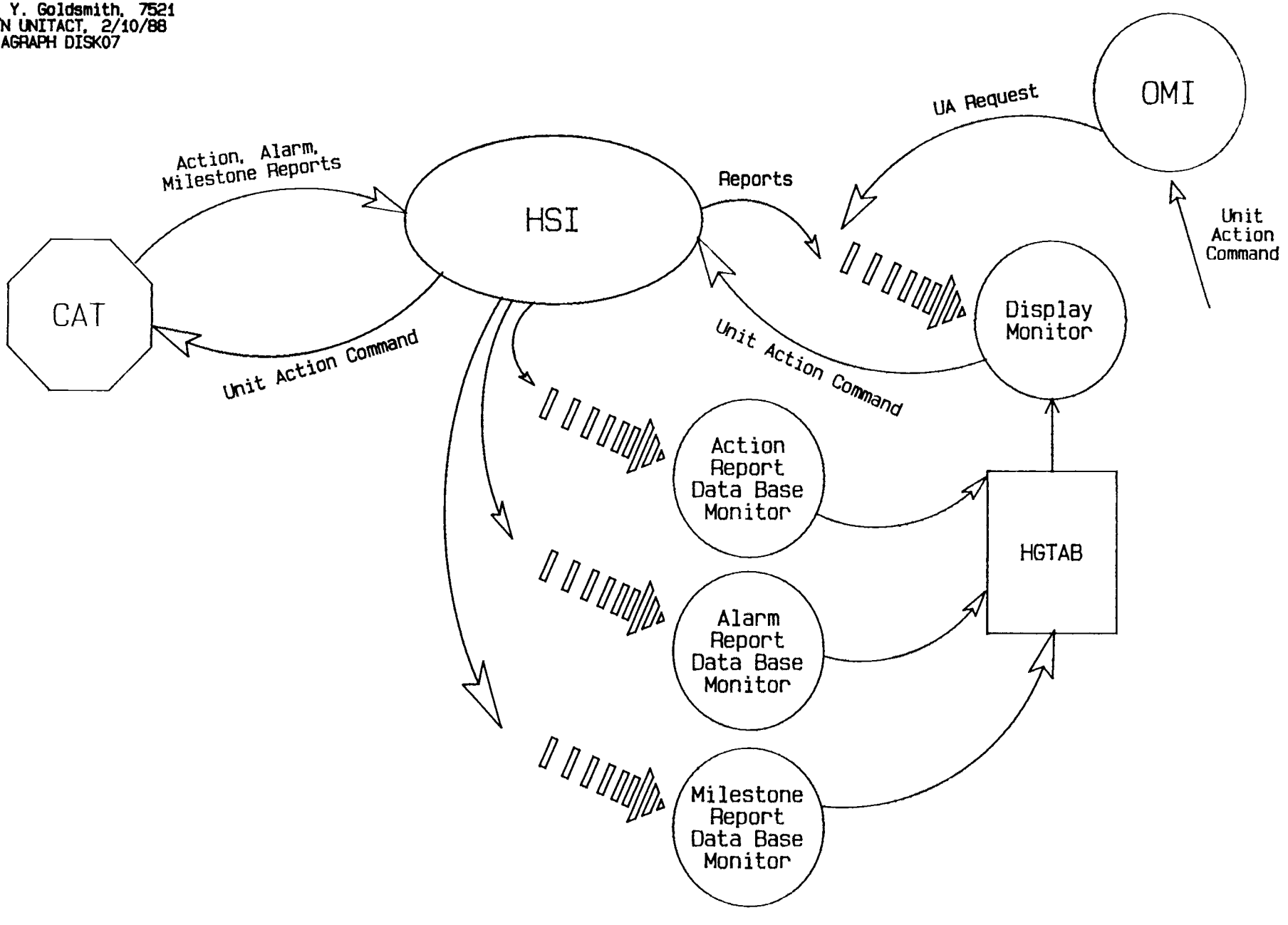

FIGURE 16 - UNIT ACTION CAPABILITY

Action Report to the Display Monitor indicating that the goal state has been reached and maintains the state until another command is received requesting a new state.

4.9 The Integrated Control Capability

The Integrated Control Capability provides automation of procedures involving multiple machine elements. The goal of Integrated Control, also called coordination, is to command machine elements to reach the proper states in the proper order. Integrated Control functions typically automate common tasks that are subtasks of many complex shot procedures. The important concept is that the states of machine elements interact and must be carefully staged to achieve a goal or to prevent damage to one or more machine element. Integrated Control functions take these interactions into account when commanding machine elements.

Integrated Control functions utilize Unit Actions as the primitive agents affecting the state of the accelerator. Integrated Control programs receive Action, Alarm, and Milestone Reports from various CATs, make decisions based on the current states and shot goals, and output Unit Action commands to modify the state of the accelerator. The Transaction Processors implementing Integrated Control functions contain large amounts of operational knowledge. Figure 17 shows the process involved in the Integrated Control Capability.

To initiate an Integrated Control function, the operator issues a touch command on some display panel. The Integrated Control process initializes its internal tables from the HGTAB data base, and maps into the desired Action, Alarm, and Milestone Reports. The Element Objective Files for machine elements participating in the current shot are read and stored internally by the Transaction Processor implementing the Integrated Control Function. The process issues the Unit Action commands necessary to reach the goal state as the state of the accelerator evolves taking reports from various CATs as cues for action. Reports are received from various CATs indicating Actions, Alarms, or Milestones. Reports may also be issued from the Integrated Control Process, indicating a higher order alarm has occurred or, simply to communicate status to a Display Monitor. Integrated Control Functions are complete when all specified machine elements have reached their target states. At this point, another Integrated Control Function may be initiated to further the shot goals.

Mixed modes of control are also possible. Some machine elements may be controlled in Remote Manual Mode or Unit Action Mode, while others are controlled in Integrated Control Mode.

The only restriction is that the elements not under Integrated Control not interact with those under Integrated Control. This prevents the dangerous situation known as

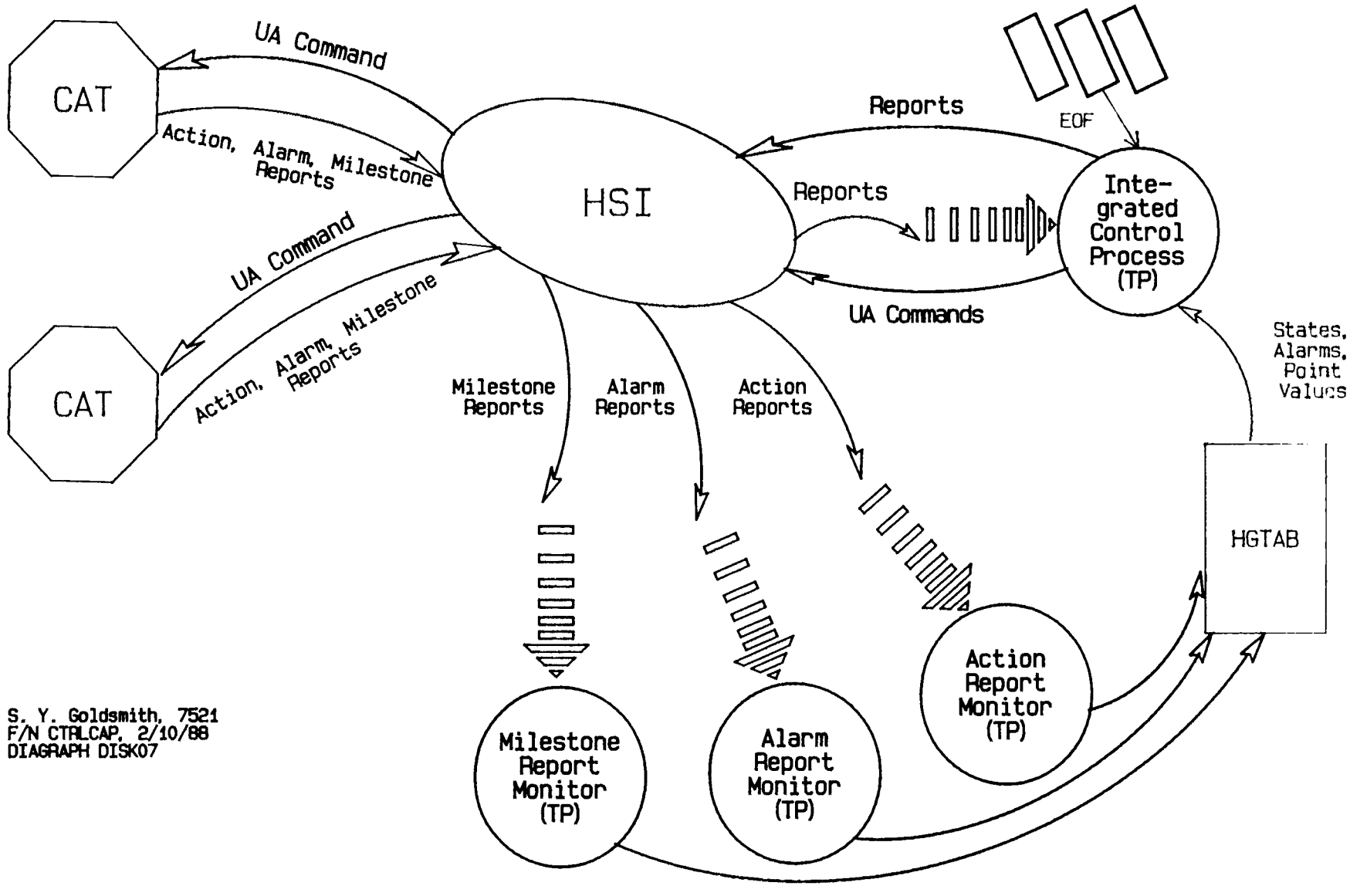

FIGURE 17 - INTEGRATED CONTROL CAPABILITY

control migration, wherein interdependencies among machine element states are not accounted for by operators or the computer system.

4.10 Automatic Staging Capability

At this writing, the Automatic Staging Capability has not been implemented, although much of the design effort went into insuring the proper interfaces were in place to accomplish automatic operation. The primary reason for not implementing the Automatic Staging Mode is that it requires a major commitment to knowledge-based systems technology. It is not clear that automatic capability is needed given the current operating modes of the accelerator. Moreover, there is risk involved in such a project, and it requires an extensive effort that is costly in terms of time and personnel. Nevertheless, should this commitment be made, much of the analysis and design has been completed that will enable automatic staging.

Automatic staging is actually two separate but interrelated processes: one off-line and one on-line. The off-line process involves taking the Shot Objective File and producing the detailed Element Objective File from it. This is a model-based planning task that requires knowledge of parameter interactions, major sequences and subsequences, equipment operating characteristics, and overall facility safety. The software element designated to accomplish this task is the Model-Based Objective Generator (MBOG).

The on-line task is required to carry out the plan produced by the MBOG. The major design problem for this element, called the On-Line Coordinator (OLC), is responding to equipment failures that invalidate the assumptions made by the MBOG about the accelerator state during the planning phase. In particular, choosing a proper abort procedure must be done on-line. This constitutes an on-line replanning task. Even more difficult is continuing a shot in the face of a failure by relaxing some system goals, allowing a marginal return of the cost of shot preparation. These issues are at the forefront of knowledge-based control systems. Some preliminary analysis has been conducted and a simple prototype has been constructed in OPS5. The results indicate that the project is technically feasible, but requires substantial resources to complete.

REFERENCES

[1] Goldsmith, S. Y., Goldstein, S. A., Butler, M. T., "PBFA-II Control/Monitor System Plan", unpublished project document, PBFA-II Project Office, 1983.

[2] Sweeny, M. A., Olsen, J. N., "Brensstrahlung Radiation Environment and Shielding Requirements for the PBFA-II Facility", SAND84-1006, July 1984.

[3] The PBFA-II C/M Design Team, "PBFA-II Control/Monitor System Initial Design," unpublished project document, PBFA-II Project Office, March 1984.

DISTRIBUTION:

1266	L. W. Irwin (5)
1266	S. A. Goldstein (2)
3141	S. A. Landenberger (5)
3151	W. I. Klein (3)
7500	D. M. Olson
3154-1	C. L. Ward for DOE/OSTI (8)
7540	T. B. Lane
7545	J. L. Mortley
8524	J. A. Wackerly
7545	S. Y. Goldsmitn (5)