**SPAWAR Systems**
**Center, San Diego 92152-5000**

Technical Document 3026
Revision A
September 2000

# Multiple Resource Host Architecture (MRHA) for the Mobile Detection Assessment Response System (MDARS)

H. R. Everett
R. T. Laird
D. M. Carroll
G. A. Gilbreath
T. A. Heath-Pastore
R. S. Inderieden
T. Tran
K. J. Grant (CSC)
D. M. Jaffee (CSC)

## ADMINISTRATIVE INFORMATION

This document was prepared by
Code D37 of the
SPAWAR Systems Center, San Diego

Multiple Resource Host Architecture

for the

Mobile Detection Assessment and Response System


Prepared by:

SPAWAR Systems Center, San Diego, CA  92152-5000


## ABSTRACT


The Mobile Detection Assessment and Response System (MDARS) program employs multiple robotic security platforms operating under the high level control of a remote host, with the direct supervision of a human operator. This document describes the major components of a distributed host architecture geared towards a single guard controlling up to thirty-two robots, and explores some of the key issues that were considered in the development phase. The objective is to field a supervised robotic security system that basically runs itself until an exceptional condition is encountered, which requires human intervention.

MDARS uses both interior and exterior security platforms for physical security within buildings and also outside of buildings. A control station directs the platforms on pre-planned patrols, random patrols, or user-directed patrols. The platforms carry payloads for intruder detection and assessment, barrier assessment, and inventory assessment. A centralized database of high-value inventory is routinely compared with observed inventory as monitored by interactive RF tag reading systems onboard the patrolling robots.

# MULTIPLE RESOURCE HOST ARCHITECTURE

## TABLE OF CONTENTS

## TABLE OF FIGURES



## TABLE OF TABLES

# 1. Background

The Mobile Detection Assessment Response System (MDARS) is managed by the US Army Product Manager, Physical Security Equipment (PM-PSE), Fort Belvoir, VA. Mr. Jerry L. Edwards, PM-PSE, is the MDARS Lead Project Officer responsible for overall program management. The Space and Naval Warfare (SPAWAR) Systems Center, San Diego (SSC San Diego) provides technical direction and systems integration functions.

## *1.1 Program Overview*

The MDARS program is a development effort to field interior and exterior autonomous platforms for physical security and inventory assessment functions. The interior platforms are designed to operate in warehouses, office buildings, hospitals, and other semi-structured, enclosed areas where people or property need protection. The exterior platforms are designed for application in storage yards, arsenals, petroleum tank farms, airfields, rail yards, and port facilities.

SSC San Diego provides technical direction for all development efforts on the program. In addition, SSC San Diego is responsible for developing the $C^3I$ architecture that provides coordinated control of multiple (up to 32) interior and exterior platforms, as well as fixed sensor systems. The MDARS program has successfully demonstrated the simultaneous control of interior and exterior robotic platforms. The MDARS Interior (MDARS-I) program is currently in the Engineering Manufacturing Development (EMD) phase, under contract with General Dynamics Robotic Systems (GDRS). The MDARS Exterior (MDARS-E) program is in the Performance Definition and Risk Reduction (PDRR) phase. GDRS developed the exterior PDRR platform under a Broad Agency Announcement (BAA) contract.

The MDARS Control Console is a distributed processing system that coordinates the operation of multiple autonomous interior and exterior remote platforms. The system is designed to run automatically with only minimal human supervision. Guard (system operator) intervention is only required when a patrolling robot encounters an *exceptional condition* such as an environmental hazard or a security breach. Exceptional conditions are prioritized as events and displayed in simple terms to the guard for action. The Control Console human-interface provides both high-level system information (for all robots) via the *Supervisor* display, and detailed operational/diagnostic information (for a single robot) via the *Operator Station* display. The human-interface provides overall situation awareness with the ability to execute pre-planned patrols, random patrols, or user-directed patrols.

The MDARS-I platform is an autonomous indoor robot configured for intruder detection and radio frequency (RF) tag reading for product identification. The platform is based on a Cybermotion K3A SPIMaster with MDARS-specific subsystems incorporated. Each platform will have two primary missions: 1) to patrol a specified area and conduct checks for intruders, and 2) to read RF transponder tags affixed to sensitive, high-valued inventory items. The product data gathered will be utilized for inventory management. Mission modules include an anti-intrusion sensor unit that utilizes both microwave and passive infrared (PIR) sensors, a controllable video camera to support guard assessment functions, and an RF tag reader for automated inventory assessment. Obstacle detection and avoidance are supported with onboard ultrasonic collision avoidance sensors, two scanning laser range finders, near-infrared proximity detectors, and a safety bumper. Each platform will be capable of approximately 12 hours of continuous operation between charges. The recharging process is conducted automatically when a battery charge falls below a specified level. The system is currently in Production Qualification Testing (PQT) and will undergo Limited User Testing (LUT) in February 2001.

In May 2000, the MDARS-E PDRR system underwent Technical Feasibility Testing at Edgewood, MD, conducted by the US Army Test Center (ATC). The platform weighs approximately 1700 pounds and measures 84 inches long by 35 inches high by 50 inches wide, with an 8-inch ground clearance. The four-wheel hydrostatic-drive configuration is powered by a 24-horsepower three-cylinder diesel engine with a 24-volt alternator and integral power steering pump. An Ackerman-steered design was chosen over a skid-steer arrangement for improved dead-reckoning capability. The vehicle was carefully designed with an extremely low center of gravity for maximum stability on uneven terrain. The MDARS-E vehicle is required to operate over paved, gravel, and unimproved roads at speeds up to 9 miles per hour, automatically avoiding obstacles and breaches. The collision avoidance strategy therefore incorporates a two-tier layered approach, wherein a long-range (i.e., 0-100 feet) low-resolution sensor (scanning laser system) provides broad first-alert obstacle-detection coverage, and shorter-range (i.e., 0-30 feet typical) higher-resolution sensors (ultrasonic sensors and a stereo vision system) are invoked for more precise obstacle avoidance maneuvering. The intruder detection system utilizes complementary sensor technologies (millimeter wave radar and forward-looking infrared (FLIR) sensors) to detect the motion of intruders within a 6.6 to 328 ft range, 360 degrees around the platform. The product assessment and barrier assessment systems utilize RF identification technology to automatically inventory tagged items and interrogate instrumented locking devices, respectively.

### *1.2 Multiple Platform Control Philosophy*

The coordinated control of multiple platforms poses some significant design concerns that must be addressed through appropriate tradeoff analysis.

One of the first questions which arises during concept formulation of the overall configuration involves the number of mobile platforms active in the system at any given time, and the corresponding number of human operators needed for effective control. Numerous secondary issues begin to arise as various schemes of implementation are considered, to include distribution of computational resources at both the host and remote ends, communication strategies between the two, and the required human-machine interfaces for real-time multiple-platform operation.

It is impractical to consider a supervised autonomous system in which a single human is tasked with real-time control of a very large number of remote platforms, since by definition a supervised autonomous system implies some degree of human-in-the-loop control. The tradeoff involved is simple: real-time response is going to suffer as the quantity and complexities of operator interactions are increased.

The totality of operator interactions is of course a function of the number of platforms involved and the amount of tasks that require operator involvement. For the currently envisioned MDARS system, with its imposed constraints dictating human involvement, the implication is that one or more platforms may be forced to suspend operations while the guard deals with a higher-priority situation involving another platform.

To eliminate this potential problem, one obvious alternative would be to make the overall system fully automatic and remove the human presence altogether. Patrol platforms would be automatically dispatched, and alarm conditions instantly reported, under a set of pre-programmed guidelines with no possible human intervention.

This option, however, is not feasible for a number of reasons:

- Current technology falls short in providing the necessary navigational and assessment
    - tools required to support this degree of autonomy.
- Political and legal considerations dictate a human-in-the-loop intervention capability for
    - safety reasons.
- A transition period will be required to allow current guard force personnel to adapt to the
    - new technology.

The solution to the problem lies somewhere in between the two extremes discussed above, and involves the right mix of human involvement and computer resources in a distributed system specifically tailored to the needs of the application.

The remainder of this document describes a command and control architecture geared towards a single guard controlling up to 32 platforms, and discusses some of the key issues considered in the actual development. The design objective is to provide a system that basically runs itself until an exceptional condition is encountered that requires human awareness or intervention.

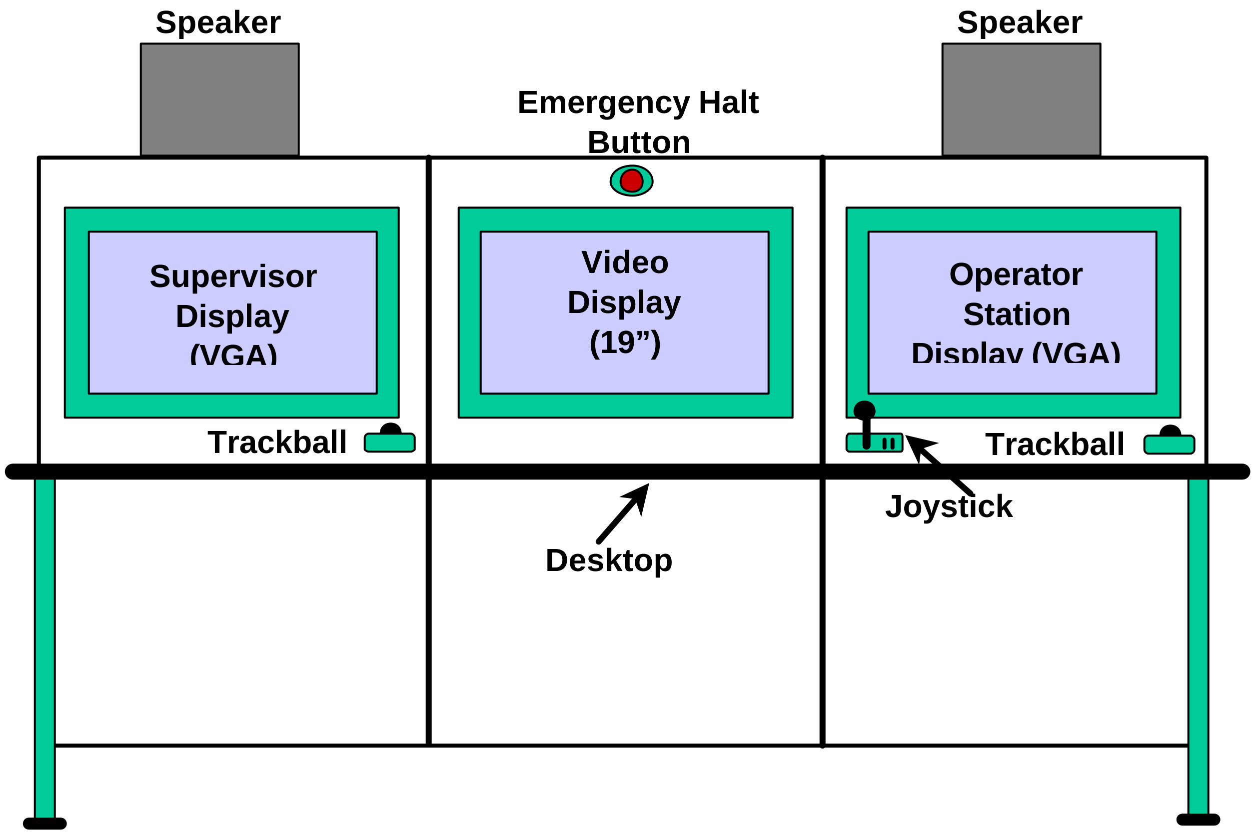



**Figure 1. Prototype Layout of Guard Control Console**

### *1.3 Developmental Approach*

The command and control console, called the Multiple Resource Host Architecture (MRHA), is being developed in a phased rapid-prototyping approach that maximizes across-the-board progress while minimizing technical risk. An iterative "build-test-evaluate" approach has been incorporated to allow user and developer feedback to continuously influence the design, while the operational requirements are systematically identified and matched to the technological needs. The operational requirements in turn have been translated into a detailed breakdown of required system functionalities.

These required functionalities have been broken out into three developmental phases, with each phase composed of three to four distinct categories to facilitate incremental development and in-house testing, and to provide a standardized mechanism for tracking and reporting on same. An additional objective was to apply limited resources to identified technical hurdles in an optimal fashion, in keeping with the degree of risk. Specific functionalities are called out in Appendix B, appropriately modified in this Revision to reflect the addition of the MDARS Exterior phase and completion dates for the Windows NT-based MRHA in addition to the original completion dates for the earlier DOS-based implementation.

### 1.3.1 MDARS Interior Phase

The MDARS Interior (MDARS-I) phase was the initial period during which program focus was on the development of an interior robotic capability. This phase includes four categories of functionalities. The following summaries are included to provide a high level appreciation for the overall intent of each of the four categories.

#### *1.3.1.1 MDARS-I, Category I: Initial Implementation*

The objectives of Category I were to provide a functional host system implementing first-level multiple robot control using one real and three virtual robots, and to ensure a good hardware and software base was established to support further development in Category II.

#### *1.3.1.2 MDARS-I, Category II: Warehouse Navigation*

The primary thrust of Category II was the control of multiple robots (two actual and two simulated robots) navigating in a dynamically changing semi-structured warehouse environment, with the command and control console physically separated from the warehouse. To meet this goal, the system was installed in an active storage warehouse at Camp Elliott in San Diego, CA. The MRHA was housed in a transportable van enclosure and located outside the warehouse. Significant development was completed in the area of robust navigation in a semi-structured environment. Cybermotion incorporated many of these features in later versions of their onboard software.

The capability of reading RF tags affixed to a limited number of items was demonstrated at Camp Elliott during Category II. Approximately 120 tags were placed on boxes distributed around the warehouse in a very low-density arrangement. Tags were read during random patrols and the data was transferred to the product assessment database. The database access computer demonstrated the manipulation and reconciliation that could be done to the data, and various types of sample reports that could be generated.

The MRHA was originally targeted for DOS-based PC machines, at the time Windows NT was not an available option. During the subsequent development of this architecture, the needed system functionalities expanded as the user requirements were explored and better defined. The DOS operating system eventually could not support the memory requirements for the software required to implement the expanded functionalities. LTC Bernard Wilson and Mr. Jerry Edwards were briefed in February 1995 of this situation, and then presented with a recommended solution: transition the MRHA software from DOS to the Windows NT operating system. With their subsequent concurrence, the transition was scheduled to take place after Category II testing was completed. The details of this DOS to Windows NT transition are discussed in Section 1.4.2.

Initial Category II testing was completed by non-SSC San Diego personnel in February 1995. During those tests, 22 Trouble Reports (TRs) and 2 Engineering Change Proposals (ECPs) were generated. All TRs and ECPs not involving the display software were successfully completed and re-tested, while the remaining TRs and ECPs were scheduled to be completed and tested after the operating system conversion.

#### *1.3.1.3 MDARS-I, Category III: Inventory Tagging and Assessment*

The primary focus of Category III was to develop and demonstrate a successful and cost-effective inventory assessment strategy that can demonstrate value-added during EUA. As previously mentioned, the ability to read and approximately locate a low density and minimal number of tags was demonstrated during Category II. Also, a stand-alone database to record and reconcile inventory information was developed. This was a good first step, but to show a cost benefit to the user, a robust inventory assessment strategy must be developed that addresses the full spectrum of issues: 1) tagging and untagging of a product, 2) programming the tag, 3) tag battery life, 4) maximum tag densities that are readable, 5) maximum tag counts that are readable, and 6) interfacing with the existing site database for inventory reconciliation and reporting. Overlooking these significant technical challenges will result in a “feasibility demonstration” system with little or no payback to the user.

SSC San Diego worked with PM-PSE and DLA to locate an active warehouse site in San Diego that would allow formulation of an acceptable and robust solution, with user input, to the existing inventory assessment tasks described above. This effort, however, was not sanctioned by DLA.

#### *1.3.1.4 MDARS-I, Category IV: Early User Assessment*

At the completion of Category III, preparation for installation of the MDARS system at an Army/DLA site was in full swing. Category IV includes the parallel development and integration of video, audio, and data relay capabilities, improved real-world navigation for unanticipated scenarios encountered in targeted facilities, and implementation of the automated navigational re-referencing routine. In addition, site-specific/user-requested emergent work that was critical to the success of the Early User Appraisal (EUA) was addressed.

### 1.3.2 System Redesign

Toward the end of MDARS-I Category II development, it became apparent that limitations of the current operating system would pose a problem for future expansion of the MRHA. The *Supervisor* software, in particular, was using nearly all of the available program memory under *MS-DOS* (640 KB). In fact, in order to perform Category II testing, we were forced to degrade operation of certain software subsystems to free enough memory for the *Supervisor* to successfully boot. This was obviously a problem that needed an immediate, long-term solution or development would not be able to proceed beyond Category II functionality.

#### *1.3.2.1 Operating System Conversion*

The memory problem stemmed from the fact that we had initially underestimated the size of compiled Ada code, and that functionalities were being added as the system evolved and the user’s requirements were better defined and understood. With the added functionalities came the supporting software that quickly consumed available memory. If the system was to be upgraded at all (which was obviously necessary since Category II functionality did not meet the specified operational requirements) a new operating system was needed.

Following Category II testing, an informal survey was performed of newly available operating systems that could replace MS-DOS and provide sufficient computing resources to support Category III/IV development and beyond. The new operating system had to be widely available, relatively inexpensive, supportable by the current target hardware (i.e., rack-mounted PCs), and one for which a validated Ada compiler was available. After extensive deliberation by senior software personnel, Microsoft Windows NT was chosen. Alternatives required either expensive hardware or did not support a validated Ada compiler, or both. Windows NT is inexpensive, is supported world-wide, hosts a number of validated Ada compilers, and executes on the current target hardware (industrial rack-mounted PCs).

#### *1.3.2.2 Software Conversion*

The conversion from MS-DOS to Windows NT required that the existing MRHA software be modified to run under the new operating system, as the old code simply would not execute under Windows NT. To effectively manage the conversion effort, a phased approach was planned whereby Category II functionality would be implemented under Windows NT, and then, after the system had been successfully tested against a baselined test plan and known trouble report set, development would proceed with Category III. This approach minimizes the risk associated with the software conversion task by varying only one aspect of the development process at a time (i.e., the change to the new operating system). Problems would have been compounded if the operating system change was coupled with the addition of new code to address Category III requirements.

Since a re-design of several major MRHA components was necessary to operate under Windows NT, it was decided that all computer software configuration items (CSCIs) not currently written in the Ada programming language would be converted to comply with CECOM direction. This impacted the Operator, Planner, Database Access Computer, and Robot Simulator CSCIs, and resulted in a more robust overall system with significantly reduced maintenance cost projections. Originally the MRHA was implemented using many pre-existing software modules written in four different programming languages: Supervisor, Link Server, Product Assessment Computer, and MDARS Support Computer software in Ada; Operator software in C++; Planner software in C; and the Database Access Computer software in an SQL-based fourth-generation language. Convergence on a single development language facilitates the use of common software components that can be shared by all MRHA application programs. In fact, nearly 50 percent of the MRHA software will be shared among the CSCIs. This factor alone will contribute to long-term software maintenance savings.

The conversion effort produced a more robust and expandable MRHA running on a powerful operating system and utilizing a common programming language that will increase development productivity in the short-term and substantially reduce software maintenance costs in the long-term.

### 1.3.3 MDARS Exterior Phase

The MDARS Exterior (MDARS-E) phase built upon the Interior phase and added functionality to support an exterior robotic capability. This phase includes three categories of functionalities. The following summaries provide a high level overview of each of the three categories.

#### *1.3.3.1 MDARS-E, Category E-I: Basic Platform Control*

The focus of Category E-I was to provide basic control of and communication with exterior platforms using MDARS Interface Design Document (IDD)-compliant messages. Capability was added to support platform status monitoring, emergency halt, Differential Global Positioning System (DGPS) message passing, random patrols, and directed sends. During this timeframe, an exterior platform simulator was also produced to facilitate MRHA development and testing.

#### *1.3.3.2 MDARS-E, Category E-II: Debug, Diagnostics, and Interface*

During Category E-II, basic debugging functions were added to primarily support troubleshooting during development and maintenance. Also, an initial built-in diagnostic capability was implemented to assist both the guard user and the maintenance technician. The human-machine interface was enhanced to support teleoperation, video feedback, and camera control. Final modifications necessary to facilitate operation in Year 2000 were implemented and tested during this period.

#### *1.3.3.3 MDARS-E, Category E-III: Product and Barrier Assessment*

The primary objective during Category E-III was to incorporate functionality required for TFT. This included the handling of RF identification information for the purposes of product inventory assessment and barrier lock-state assessment.

### 1.3.4 MDARS Interior and Exterior Combined Phase

The MDARS-I and E phase builds upon the Interior and Exterior phases and extends capability to support combined operational functionality. This phase is currently being defined to meet overall program goals.

## *1.4 Systems Integration*

Integration activities take place throughout the entire Category development cycle. Following each phase of Category development, however, is a period of system-level integration for both hardware and software components. During these periods of integration, all of the major subsystems are brought together and their interfaces tested and debugged. The software components are locked in configuration management, then a new version is built and installed. After integration is accomplished, in-house Category testing is performed. Category testing is carried out against a formal set of test plans and descriptions with the results recorded and published. To facilitate independent evaluation, non-SSC San Diego testers are often invited to perform Category test procedures.

Systems integration does not in and of itself contribute to increased operational capability; it simply brings together the pieces that embody functionality into a system that can be tested and subsequently operated effectively and reliably. It is, however, a very necessary step in a software-intensive development program that is often overlooked or seriously underestimated.

This philosophy does not mean the system had to be perfect in all respects, however, or that it should be over developed to the point of excess. It simply means that care must be taken to ensure the proper mix of robust functionalities to satisfy preliminary and realistic objectives, with follow-on efforts to harden and optimize for even greater payback. Either extreme can be fatal to an otherwise healthy program. The optimal solution is somewhere in the middle, and it takes a careful and conscientious effort on the part of the developing organization to merge the capabilities and limitations of the technology with the needs of the user. There is no substitute here for first-hand experience and a healthy regard for previous lessons learned.

It was instructive as well to examine the concept of systems integration from the standpoint of technical feasibility testing. Piece-wise demonstrations can show feasibility of all the bits and pieces of needed technology, individually satisfying all the requirements listed in the Operational Requirements Document (ORD). Yet the system as a whole can fail miserably for a number of reasons. In other words, care must be taken to ensure the program doesn't wind up with something that meets the letter of the ORD but not the intent. The system must be soldier proof; if it takes an engineer to run it, it is of little value to anyone, even if it does satisfy the stated needs of the ORD. Above and beyond that technicality, MDARS must satisfy the real-world needs of the user's actual daily routine, which is not addressed in the ORD. Simply put, if it doesn't make the user happy in terms of value added, it fails the acid test. Any hidden vulnerabilities that detract from a "headache free" solution seriously degrade the overall effectiveness.

## 2. System Overview

A high-level block diagram of the Multiple Resource Host Architecture (MRHA) is presented in Figure 2. The heart of the system is a Pentium-based computer with a high-resolution display, to be referred to as the Supervisor, as shown at the top of the diagram. This represents the highest level in the control hierarchy, both from a distributed computational resources as well as a man-machine interface point of view. The Supervisor maintains a ready representation of the "big picture," scheduling and coordinating the actions of the various platforms while displaying appropriate status and location information to the guard.

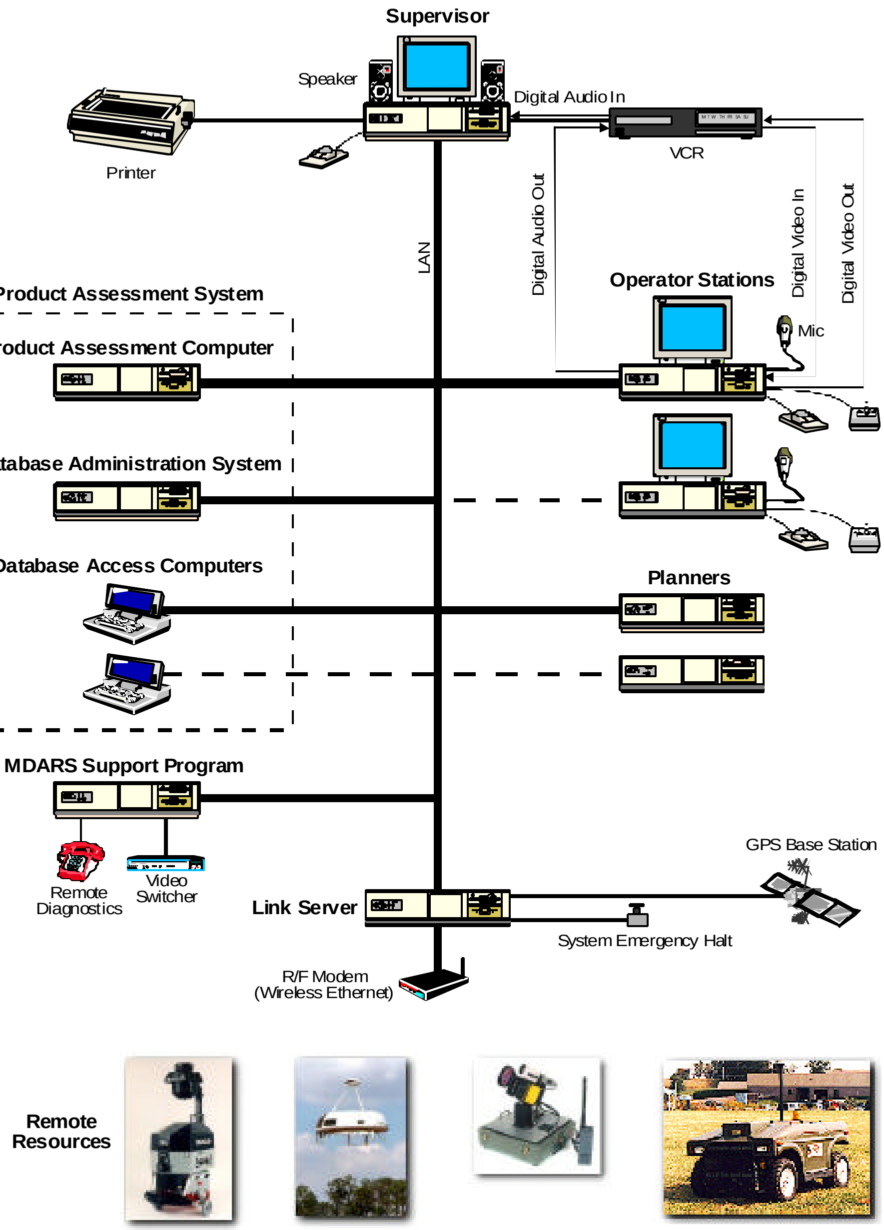



**Figure 2. Block Diagram of the Expandable Host Architecture**

The Supervisor display monitor is located on the left side of the prototype guard console shown in Figure 3. Based on observations made during an early MDARS Operational Test Site Survey, a rack-mounted display may not be feasible at some installation security centers. For example, at Letterkenney Army Depot the MDARS monitor(s) would probably have to be installed at an existing workstation, which implies usage of desktop-style VGAs. Rack-mounted computer equipment such as shown in Figure 4 would then be installed in an adjacent room.

**Figure 3. Prototype of Guard Control Console at SSC San Diego**

The Supervisor has at its disposal a number of Planner computers, linked over a common high-speed local area network (LAN) as shown in the block diagram. These Pentium-based industrial PCs are mounted in a 19-inch equipment rack adjacent to the display console as indicated in the photo of the SSC San Diego prototype shown in Figure 3.

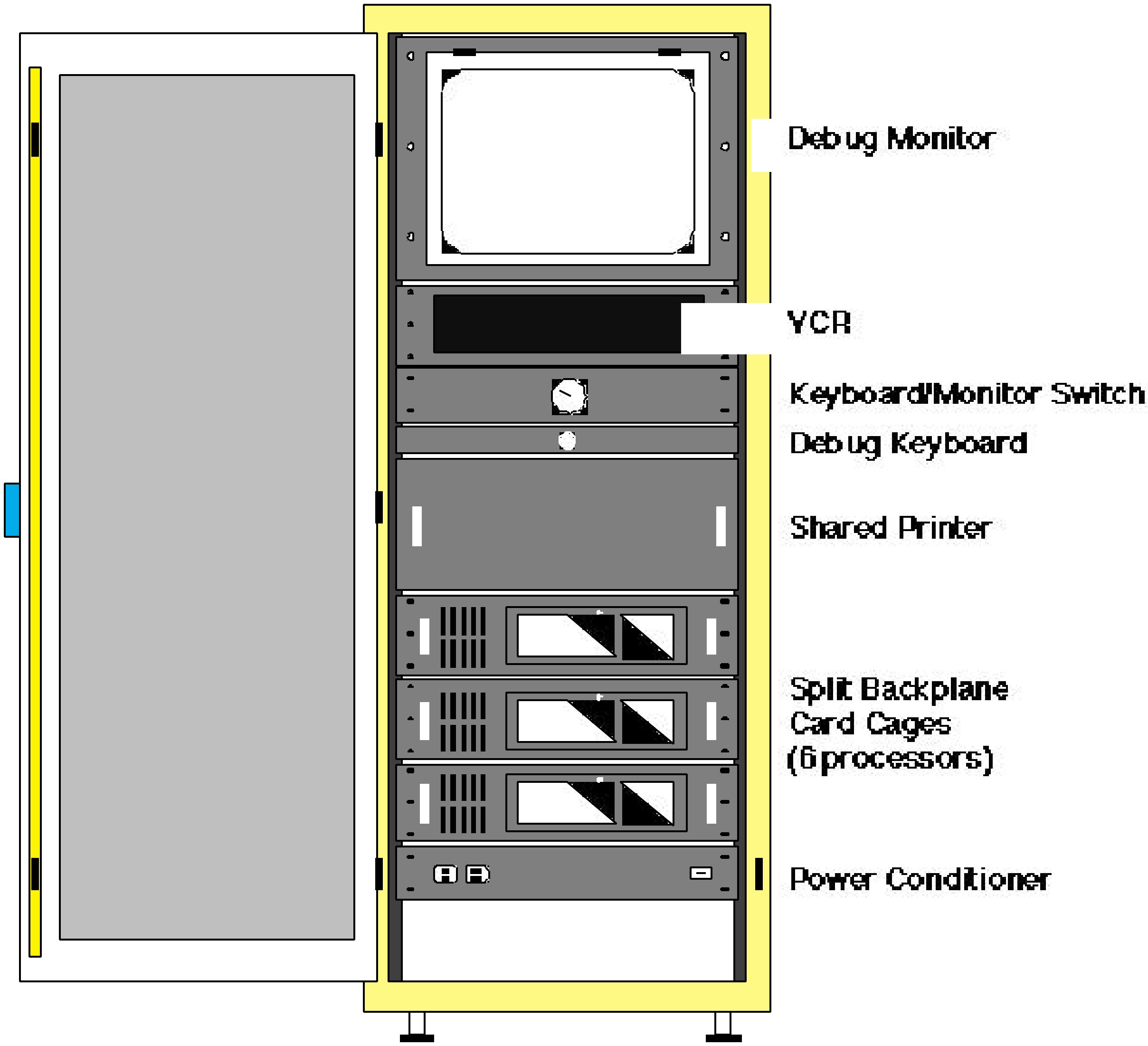



**Figure 4. Optional 19-inch Rack for MRHA Hardware**

Similarly, the Supervisor has access over the LAN to one or more Operator Stations as shown in Figure 2. These Operator Stations are essentially individual control stations with VGA-display capability that can be assigned to a particular platform when the detailed attention of a guard is required. In this fashion, the Supervisor can allocate both computational resources and human resources to address the various situations that arise in the control of a number of remote platforms.

All the distributed resources within the host architecture communicate with the various remote platforms via an Link Server, which is similarly interfaced to the host LAN. The Link Server essentially acts as a gateway between the LAN and a number of dedicated full-duplex spread-spectrum RF modems operating on non-interfering channels. The various resources (Supervisor, Planner, Operator Station) on the host LAN can thus simultaneously communicate as needed in real-time with their assigned remote platforms.

The number of Planner and Operator Station modules resident on the host LAN can be varied as implied in Figure 2 in proportion to the number of remote platforms employed. Preliminary considerations as discussed in the initial MDARS Work Package Review Meeting at ARDEC on 12 September 1991 called for the integration of a number of embedded Planner computers with a one-to-one correspondence to the number of remote platforms. This approach, while clearly the lowest risk alternative, was not viewed as practical since these machines would be largely idle under normal circumstances because a *virtual path* planning operation takes only a fraction of a second to generate a sequence of route segments, which is then downloaded to the platform.

Accordingly, it was decided that the host architecture depicted in the block diagram of Figure 2 represented the optimal solution in terms of minimal hardware configuration without significant tradeoffs in performance and response time. The initial prototype systems being developed by SSC San Diego will be configured with a Supervisor, two Planners, one Operator Station, and one Link Server for coordinated control of up to 32 patrol units (Laird, et al, 1993).

The major components of the MRHA (Supervisor, Planner, Operator Station, Link Server, Product Assessment Database, MDARS Support Program, and Local Area Network) will be discussed in the following sections. The current developmental status of each module is provided at the end of the respective discussions.

# 3. Supervisor

The Supervisor is the Master Control System (MCS) and will be the primary interface to the MDARS system for the guard. During normal operation (no intruders, no obstacles) the Supervisor will execute random patrols for all platforms, display high-level graphical status and location information, and perform scripted operations. Any hands-on control by the guard in response to a situation requiring human intervention (i.e., alarm condition, teleoperation) takes place at the Operator Station (Section 3.3).

Automatic assignment of resources (Planner, Operator Station) will be made by the Supervisor in response to exceptional conditions as they arise, based on the information contained in a special blackboard-type data structure that represents the overall detailed status of every platform in the system. Such exceptional conditions are referred to as events, and typically require either a Planner, or both a Planner and an Operator Station. Example events include: 1) an intrusion alarm, 2) a lost platform, 3) a failed diagnostic, 4) a low battery, and, 5) an off-line platform.

The four highest-priority Events will be displayed in the Event Window on the Supervisor display screen, as discussed in Section 3.1.2.5. The Event Window provides the guard with the ability to track exceptional conditions that have occurred involving other platforms that may not be in that portion of the map currently displayed in the Map Window.

The Link Server will maintain periodic communication with each platform, requesting a certain set of pre-determined status variables in order to make current information readily available in the Supervisor blackboard. The Supervisor will assign the highest priority need as represented in this blackboard to the next available Planner or Operator Station.

## *3.1 Supervisor Functions*

The following general functions have been identified for the Supervisor CSCI:

- Initialization
- Display
- Command
- Event Processing
- Housekeeping
- User Interface
-

These will be discussed in the following subsections. Specific functionalities falling under these general function areas are presented in Appendix B.

### 3.1.1 Initialization Functions

The Supervisor first processes command line options (see 3.1.1.1), followed by initialization file entries (see 3.1.1.2). Once configured, the Supervisor determines the types of all CSCI computers found during network startup, establishes network connections, and sends a Health Check to each CSCI computer. Each Link Server computer will be polled for Platform Health information, providing the Supervisor with a list of all platforms found at startup. This information is incorporated with initialization file information to create Platform Configuration Data, which is then sent to all CSCI computers. The Supervisor then initializes its display screen, and posts initial Robot Lost events for all valid platforms. Normal system monitoring operations will then commence.

#### *3.1.1.1 Command Line Options*

The Supervisor will commence normal operations when invoked using only the program name (*super.exe*). Certain behaviors may be turned on or off using command line parameters, such as disabling sound, enabling packet logging, or using an alternate initialization file. All available options are listed in the Design Document for the Supervisor CSCI of the MDARS MRHA.

#### *3.1.1.2 Initialization File*

The Supervisor is designed to be highly configurable as different installations may have different requirements for MDARS. The Supervisor may be configured for different operations by modifying the initialization file (*super.ini*). The most important function of the initialization file is to specify the number of platforms controlled by the system, and the map, safe zone, and charging location information for each platform. This information will passed down to a Planner when a recharging or referencing operation is required. Other entries may be included to override default values for Event parameters, sound file locations, and diagnostic error handling, as well as scheduling script execution to perform specific tasks at specific times. Specific formats for each data type are listed in the Design Document for the Supervisor CSCI of the MDARS MRHA.

### 3.1.2 Display Functions

The Supervisor display screen is divided into six specific windows (see Figure 5):

- *Help Bar Window*
- *Menu Window*
- *Status Window*
- *Map Display Window*
- *Event Window*

The various display features and the limited number of high-level functions that the guard can perform at the Supervisor monitor are discussed below.

#### *3.1.2.1 Help Bar Window*

A Help Bar is provided to show single-line help messages, amplifying information on screen objects, and to display current time. The message area will address the object currently under the mouse cursor. The Help button is provided to display on-line help for using the Supervisor CSCI.

### *3.1.2.2 Menu Window*

A row of menu buttons is located near the top of the Supervisor display screen as shown in Figure 5. The graphically portrayed menu buttons will be activated by the guard using the mouse, and basically are extensions of the hardware *select button* physically located on the mouse. When the guard places the mouse cursor on the location of a particular menu button and then clicks the hardware mouse *select button*, the software interprets that action as though the selected menu button had in fact been pressed.

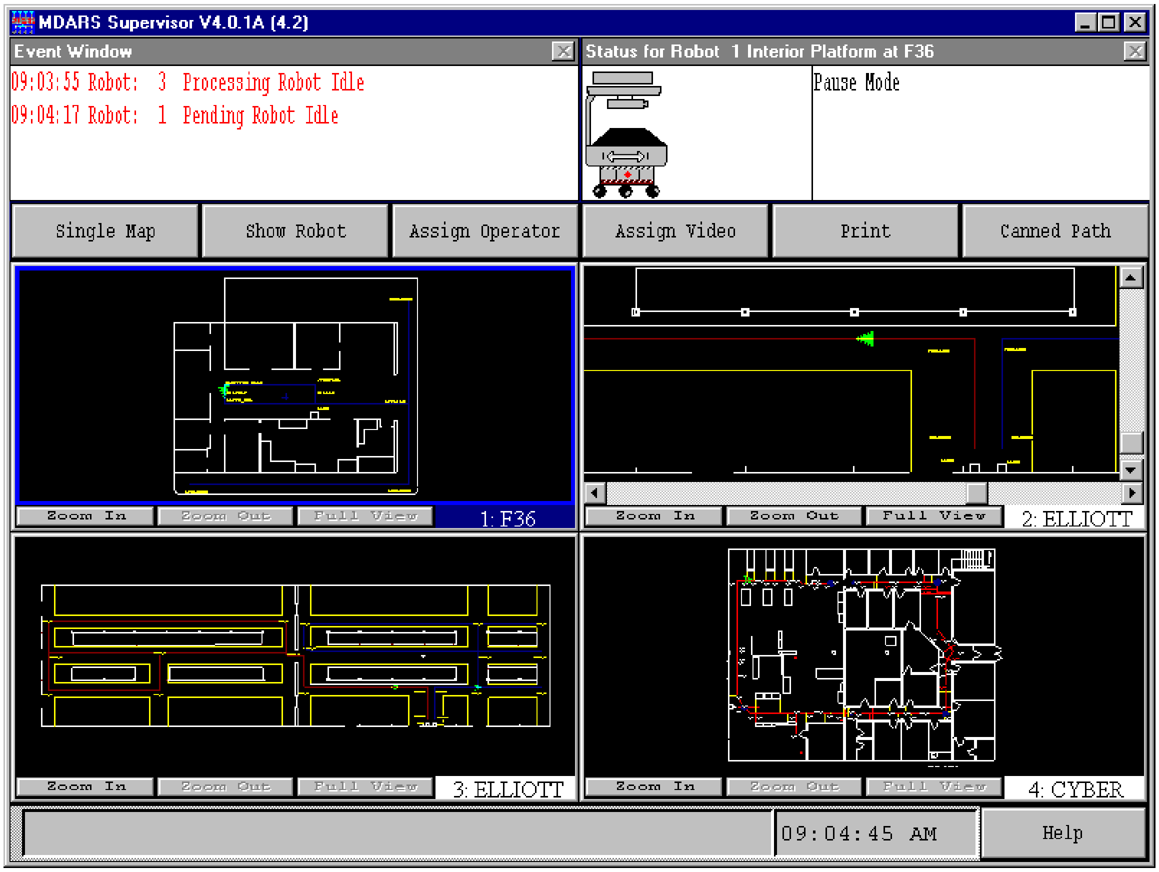



**Figure 5. Actual Screen Dump of Prototype Supervisor Display**

After first clicking on the desired menu function, the guard then selects the platform to which the function will apply. The *platform selection process* offers three methods for selection:

- Using the *Platform Select Listing* - Whenever a button is pushed which requires a subsequent platform selection, a *Platform Select Listing* is overlaid on the *Status Window*. The guard may make a selection by clicking the mouse on the appropriate line in this listing.
- Using the *Event Window* - The guard may also select a platform by clicking on any of the *events* posted in the *Event Window* (see Section 3.1.2.5).
- Using the Platform Icons - The guard may alternatively select a platform by clicking on the associated platform icon (graphical representation of the platform) on the active map.

At any time, moving the mouse cursor into close proximity of a platform icon will cause the platform identifier to be displayed in the Help Bar information window. Clicking the mouse near the icon while the ID annotation is visible will cause the patrol unit to be selected. Moving the mouse cursor away from the icon will cause the ID annotation to disappear. Canceling the unit selection process may be done by clicking on the Cancel button in the Robot Select Listing.

The Robot Select Listing will provide a Cancel button to terminate the selection process for any command. A button titled Video Off is provided for the Assign Video function to disable all video transmitters on all platforms.

The following menu button commands are provided in the Menu Window:

**Single/Four Map** Toggles the map display between single map display and split screen map display modes. The label on the button changes from *Four Maps* to *Single Map* and back based on the current state of the map display.

**Show Robot** Enables the guard to display status and see the associated map of a patrol unit that may not be currently displayed. (This button uses the *platform selection process.*)

**Assign Operator** Manually assign an Operator Station to next platform selected. (Uses the *platform selection process.*)

**Assign Video** Assign the video and audio link to the next platform selected. This function is only available when no platforms are assigned to the Operator Station (Uses the *platform selection process.*)

**Print** Enables the guard to print on-demand a listing on the attached printer of all *events* that have occurred since last printout

**Canned Path** Allows user to execute canned path functions for the next selected platform such as end of shift functions, or inventory patrols. (Uses the *platform selection process.*)

#### *3.1.2.3 Status Window*

The Status Window (depicted on the upper right side of Figure 5) derives its information for display purposes from the blackboard data structure, as partially illustrated below in Table 1.

**Table 1. Portion of blackboard-type data structure used to represent status information for all platforms.**

| Platform ID | X Pos | Y Pos | Platform Heading | Battery Voltage | Survey Threat | Survey Vector | Charging Status |
|---|---|---|---|---|---|---|---|
| **platform1** | 54.6 | 89.0 | 0 | 25.2 | 0 | | Y |
| **platform2** | 23.1 | -195.6 | 90 | 25.2 | 0 | | Y |
| **platform3** | -45.2 | 0.0 | 270 | 25.0 | 0 | | Y |
| **platform4** | 249.0 | -75.8 | 345 | 25.5 | 5 | | Y |
| **platform5** | 112.9 | 100.2 | 135 | 25.3 | 89 | 75 | N |
| **platform6** | 76.4 | 4.9 | 60 | 24.8 | 20 | 330 | N |
| **platform7** | -300.9 | -167.3 | 225 | 25.0 | 0 | | N |
| **platform8** | 10.0 | -35.6 | 180 | 24.6 | 0 | | N |

The title bar of the Status Window is used to display the platform identifier, platform kind, and patrol zone indicator. The left side of the window contains a graphical representation of the platform currently selected for display. The picture resembles the platform being displayed, whether interior or exterior, and active subsystems on the platform are animated to show current status. Many vehicle and environmental status values are graphically depicted with icons to the right of the platform image, such as fire, intrusion or smoke. Placing the mouse cursor on any item in the display causes a one-line description of the graphical object to be displayed in the Help Bar information display. Up to three status indications may be graphically displayed, as well as a maintenance worker icon indicating an off-line status.

The right side of the display is used to display text strings indicating exceptional status indicators or active subsystems on the platform. These status indications may be amplifying information for graphical status indicators in the left side of the window, or status values that are difficult to interpret graphically. The platform's current operating mode is always displayed at the top of the window, followed by zero or more status messages generally presented in order of severity. High severity messages are displayed with white text on a red background, medium severity with black text on a yellow background, and normal message with black text on a white background. Non-standard statuses are displayed with white text on a blue background (such as Camera On or Telereflexive); these are not error conditions, just unusual situations.

### *3.1.2.4 Map Display Window*

#### 3.1.2.4.1 Map Files

There will be a map file associated with each platform in the system, as specified in the Supervisor configuration file. All map files will be located in the same directory as the Supervisor executable.

Individual platform ID numbers are specified in the Supervisor initialization file, and matched up with information in the Link Server's initialization file to associate a given platform identifier with a specific Internet Protocol (IP) address and port number so that the platforms are uniquely identified. To graphically portray the location of a specific platform, the Supervisor cross-references the platform ID with the configuration file map listing, and loads the appropriate map for display. The Supervisor reads the same map files as the Operator Station, the Line-MaP format (LMP), derived from AutoCAD DXF-format files. This is a tagged-image file containing text and line segments that describe a particular patrol area.

#### 3.1.2.4.2 Display Modes

The Map Display Window has two modes: single-map and four-map. Under four-map mode, up to four maps may be simultaneously displayed. When multiple maps are presented, one map is always designated the "active map". This map is identified with a blue border, and the platform's status will be displayed in the Status Window. To activate a different map in four-map mode, the guard clicks the mouse anywhere within the desired map. Each map has a single platform associated with it and identified in the lower right corner of the map window, and that platform will be identified with a filled-in triangle, indicating the location and heading of the platform. If other platforms occupy the same patrol area, they will be represented with hollow triangle icons, also indicating the relative position and heading of the platform. The user may place the mouse cursor near any icon to get positive identification.

#### 3.1.2.4.3 Map Scrolling

On startup, the Supervisor Map Display Window will automatically display the platform listed at the top of the Event Window. If there are no displayed events in the Event Window, the active map will default to the map corresponding to the lowest platform identifier. Active map selection can also be done by clicking in the map area of the desired map.

The map is initially displayed at the Fully Zoomed Out level, though the user may choose to change the zoom level using the menu buttons attached to each map window pane. The Supervisor will automatically scroll the map display to ensure a moving platform remains in view at all times. This automatic scrolling occurs when a patrol unit is within one platform-width of an edge of any map display, with a motion vector that would take it "off screen." The scrolling will optimize the probable on-map display time by scrolling the map in one of eight compass directions, based on the patrol unit's vector. Manual scrolling may be performed by the guard using scroll bars whenever a particular map is zoomed in (i.e., you can't scroll a map that is zoomed all the way out.) If the platform associated with the window is not moving, the user may scroll the map to view any portion desired, though the map will automatically re-center on the platform if it begins moving again. Only the filled-in icon is guaranteed to remain visible in any given Map windowpane. The following buttons are provided in all Map windowpanes to control the display:

| | |
|---|---|
| **Zoom In** | Zooms in on the current center of the map in pre-specified increments. |
| **Zoom Out** | Zooms out from the current center of the map in pre-specified increments. |
| **Full View** | Displays map in fully "zoomed out" state |

#### 3.1.2.4.4 Color Coding

Color coding of icons will be employed on the Supervisor to convey high level information regarding the status of individual platforms in keeping with the color scheme employed under ICIDS, as described in the following extract from page 45 of the "Corps Of Engineers Guide Specifications, Cegs16725, Intrusion Detection Systems":

> *2.4.18.7 Graphic Display Software*
>
> *Graphic display software shall provide for graphics displays that include zone status integrated into the display. Different colors shall be used for the various components and real-time data. Colors shall be uniform on all displays. The following color-coding shall be followed:*
>
> a. *RED shall be used to alert an operator that a zone is in alarm and that the alarm has been acknowledged.*
>
> b. *FLASHING RED shall be used to alert an operator that a zone has gone into an alarm or that primary power has failed.*
>
> c. *YELLOW shall be used to advise an operator that a zone is in access.*
>
> d. *GREEN shall be used to indicate that a zone is secure or that power is on."*

Accordingly, initial color coding of Supervisor icons (against a black background) will be as follows:

**Green** - Normal condition
**Red** - Alarm condition

#### 3.1.2.4.5 Video/Audio Link Assignment

The icon for the platform assigned video/audio capability will appear in the Map Window display with a V-shaped pattern representing the maximum camera field-of-view and current direction of coverage to convey said status to the guard. All video/audio transmitters on the remaining platforms will be deactivated.

### *3.1.2.5 Event Window*

Recall that events are exceptional conditions that require either a Planner, or both a Planner and an Operator Station. Such events can be of two types: assignable events, for which the Supervisor can automatically allocated resources, and, 2) non-assignable events, for which the Supervisor cannot automatically allocate resources. Typical assignable events include an intrusion alarm, a lost platform, a failed diagnostic, or a low battery, whereas example non- assignable Events would be a platform placed in Directed IDS Mode, or Offline Mode, etc.

Assignable event priority is likely to be site-specific, and therefore will be represented in the Supervisor configuration file which can be edited on location if necessary by a service technician. A possible prioritized listing of events is presented below in Table 2.

**Table 2. Events (exceptional conditions) for which the Supervisor may or may not automatically assign resources, listed in descending order of priority, which is site specific.**

| Assignable | Event | Source |
|---|---|---|
| Yes | Manual Assignment | Supervisor |
| Yes | Intrusion Alarm Condition | Platform |
| No | System Emergency Halt | Link Server |
| Yes | Platform Lost | Platform |
| Yes | Fire Alarm | Platform |
| Yes | Emergency Halt Recover | Supervisor |
| Yes | Failed Robot Diagnostic | Various |
| No | Lost Communications | Supervisor |
| Yes | Air Pollution Alarm | Platform |
| Yes | Platform Trapped | Planner |
| Yes | Platform Blocked | Platform |
| Yes | Low Fuel | Platform |
| Yes | Low Battery | Platform |
| No | Directed IDS | Operator Station |
| Yes | Platform Idle Condition | Platform |

The five highest-priority events received by the Supervisor will be listed in the Event Window at the upper left corner of the display (see Figure 5). The highest priority event will always be at the top of the list. For display purposes only, non-assignable events have a higher priority than assignable events.

Multiple events of the same priority will be displayed in chronological order, with the exception of intrusion alarms, which will be ranked in accordance with perceived threat level. The event notification time will be displayed in the leftmost column of the Event Window.

In general, each event listed within the Event Window has a unique platform ID associated with it. The exception to this is Emergency Halt, which will have the platform ID of "ALL". The unique platform IDs within the Event Window are valid select points for the menu button commands presented in Section 3.1.2.2. (Any point on the line associated with a platform listing will be interpreted as a select for that platform.)

The Event Window will only display the highest priority event for each platform. If a platform has a lower priority event in the Event Queue, but the event is not being currently handled when a higher priority event is received, then the lower priority event will be removed from the queue, and the higher priority event will be inserted. In many instances, when a higher priority event is handled, the condition that caused the lower priority event will be gone. If the condition is still persistent, then the original event will be raised again.

If the lower priority event is currently being handled by another CSCI, then the new higher priority event will also appear in the Event Window, and will be assigned as soon as the lower priority event has been completed. Only one CSCI at a time may service a platform. In the future we may send an abort message to the CSCI handling the lower priority event to have the higher priority event handled more quickly.

The color for assignable events will be red, whereas non-assignable events will be portrayed in black text. Events that have been addressed by the Supervisor through assignment of resources will be removed from the list at the time of problem resolution.

The text on each line indicates the current status for the given event. If an event is waiting to be handled, the status will be listed as Pending. If assigned to a Planner for automatic handling, the status will be Processing. If assigned to an Operator Station, the status is listed as Assigned For. This status coding is used to help the user understand the listed event does not indicate a current platform status, but rather a system occurrence currently being handled.

| | | |
|---|---|---|
| 20:26:30 | Robot #1 | Assigned for IDS ALARM |
| 20:29:05 | Robot #2 | Pending MANUAL ASSIGNMENT OPERATOR 1 |
| 20:28:00 | Robot #4 | Pending PLATFORM LOST |
| 20:30:10 | Robot #3 | Processing LOW BATTERY |

**Figure 6. Sample Event Window. The first column indicates time event was reported to Supervisor.**

All active *events*, and not just those displayed in the *Event Window*, will be entered in the Supervisor Log on the hard drive as they are received, and again when resolved. This log may be printed out on the printer at the end of each guard shift, or when so required by the guard using the *Print* button described in Section 3.1.2.2. [Need some more user feedback here on content and frequency of desired hard copy.] In addition, an audible alert (synthesized speech) will be employed to alert the guard each time a new *event* is reported.

#### *3.1.2.6 Command Functions*

The individual menu button command functions will be discussed in more detail in the following subsections.

##### 3.1.2.6.1 Show Robot

The blackboard data structure contains too much status information to present on a single display screen for all the platforms at once. Even if this were physically possible, it would bury the relevant information for a particular event in a sea of unnecessary data. The guard is likely interested only in abnormal conditions, which are automatically flagged and brought to his attention by the Supervisor in the Event Window, discussed in Section 3.1.2.5.

The Status Window is displayed at all times in the upper right corner of the Supervisor display. Show Robot causes the associated platform's status to be displayed, in addition to its associated map (in the Map Window).

#### 3.1.2.6.2 Assign Operator

This feature, known as manual assignment, provides the guard with a means of manually selecting which platform to download to an Operator Station for one-on-one control. A Planner is automatically assigned as well (assuming one is available.) The guard clicks on the Assign Operator button, and then selects the desired platform. Manual assignment will have the highest priority, and the time and date of any manual assignment are automatically recorded in the Supervisor Log.

#### 3.1.2.6.3 Assign Video

The Assign Video menu button is used by the guard to select which platform should have an audio/video channel open. The platform icon selected will be assigned the video and audio RF link, and its associated transmitter will be powered up. The icon thus assigned video/audio capability will then appear on the Supervisor Map Window display with a V-shaped pattern (representing maximum camera field of view and current pan direction) to convey said status to the guard. All video/audio transmitters on the remaining platforms will be deactivated.

When an Operator is assigned, the video link is automatically assigned exclusively to that platform, and may not be manually reassigned. The actual camera selected (Surveillance or Driving) will be determined by the Platform onboard the remote platform in accordance with the current mode of operation. The platform will control the audio/video link until it is released from the Operator Station.

It should be noted, however, that the Supervisor does not immediately assign video to a higher priority event if the Operator Station is involved with another platform. For example, the guard may be manually driving a particular platform in teleoperated mode when an alarm condition arises with another platform. The higher priority need is queued, the Message Window on the Operator Station advises the guard, but the platform being teleoperated doesn't lose its video link until the guard stops the platform and clicks on the Release button. When the Supervisor then assigns the next platform to the Operator Station, the video link is automatically assigned to that platform as well. These features will be discussed in detail elsewhere in this document.

In general, automatic video assignment occurs whenever an Operator is assigned. Potential future configurations, involving multiple Operator and multiple link configurations, will have a conflict resolution mechanism which is currently undefined.

#### 3.1.2.6.4 Print

The Print function causes a submenu to display, providing the user with options of On-Line, Off-Line, and Flush. On-Line causes events to be printed on the attached printer as they occur. Off-Line disables this function, and causes the system to store event information on the system hard drive (if so configured). Flush enables the guard to print on demand a hard copy of all events that have occurred since the last time *Flush* was invoked.

##### 3.1.2.6.5 Single/Four Map

This button is used to toggle the *Map Display Window* between *single-map mode* and *four-map mode*. When the map display is in *single-map mode*, the active map takes up the full *Map Display Window*. When the display is in the *four-map mode*, four individual maps will be displayed in the *Map Display Window*, and the active map is highlighted with an emphasized border. To designate a different map as the active map while in *four-map mode*, the guard clicks the mouse within the desired map on the split-screen *Map Display Window*.

#### *3.1.2.7 Event Queue Processing*

All exceptional conditions reported to the Supervisor are represented in the *Event Window* as *non-assignable events* or as *assignable events*.

##### 3.1.2.7.1 Non-Assignable Events

*Non-assignable events* are those exceptional conditions for which the Supervisor cannot automatically allocate resources. Some examples of non-*assignable events* are discussed below.

###### 3.1.2.7.1.1 Emergency Halt

Emergency Halt can occur by one of two means: 1) the big red switch, or, 2) by Link Server failing to get a response over the net from anyone. The effect is to halt all robots, and dispatch a message.

In the case of an automatically generated emergency halt, a power-on reset of the system will be used to clear the stop. In the case of the Big Red Switch, when the switch is pulled out, Link Server sends a message to Supervisor indicating a button-out condition.

###### 3.1.2.7.1.2 Directed IDS

*Directed IDS* occurs as a result of *Operator* completing with a platform *Directed IDS* status, cleared by *manual assignment* to operator with a completion status indicating normal completion

###### 3.1.2.7.1.3 Lost Communications

*Lost Communications* occurs as a result of failed retries by the Link Server to get data from a platform. The Link Server returns a status age flag indicating the status information is *Stale*, then *Bad*. When status age is *bad*, the Supervisor deems that unit has lost communications capability. The unit is considered to be in essence off-line. The event will be cleared automatically when the Link Server begins sending *current* status packets. The event may also be cleared by manually assigning the platform to an Operator Station.

###### 3.1.2.7.1.4 Off-Line

An Off-Line situation occurs when an Operator Station releases the platform in an off-line status. This event can only be cleared by the guard manually assigning the platform to an Operator and removing the off-line condition.

##### 3.1.2.7.2 Assignable Events

*Assignable events* are those exceptional conditions for which the Supervisor can automatically allocate resources. Some examples of *assignable events* are discussed below.

###### 3.1.2.7.2.1 Low Battery

A low battery condition will be reported in the Supervisor *Event Window*. The interior platform will set a Low Battery status bit when the battery voltage falls below an absolute minimum threshold. The Supervisor will direct the platform to its charging station as soon as the platform reports an idle status (i.e., completed its last assigned path, not under manual control). The path program used to send a platform to the dock must be written to hold the platform on the charger until the charging cycle is complete. Once this program terminates, the platform is considered fully charged and available for patrol activities.

###### 3.1.2.7.2.2 Intrusion Alarm

If a platform is in *IDS Mode* experiencing an intrusion alarm condition, the platform icon will appear in red, and a perceived threat vector will be displayed on the screen. An *Alarm* status will appear in the Supervisor *Event Window*, and an audible alert will sound (Section 3.1.3). The platform will automatically be assigned a Planner and an Operator Station as discussed in Section 3.2.

###### 3.1.2.7.2.3 Low Fuel

A low fuel condition will be reported in the Supervisor *Event Window*. The interior platform will set a Low Fuel status bit when the fuel voltage falls below an absolute minimum threshold. The Supervisor will direct the platform to its charging station as soon as the platform reports an idle status (i.e., completed its last assigned path, not under manual control). The path program used to send a platform to the dock must be written to hold the platform on the charger until the charging cycle is complete. Once this program terminates, the platform is considered fully charged and available for patrol activities.

###### 3.1.2.7.2.4 Platform Blocked

If a platform is blocked, a Planner will be automatically assigned, and *Blocked* status will appear in the Supervisor *Event Window* (Section 3.1.3).

###### 3.1.2.7.2.5 Platform Failure

A platform failure will appear in the *Event Window*, and the guard can use the mouse to select the icon or ID listing which has indicated a problem. When the guard clicks on the *Show Robot* button, and then on the icon, the *Status Window* will depict the condition of the various critical elements for that particular platform. A Planner and Operator Station will automatically be assigned. This will only be posted and assigned if the failure is more severe than “informational”.

###### 3.1.2.7.2.6 Lost or Unreferenced Platform

A lost platform results when the actual navigational parameters (X, Y, and heading) differ enough from the perceived navigational parameters maintained by the platform to where the platform no longer can successfully execute virtual paths. An *unreferenced* robot is a special case of the above, where the perceived navigational parameters of location and heading are cleared by the platform, such as when the platform is first powered up.

A *Platform Lost* condition is most likely caused by an accumulation of dead-reckoning errors or loss of GPS signal for the exterior platform. A common situation occurs when the platform misses certain environmental navigation aids, such as retroreflective stripes, when executing a path program. Another common lost situation occurs when a platform is significantly nearer or farther to walls either to the side or in front of a platform than the platform program indicates. This can typically happen in one of two ways:

- The wall is ahead of the platform, and is being used as the navigational reference for an *Approach* instruction.
- The wall is to the side of the platform, and is probably being used as a Wall-Following reference.

In the first case, the platform is closer to the wall than dead reckoning says it should be, and hence perceives its destination as either closer to the wall than its collision avoidance sensors will allow, or perhaps even inside the wall.

In the second case, the platform's heading is probably slightly off, causing it to head into the wall at an oblique angle. This situation usually means the platform was already too close to the wall when it started execution of the current path, and was therefore unable to use the wall as a reference.

A *Platform Lost* or *Platform Unreferenced* condition may be detected in one of several ways.

- The Platform *Status Robot Lost* bit indicates that the position is invalid.

A flag on the robot is set by the Platform indicating that the platform is lost. The platform will be automatically assigned to an Operator Station for a Robot Lost event.

- The platform is determined to be too far from a starting virtual node to successfully execute a program.

The Supervisor will receive notice from the Planner that the platform is not at a virtual node when attempting a random patrol. If the Operator Station is attempting to perform a directed send, a similar message will be displayed. If not already on the Operator Station, the platform will be assigned for a Robot Lost event.

- The platform indicates a persistent *Blocked* status.

If the platform's real location is different than its perceived position, a platform program may end up directing the platform under a rack, into a crate, or other object in the platform's environment. If the platform is unable to extricate itself, it will be assigned to the Operator Station with a *Robot Trapped* event for assessment by the guard. Using video from the platform, the guard will need to drive the platform away from the obstacle and then reference the platform to enable patrols.

- Procedurally by the guard, when comparing the video display to the map display.

The guard may be able to determine the platform is not where the icon on the map indicates, either by comparing the video returned from the platform and the icon location on the map, or by noticing the icon indicates the platform is in an impossible location, such as in the middle of a rack or a wall. This typically occurs after the robot has been manually driven for a long distance, or sent on a long unrestricted path, either one of which will induce significant dead reckoning errors.

In all cases, the guard must use the Operator Station's joystick to drive the robot in Manual mode to the vicinity of a referencing location and reference the robot using the Reference command. Once the reference action completes, the platform and the MRHA will know precisely where the platform is located and will be able to place the platform back on patrol.

#### 3.1.2.7.2.7 Emergency Halt Recovery

Following an Emergency Halt action, an Emergency Halt Recover event will be generated for each platform in the system. This is done by examining the emergency mode bit of the system status word. It is a Supervisor responsibility to perform this decoding in the course of status monitoring for all platforms.

The Supervisor must decode one of these for each robot in the system. A platform which has been Emergency Halted is deemed lost, and so the normal Operator/Planner Assignment for Platform Lost applies. Emergency Halt Recover events are posted to the log, just like other MDARS events.

### 3.1.2.7.3 The Automatic Assignment Function

In response to *assignable events*, automatic assignment of resources will be made by the Supervisor based on additional information contained in the blackboard. This information represents the detailed status of every platform in the system, as well as the availability of the resources themselves. The example illustrated below in Table 3 shows where platform5, with the highest priority need, has been assigned Planner No. 2, and Operator Station No. 1. Platform3 has been assigned Planner No. 1, but no Operator Station, as none was required. Platform1 and platform7 are queued with lower priority needs, awaiting the availability of resources.

**Table 3. Layout of that portion of the blackboard data structure supporting the assignment function. Assignment priority is taken from Table 2.**

| Platform ID | Assignment Priority | Assigned Planner | Assigned Operator | Map Name |
|---|---|---|---|---|
| platform1 | 4 | | | B100 |
| platform2 | | | | B203 |
| platform3 | 3 | 1 | | B300 |
| platform4 | | | | R151 |
| platform5 | 1 | 2 | 1 | B100 |
| platform6 | | | | B203 |
| platform7 | 5 | | | B300 |
| platform8 | | | | R151 |

#### 3.1.2.7.3.1 Rules for Resource Assignment

The Supervisor will assign the highest priority need as represented in this data structure to the next available Planner or Operator Station, in accordance with the following rules:

If a platform is reports a blocked status, the platform will be assigned to a planner. If the obstacle was previously undetected, a Planner and an Operator Station will be assigned to provide the guard an opportunity to evaluate the situation.

If an object temporarily blocks a platform, the Planner will attempt to negotiate the object.

If the platform is blocked for a long time (i.e., three planning attempts), or if the *unrestricted path planning algorithm* is unable to get around the object (trapped), then an Operator will be assigned to the task as well. The platform may become trapped because there are simply too many obstacles between it and the destination or because the platform is "lost". If the former, the guard needs to tell the platform to go to a different location. If the latter, the platform will need to be re-referenced (see Section 4.7).

- If the platform becomes lost, an Operator and Planner should be assigned to it.
- If a possible intruder has been detected, then both a Planner and an Operator should be assigned to the platform.
- If a diagnostic check fails, a Guard should be notified, and a Planner and Operator Station assigned.
- Planners assigned to an *event* without an Operator Station will automatically be returned to an available status upon successful completion of the response action, without any guard intervention. If the assigned action cannot be successfully completed, the Supervisor is notified and an Operator Station is assigned.
- If the Operator Station is already assigned, but a Planner is available, the lower priority *events* that only require a Planner will be assigned to the available resource ahead of the queued *events* that require both a Planner and an Operator Station.
- There should be at least one more Planner in the system than the number of Operator Stations. This ensures queued *events* requiring the attention of the guard will not interfere with the continued execution of routine random patrols of the other platforms.

In making an automatic assignment, the Supervisor will first determine if the needed resources (Planner, Operator Station, etc.) are available by checking the assignment columns of the blackboard as depicted in Table 3. Assuming the resources are not already committed, the Supervisor modifies the blackboard to reflect the new assignment, and then downloads the following to the appropriate computational resources:

- The platform ID.
- The assigned Planner.
- The assigned Operator Station (if applicable).
- The problem (event code), or reason for assignment.

This information is then used by the assigned resources to resolve the problem, or perform whatever function the Supervisor had in mind.

Resource assignment is done on the basis of single point-to-point message communication across the LAN. This is to avoid race conditions and possible priority conflicts. Thus, when Supervisor is assigning an Operator and a Planner on behalf of some MDARS event, the Supervisor will emit an Assign_Operator message with the *planner_id* as part of the data for this message.

It is up to the Operator to then initiate the planning dialog with the Planner. Similarly, when the Operator is complete, the Operator forwards as a component of the completion message, the planner completion data. Special status fields will indicate if the planner was not evoked, if a planner failure occurred, or if a long planning operation is in progress.

The following special conditions must be supported:

- No Planner available -- in the event that no Planner is available, but an Operator is available, the Operator will be assigned. When a Planner becomes available, the Supervisor will issue another assignment that includes the available Planner ID, and Planner data as needed.
- Planner/Link Server Failure -- the Operator completion status message must support the ability to indicate a resource failure on either Link Server or Planner as well as robot failure
- IDS Alarm Conditions -- When an Operator is assigned on behalf of an MDARS event such as *Platform Trapped*, an IDS alarm condition could occur. The Supervisor has already assigned Planner and Operator resources for that patrol unit. In this case, rather than going through a lengthy reassignment scenario, the Supervisor will send a text message to the Operator as a reminder that the alarm has occurred.

#### 3.1.2.7.3.2 Clearing of *Events*

*Events* are cleared under the following conditions:

- *Manual Assignment* - When Operator returns Operator Complete status.

- *IDS Alarm* - When Operator returns Operator Complete status and IDS composite threat level is below threshold.
- *Platform Lost* - When Operator returns Operator Complete status and lost bit is cleared in platform status.
- *Failed Diagnostic* - When Operator returns Operator Complete status, and platform taken off line or platform no longer reports failed diagnostic.
- *Low Battery* - When Planner returns plan complete status, and not-charged status bit is cleared.
- *Platform Trapped* - When Operator returns Operator Complete with status normal.
- *Platform Blocked* - When Planner returns plan complete, with status trapped or normal.
- *Platform Idle* - When Planner returns planner complete status and normal platform status
- *Emergency Halt* - When *Emergency Halt* message with button out received by Supervisor
- *Emergency Halt Recover* - When Operator returns Operator complete status with normal platform status

This assumes the completion indicated no problems. Exceptional events could result in the event being assigned to new resources for handling. For example, if a Planner failed (refused to respond) to an Operator, Supervisor would go through an assignment cycle with another Planner.

##### 3.1.2.7.3.3 Reassignment of a Suspended Platform

If a platform is suspended by the guard in *Directed IDS Mode* or *Off-Line Mode,* it can not be recovered automatically by the Supervisor for reassignment to an Operator at a later time. *Directed IDS* is thus a *non-assignable event* and will be listed in the *Event Window* in black text. To recover a platform that has been placed in *Directed IDS Mode*, the guard must perform a *manual assignment* at the Supervisor, then free the platform for further random patrols at the Operator Station, as discussed in Section 3.3.1.13.

### *3.1.2.8 Housekeeping Functions*

#### 3.1.2.8.1 Robot Program Storage

Each time a Planner downloads a program to a platform, the Planner sends a copy of the program to the Supervisor. The Supervisor stores the last program for each platform, and downloads the program to another CSCI upon request. The Planners use the previous program when performing a collision avoidance maneuver to determine the path being attempted by the platform when the obstacle was encountered.

#### 3.1.2.8.2 Configuration File Management

The Supervisor Computer will maintain two types of system configuration options: program configuration and site- or installation-specific configurations. Examples of program configuration options include:

- Printer Configuration

- Startup Delay

Examples of site- or installation-specific configuration options include:

- Number of patrol units for the site
- Names of map files
- Map to Patrol Unit Assignment Table
- Frequency at which to dump log to printer

It is likely that only maintenance personnel would modify the site- or installation-specific configuration file, and that only software development/maintenance personnel would modify the program configuration file.

#### 3.1.2.8.3 Dispatcher Database File Management

For each building/map under surveillance there will be a database of virtual paths. This database is constructed in an off-line fashion (using the Virtual Path Database Editor discussed in Section 3.2.1.3.2) from the individual *virtual path* programs. Creation and maintenance of this database will be performed by service personnel only.

### *3.1.2.9 User Input/Output Devices*

The Supervisor and Operator Stations have been similarly configured to provide the guard with consistent, user-friendly visual displays. This approach will result in reduced guard training time and improved accuracy.

#### 3.1.2.9.1 Input Devices

The guard needs to communicate with the Supervisor to utilize the command options available. The user interface device enables the guard to input commands, select options, and designate platform icons on the Supervisor display screen.

The process of selecting an appropriate interface device consists of five steps: define problem; identify and limit candidate devices; examine comparison studies and user and expert experiences; prototype candidate devices (as necessary); and evaluate, select, and implement the chosen device.

Defining the application is straightforward: selecting menu items and platform icons presented on the Supervisor display. The functionality imposed by the application is two- dimensional cursor control and a selection button. Other factors that warrant consideration during the interface device selection process include desired interface response time and accuracy, the targeted user (non-technical, non-supervised), the physical space available, probable exposure to contaminants, ergonomics, and interface consistency.

A number of interface devices are available; the field was limited to three candidates (mouse, trackball, and touch screen) based on functionality, experience, and initial literature reviews. Applicable comparative studies on the candidate devices were reviewed (Albert, 1982; Helander, 1988; Karat, et al, 1986; Sears, Shneiderman, 1991) and user feedback was solicited. Prototype mouse and trackball interfaces have been developed with the Supervisor Display. A capacitive touch screen was tested on a similar interface application.

An evaluation summary of each device follows:

**Table 4. Interface Devices.**

| Device | **Mouse** |
|---|---|
| Advantages | Low Cost<br>Accurate |
| Disadvantages | Requires some training and practice<br>Susceptible to contamination (food, dirt)<br>Susceptible to abuse |
| Unit Cost | $40 - $60 |

| Device | **Trackball** |
|---|---|
| Advantages | Low Cost<br>Accurate<br>Stationary<br>More durable than mouse |
| Disadvantages | Requires some training and practice<br>Susceptible to contamination (food, dirt) |
| Unit Cost | $40 - $60 |

| Device | **Touch Screen** |
|---|---|
| Advantages | Intuitive interface (minimal training)<br>Accurate on targets larger than 0.33" x 0.5"<br>Smaller targets with software assistance<br>Rapid selection<br>Requires no additional desk space |
| Disadvantages | High unit and development costs<br>Some screen technologies susceptible to scratches<br>Accidental activation of controls<br>Finger partially obstructs display<br>Potential user arm fatigue |
| Unit Cost | $500-$1500 |

At this point in the evaluation, several conclusions can be drawn. The mouse and trackball have been tested on the Supervisor display and are functionally acceptable. The mouse and trackball look identical to the display software. A touch screen can be successfully implemented, but it would require display (menu) modification, and software development/adaptation. Finally, no evidence was found to support touch screen implementation that justifies the higher per unit and development costs.

Based on the above conclusions, initial systems will be fielded with a trackball-input device (optional mouse) for the Supervisor. User feedback on these initial systems will be evaluated and any redesign of the interface will be done at that time.

#### 3.1.2.9.2 Output Devices

##### 3.1.2.9.2.1 Smart Switch/Printer Control

A printer is attached to LPT1 via an intelligent parallel autoswitch that allows the same printer to be utilized by the Product Assessment Database computer (Section 7.0).

##### 3.1.2.9.2.2 VCR Control

A serial I/O port is used to communicate with an RS-232-controllable VCR.

##### 3.1.2.9.2.3 Sound Card

A Sound Blaster soundboard is installed in the Supervisor computer to provide pre-recorded messages to the user. These messages are intended to alert the user to extraordinary circumstances such as a newly posted high priority event (see Figure 6). Ordinary events (such as Robot Idle) will be handled in a routine fashion without disturbing the user. Each *event* is configurable (via the initialization file (see 3.1.1.2) to specify if a sound file is to be used, and if so, which sound.

## *3.2 Current Status*

The Supervisor runs under Windows/NT 4.0 or above, and is capable of controlling and displaying up to four platforms for 24-hour random patrol operations. The system will automatically send platforms to charging stations when low battery conditions are reported. Scripts can be scheduled automatically via the initialization file or manually executed to perform inventory or security operations, and to schedule patrol operations for several platforms within a warehouse environment.

Current documentation status for the Supervisor is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)
Design Document for the Supervisor CSCI of the MDARS MRHA (SSC SD, 2000)

# 4. Planner

The current MDARS navigation scheme is basically an integration of the Cybermotion *Dispatcher* and the SSC San Diego *Planner*, which were separately employed on the Phase I prototype to generate *virtual paths* and *unrestricted paths*, respectively. Integration of these two planning algorithms was accomplished by SSC San Diego in FY92 under a Cooperative Research And Development Agreement (CRADA) with Cybermotion, thus giving rise to the term "Planner."

Some consideration was given to porting the navigation algorithms down to the individual platforms for the Phase II effort, as was done in the case of the security assessment software. This was not deemed feasible for a number of reasons:

- The navigational tasks involved in the coordination of eight or more platforms must by necessity have some form of centralized control.
- In order to geographically correlate the location of inventory with the robot's position, there must be some centralized database and world model.
- The requirement for the guard to graphically monitor the locations of all robots on some form of map display essentially means that some Supervisor-type hardware has to be (centrally) located at the control console anyway.

Before reviewing the functions of the Planner, it is helpful to present a brief overview of autonomous navigation, and the techniques employed on MDARS for path planning and collision avoidance.

## *4.1 Autonomous Navigation*

A wide variety of techniques have been developed over the years for the autonomous navigation of indoor vehicles (Borenstein, et al., 1996). For purposes of this discussion, however, these may be grouped into three general categories: (1) guidepath following, (2) unrestricted path planning, and (3) virtual path navigation. Each of these methods has advantages and disadvantages, which determine its appropriate application, as will be discussed in the following sections. The MDARS navigational control scheme seeks to integrate the desired features of all three techniques into a robust navigational package better able to cope with the varied demands of real-world operation.

### 4.1.1 Guidepath Following

The simplest form of autonomous control is sometimes termed guidepath following, and involves a navigational control loop which reflexively reacts to the sensed position of some external guiding reference. Physical paths including slots, buried wires, optical stripes, and other methods for almost thirty years have guided industrial vehicles. Such automated guided vehicles (AGVs) have found extensive use in factories and warehouses for material transfer, in modern office scenarios for material and mail pickup and delivery, and in hospitals for delivery of meals and supplies to nursing stations.

The most common guidepath following schemes in use today involve some type of stripe or wire guidepath permanently installed on the floor of the operating area. Specialized sensors on the front of the platform are used to servo-control the steering mechanism, causing the vehicle to follow the intended route. These guidance schemes can be divided into three general categories (Everett, 1995): (1) those which sense and follow the AF or RF field from a closed-loop wire embedded in the floor, (2) those which sense and follow a magnetic tape in or on the floor, and (3) those which optically sense and follow some type of stripe affixed to the floor surface.

Various implementations of the stripe-following concept exist, including the most simplistic case of tracking a high-contrast (dark-on-light, light-on-dark) line. More advanced optical systems have been developed which track a special reflective tape illuminated by a near-infrared source, and a chemical stripe which glows when irradiated by ultraviolet energy.

Advantages of guidepath control are seen primarily in the improved efficiency and reduction of manpower which arise from the fact that an operator is no longer required to guide the vehicle. Large numbers of AGVs can operate simultaneously in a plant or warehouse without getting lost or disoriented, scheduled and controlled by a central computer which monitors overall system operation and vehicle flow. Communication with individual vehicles can be over RF links or directional near-infrared modulated light beams, or other means.

The fundamental disadvantages of guidepath control are the cost of path installation and maintenance, and the lack of flexibility in the system; a vehicle cannot be commanded to go to a new location unless the guidepath is first modified. This is a significant factor in the event of changes to product flow lines in assembly plants, or in the case of a security robot which must investigate a potential break-in at a designated remote location.

#### 4.1.2 Virtual Paths

The virtual path concept was developed by Cybermotion to provide a routine mechanism for correcting dead reckoning errors in the normal course of path execution. Each desired route is pre-programmed by a technician to take advantage of any available environmental cues that the robot can recognize with its sensors. Each path begins and ends on named virtual nodes as shown in Figure 7. A database is constructed that associates each virtual node with one or more virtual path segments entering or leaving that location. The Planner uses this database to link several discrete virtual path segments together to form a complete virtual path from any given node to any other node.

Cybermotion's virtual path programming is based on a hierarchical structure that allows for easy integration of new sensor systems. The primary movement instructions are the *RUN* instruction and the derivative *RUNON* instruction. These instructions have as their arguments only target speed and the destination coordinates. Given only the *RUN* instruction, a vehicle will turn toward the destination, and accelerate to running speed. Using a ramped velocity profile, the vehicle will begin slowing in order to reach a smooth stop at the destination.

A *RUNON* instruction operates exactly like a *RUN* instruction, except that a *RUNON* preceding another *RUN* or *RUNON* will cause the K3A to execute an arcing turn between the path legs. The radius of this turn is determined by a variable, which can be modified using the "RADIUS" instruction.

The K3A platform has a serial communications network through which it can communicate with its sensor subsystems. When a K3A executes a power-up, it polls this "Control Link" to determine what sensor systems are available. A K3A platform with no sensors will execute a *RUN* or *RUNON* instruction solely by use of its odometry.

Odometry is defined as the process by which a mathematical algorithm is triggered every time the vehicle moves a fraction of an inch. This process is independent of the mode of the platform, and will trigger even if the vehicle is being pushed. Once triggered, the algorithm reads the steering encoder, calculates the relative translation, and updates the vehicle's current position estimate. This estimate is kept in RAM memory, and may be read and modified in a variety of ways. The platform recalculates the direction to its destination continually during a run, allowing its position estimate to be modified at any time (Holland, et al, 1990).

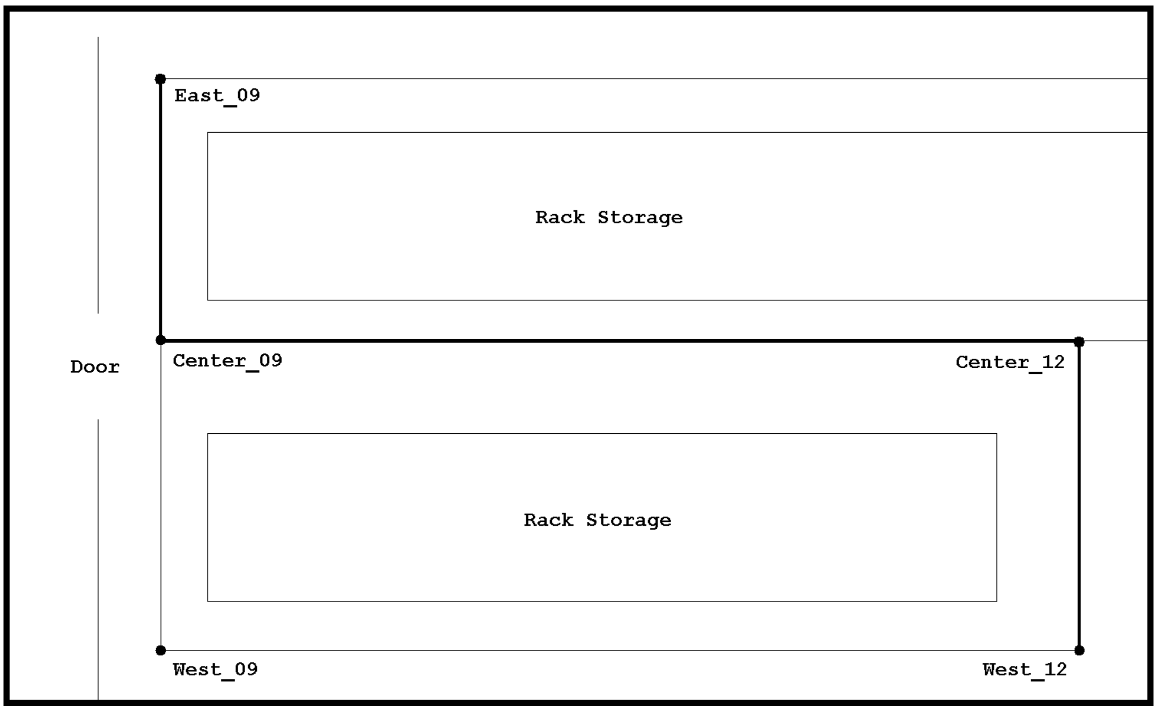



**Figure 7. Virtual path between virtual nodes East_09 and West_12.**

The *Navmaster* is a basic autonomous vehicle, consisting of a K3A platform and a sensor turret with an RF data link and a sonar system. A K3A platform, which finds a sonar system on board at power-up, will also execute a *RUN* instruction by odometry, but it will begin polling the sonar as it executes the instruction. The vehicle may slow, stop, or veer away slightly in response to an obstruction. When the vehicle is clear of the obstruction it will return to speed and drive toward its programmed path destination. To provide control over the parameters that govern this process, a number of instructions have been added to the language, including: *AVOID* (set safety stand-off distances), *WARN* (set beep warning distance), and *C_DEFL* (control veering action) (Holland, et al, 1990).

It is possible to correct the position and heading estimate of the vehicle on-the-fly. This is accomplished by preceding a *RUN* or *RUNON* instruction with a *WALL*, *HALL*, or *APPROACH* instruction. The *WALL* instruction causes the vehicle to monitor the relative range of a *WALL* on either side of the vehicle. This instruction assumes that the path runs parallel to the wall. As the *RUN* instruction executes, points are collected along the wall, and an attempt is made to fit them to a straight line. If the points fit a line within programmable limits, both a heading and single axis position correction is made (Holland, et al, 1990).

The *APPROACH* instruction corrects the vehicle's dead- reckoned position along an axis normal to a wall as it approaches that wall at the end of a *RUN*. *APPROACH* also works with a *RUNON*, even though the vehicle never reaches the destination on an intermediate leg. A typical program using wall navigation to run down two corridors would be:

```
WALL 2,-300             ;Expect a wall for two runs,
                        ;three feet to left
APPROACH NC,300,NA      ;Path to "CORNER" approaches a
                        ;wall at a range of 3.00 feet.
RUNON FAST,CORNER       ;Run at a speed defined as
                        ;"FAST" to a place called
                        ;CORNER.
RUNON SLOW,END          ;Run at a speed defined as
                        ;"SLOW" to a place called END.
```

It should be noted that this program requires only four instructions and provides the vehicle with a rich source of navigational data. The vehicle does not "follow" the wall, but simply uses it as a navigational reference. If a wall is not detected along a path, a navigation error is counted. If this error count exceeds a programmed limit, the vehicle will halt with a "LOST" status (Holland, et al, 1990).

Virtual paths may be programmed in the format above, or programs like this may be generated automatically by drawing paths on a CAD map of a building. The vehicle is thus given only highly condensed, pertinent navigation information. Path programs may also be programmed by walking the vehicle through the route, although this method is only useful with relatively short paths, due to the limited accuracy of the vehicle's odometry. Of the various methods available, the CAD map approach has been found to be the most useful (Holland, et al, 1990).

### *4.2 Planner Functions*

The following general functions have been identified for the Planner CSCI:

- Initialization
- Display
- Random Patrols
- Intrusion
- Directed Movement
- Housekeeping
- User Interface

- Diagnostics

These will be discussed in the following subsections. Specific functionalities falling under these general function areas are presented in Appendix B.

### 4.2.1 Initialization Functions

The Planner must perform the following operations before it is available as an MRHA resource:

- Process command-line arguments (i.e., these typically control debug mode and packet logging.
- Read the initialization file containing site-specific parameters.
- Connect to other computers on the MRHA network.

### 4.2.2 Display Functions

A rack-mounted diagnostic VGA display can be connected to the Planner, providing direct access to various Planner functionalities for developmental purposes only. This display will be removed once the system is ready for fielding.

### 4.2.3 Random Patrol Functions

The Supervisor will automatically assign idle platforms as the situation permits to a Planner for random patrols. The Planner will generate a random virtual path patrol route, and download it via the Link Server to the assigned platform. This onboard program will contain instructions that cause the platform to halt and enter Survey Mode at randomly chosen virtual points along the path. The length of time spent in the Survey Mode at the selected waypoints can be preprogrammed, but most likely will be randomly selected by the Planner.

When a platform arrives at the final destination of the random patrol route, it will report back an Idle Mode status to the Supervisor. The Supervisor will then reassign a Planner, which will generate and download a new random patrol route.

### 4.2.4 Intrusion

When the IDS module onboard the platform determines that an intruder is in the area, the platform will alert the Supervisor via the polling function of the RF Link Server (Section 3.4.2). The Supervisor then assigns a Planner and an Operator Station to enable a guard to look at the situation. It is then the guard's responsibility to determine whether or not there is an actual intrusion. This human assessment may involve some teleoperation and path planning, which is the reason a Planner is always assigned as well as an Operator Station.

### 4.2.5 Directed Movement

Virtual path routes created by the random path generator will involve Survey stops, as explained in Section 4.2.3 above. In the case of directed movement, where the guard Sends the platform to a particular destination, it is desirable to execute the path as quickly as possible. In this situation, the Planner will eliminate the Survey operations altogether, and the path will be executed with no pauses at intermediate virtual points.

#### 4.2.6 User Interface

No user I/O devices are connected to the Planners, except for the VGA displays and keyboards employed during development and troubleshooting.

### *4.3 Current Status*

All Category IV functionalities have been implemented. Debugging is currently in progress.

Current documentation status for the Planner is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the Planner CSCI of the MDARS MRHA (SSC SD, 2000)

# 5. Operator Station

Detailed hands-on control of a remote platform takes place at the Operator Station. The Operator Station is assigned along with a Planner at the request of the guard or automatically in response to an *exceptional event* requiring human intervention. Specifically, the Operator Station will provide the means for the guard to:

- Assess detailed platform status and diagnostic information
- Control remote sensor systems
- Perform directed platform navigation (with the aid of a Planner)
- Manually teleoperate a platform
- Assess a suspected intrusion
- Place a platform in an off-line/power-down mode
- Halt the platform in a static intrusion-detection mode
- Recharge the batteries on a platform
- Re-reference the position and heading information for a platform
- Halt a moving platform
- Operate the system in a degraded state

The guard will also have the ability to suspend the current operation on the Operator Station if a higher priority need should arise. This allows the guard to service another platform before returning the one he is currently working with to random patrolling.

## *5.1 Operator Station Functions*

The following general functions have been identified for the Operator Station CSCI:

- Initialization
- Display
- Command
- User Interface
- Housekeeping

These will be discussed in the following subsections. Specific functionalities falling under these general function areas are presented in Appendix B.

### 5.1.1 Initialization Functions

During the initialization process, the Operator Station software reads and processes a configuration file that permits tailoring the display to support preferences, on-site needs, and overall system configuration.

Also during this time, the Supervisor sends a message containing platform information to the Operator Station. The Operator Station checks to see that all the necessary map and database files are stored on the hard drive to support these platforms.

### 5.1.2 Display Functions

The Operator Station display provides a guard with the means to interact one-on-one with the assigned platform. The Operator Station is described in detail below and shown in Figure 8.

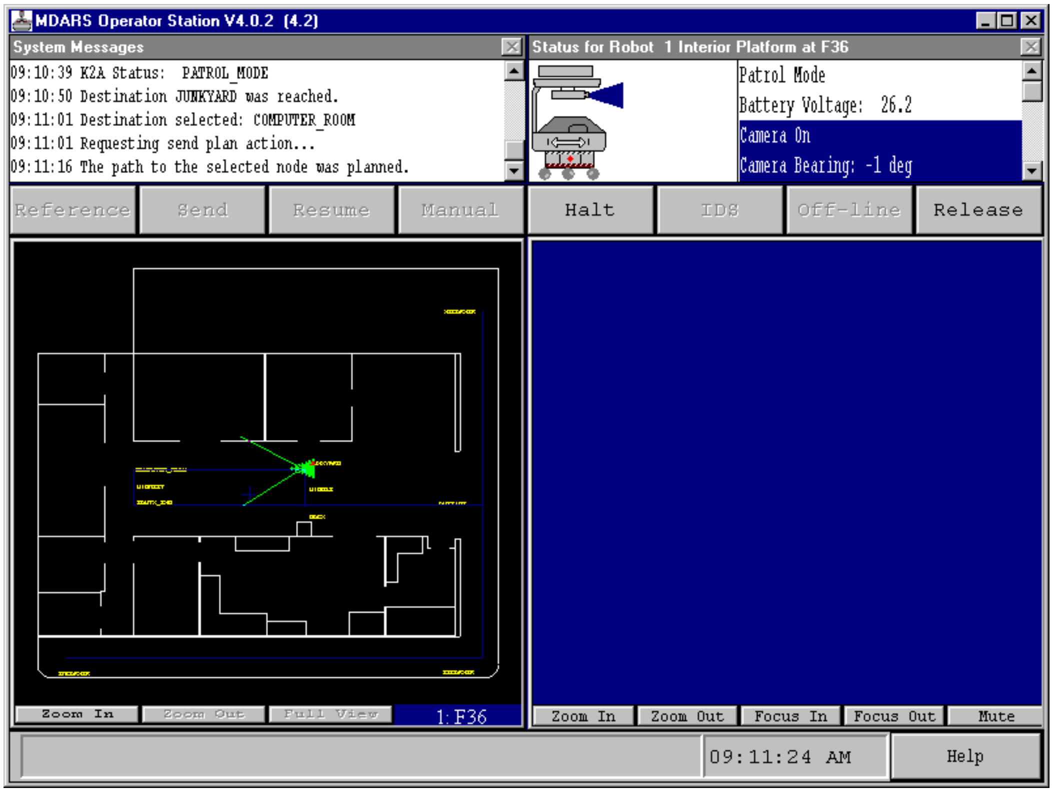



**Figure 8. Operator Station Screen**

#### *5.1.2.1 Unassigned Display Screen*

When the Operator Station resource is not assigned to interact with a specific platform, the video from the platform with the active video link (as selected by the guard from the Supervisor) is displayed on the screen. At the top of the video screen, an information banner displays the platform identification number and the patrol area.

When no platform video links are active and the Operator Station resource is not assigned to interact with a specific platform, a blue screen with black text displaying "Operator Station" will be shown.

#### *5.1.2.2 Standard Display Screen*

Upon assignment, the standard Operator Station screen will be displayed. The CSCI name (Operator Station) and current version are displayed in a title bar located at the top center area of the screen. An entrance window will be displayed to inform the guard of the reason for the assignment and suggested action, if appropriate. The guard must acknowledge this information by moving the cursor to the ***OK*** button and clicking an input device (i.e., mouse, and trackball). The screen cursor used for on the Operator Station standard screen will have a an arrow shape.

The Operator Station is functionally divided into six separate windows:

*Message Window*
*Status Window*
*Map Display Window*
*Video Display Window*
*Button Menu Window*
*Helpbar Window*

Overlay windows are employed to inform the guard of a new situation that may require his attention (critical system overlays), to provide task status information (task status overlays), to provide instructions and menu options for completing a command (sub-menu overlays), and to report an MRHA unrecoverable system failure (Shutdown overlay). Critical system messages are overlaid on the *Message Window*. The following critical message overlay windows are utilized:

*Critical System Message Overlay Windows*
- *Higher Priority Event*
- *System Emergency Halt*
- *System Emergency Halt Cleared*
- *Platform Emergency Stop*
- *Platform Emergency Stop Cleared*
- *MRHA Diagnostic Failure*
- *Command Failed*
- *Platform Lost*
- *Selected Destination Not Reached*
- *Path Plan Time Out*
- *MRHA Communication Failure*

Task status overlay windows are overlaid on the *Status Window*. The following task status overlay windows are utilized:

*Task Status Overlay Windows*
- *Send Path Plan*
- *Reference Path Plan*
- *Collision Avoidance Maneuver*
- *Platform Command*

Sub-menu overlay windows are also overlaid on the *Status Window*. The following sub-menu overlay windows are utilized:

*Sub-Command Overlay Windows*
- *Send Destination Selection*

*Reference Destination Selection*
*Platform Release (exit) Selection*
*Directed Survey*
*Off-line*

5.1.2.2.1 Message Window

This window is located in the upper left area of the screen has an identifying caption, *System Messages*, at the top. It is used to display all system messages in the format of a scrolling log (Figure 8). All messages are time stamped and stored in chronological order with new messages being added to the bottom of the log. A horizontal scroll bar is located along the right hand side of the *Message Window*. The user can use the scroll bar controls to review past messages in the log.

5.1.2.2.1.1 Standard System Messages

Standard system messages are generated by the system to keep the guard informed of normal system operations. These messages are posted directly to the message log and cause the message log to be automatically scrolled to the new message (if it had been manually scrolled to a previous message). Standard messages require no guard action or acknowledgment, their purpose is to provide the guard with information to track what is going on with the system.

5.1.2.2.1.2 Critical System Message Overlay Windows

Critical system messages are generated by the system to inform the guard of an extraordinary situation or event. These messages are posted in a overlay window over the message log (Figure 8) and are accompanied by an audible beep. They require guard input that ranges from an acknowledgment (OK button select) to a decision selected from presented button options. If a guard response is not received for a pre-set period of time, another audible beep is generated. After a guard response is received, the overlay window is erased and the message is posted in the message log and the log is automatically scrolled to the posted message. A thick color band inside the window border indicates the severity of the message. The color scheme used is yellow for critical and red for fatal.

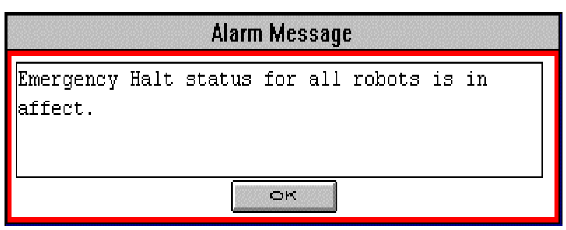



**Figure 9. Critical Message Overlay Window**

5.1.2.2.2 Status Display Window

The *Status Window* is located in the upper right area of the screen adjacent to the *Message Window* on the Operator Station (Figure 8). The window has an identifying caption at the top that includes the platform identification number, the platform type (interior, exterior), and the patrol area. The *Status Window* is used to convey to the guard detailed information on the status of the assigned platform. The status information is visually organized into two “cells” (rectangular boxes) displayed side-by-side in the window area. The cell on the left displays a graphical representation of the platform and mode information. The cell on the right contains a textual representation of current system status information listed in order of importance. Any item in the status window can be selected to solicit more information.

The system status information color coding will include:

| | |
|---|---|
| Black text on white background: | Normal condition |
| Black text on yellow background: | Warning condition |
| White text on red background: | Alarm condition |
| White text on blue background: | Non-standard condition |

5.1.2.2.2.1 Task Status Overlay Windows

Task status messages are generated by the system to inform the guard of an event or action that is in the process of being completed. These messages are posted in a overlay window over the *Status Window* (Figure 10). They contain a textual message describing the event and a graphical status bar that fills with green shading as the event nears completion. Task status messages require no guard response but offer a cancel button option. When the event is complete, the overlay window is erased and an event completion message is posted in the message log.

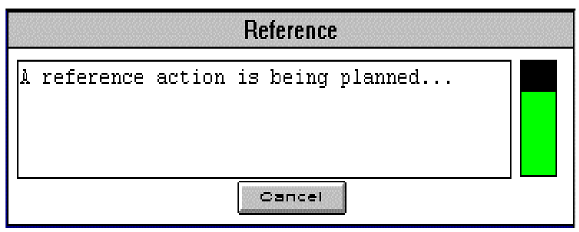



**Figure 10. Task Status Overlay Window**

### 5.1.2.2.3 Map Display Window

The Map Display Window is located just below the Message Window as shown in Figure 8. Horizontal and vertical scroll bars are located along the bottom and right hand side of the Map Display Window, respectively. Below the horizontal scroll bar, four equally sized cells exist. The first three cells (from left to right) are map display control buttons--***Zoom In***, ***Zoom Out***, and ***Full View***. The fourth cell contains a text display indicating the platform identification associated with the map and the name of the patrol area.

The color code scheme for map icons employed on the Operator Station is as follows:

**Green** - Normal condition
**Red** - Alarm condition

5.1.2.2.3.1 Map

There is a map file associated with each platform in the system, as specified in the Supervisor configuration file. The map file information will be passed from the Supervisor to the Operator Station during the initialization process and whenever a platform is added to the system.

##### 5.1.2.2.3.2 Platform Icon

A filled in triangular icon will be displayed on the map to represent the current location of the remote platform. The color of the platform icon follows the color code scheme described above in the *Map Display* section. Any non-filled-in triangles represent other platforms in the patrol area.

##### 5.1.2.2.3.3 Threat Vector

During an intruder alarm, the Operator Station will graphically portray a vector showing the bearing to the detected intruder. The color of the displayed vector maintains consistency with the color code scheme described above in the *Map Display* section.

##### 5.1.2.2.3.4 Camera Icon

When the Operator Station is assigned control of a remote platform, the system automatically assigns the video link to that robot. A "V"-shaped icon, representing the on-board camera, will be displayed on the robot icon and will track the actual camera bearing. If the video link is not operational between the host and assigned robot, the camera icon is erased from the display. The color of the camera icon follows the color code scheme described above in the *Map Display* section.

##### 5.1.2.2.3.5 Virtual Points

When engaged in a send or reference command, virtual navigation points are displayed on the map. Virtual points are pre-designated navigation way- and end-points and are described in detail in Section 4.1.3 Virtual Paths. The displayed virtual points are sized relative to the map and zoom level and subject to a minimum pixel size to ensure readability. The color of the virtual nodes is red.

#### 5.1.2.2.4 Video Display Window

When the Operator Station is assigned control of a remote platform, the system automatically assigns the video link to that platform. The video link can not be assigned to another platform while the Operator Station is engaged. The video image from the remote camera is displayed in the *Video Display Window*, located to the right of the *Map Display Window* (Figure 8). Below the video image are five control buttons--***Focus In***, ***Focus Out***, ***Zoom In***, ***Zoom Out***, and ***Mute***. If the video link is not operational between the host and assigned robot, the buttons will appear gray-shaded to indicate deactivation. In addition, the *Status Window* indicates that the video link is unavailable.

#### 5.1.2.2.5 Button Menu Window

There is a horizontal row of menu buttons below the *Map* and *Video Display Windows* as shown in figure 5-1. These buttons allow the guard to input commands to the system; available commands include the following:

| | | |
|---|---|---|
| ***Reference*** | - | Reset the platform location based on execution of a referencing procedure |
| ***Send*** | - | Send platform to a specified destination |
| ***Resume*** | - | Restore motion after a platform is halted |

| | | |
|---|---|---|
| ***Manual*** | - | Teleoperate (manually drive) the platform |
| ***Halt*** | - | Halt the motion of the platform |
| ***Off-line*** | - | Powers down the platform and removes it as a resource |
| ***IDS*** | - | Place a platform in intruder detection mode |
| ***Release*** | - | Release control of platform and free Operator Station |

#### 5.1.2.2.6 Sub-menu Overlay Windows

Some commands require additional guard input to complete. When the guard initiates a multiple step command, sub-menu overlay windows appear overlaid in the *Status Window* location. The guard must choose from the sub-menu button options to complete the initiated command.

##### 5.1.2.2.6.1 Send Overlay Window

When the **Send** button is selected a sub-menu overlay window is displayed providing assistance information to the guard on how to proceed to move the platform to a new location (Figure 11). A cancel command option is also offered.

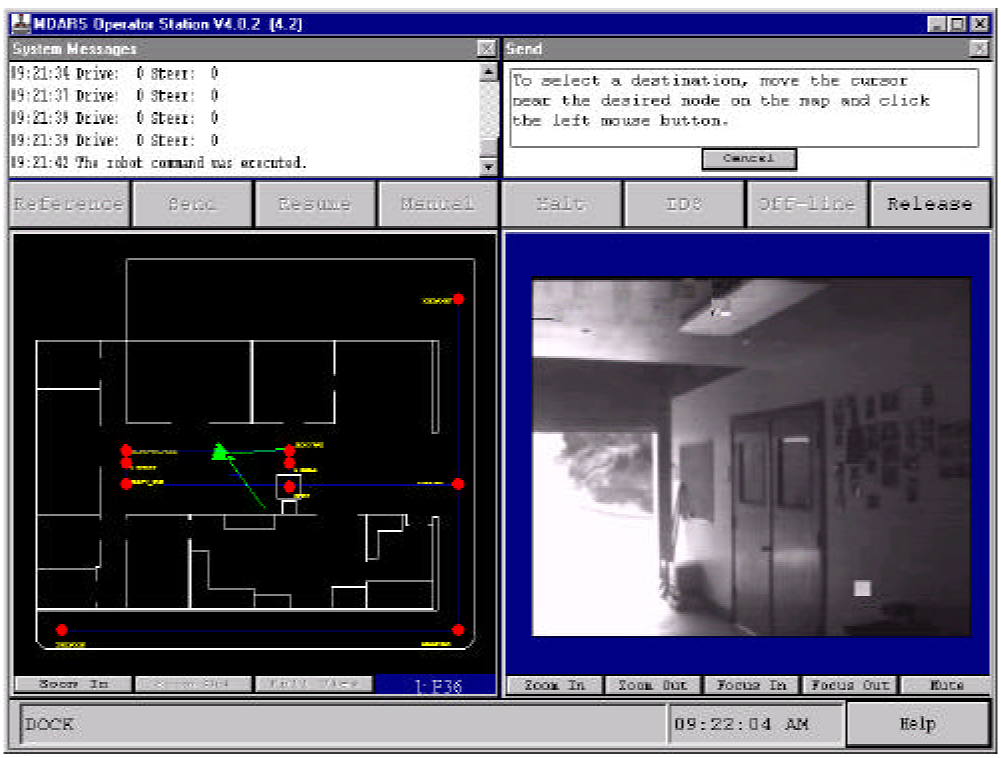



**Figure 11. Send Overlay Window**

##### 5.1.2.2.6.2 Reference Overlay Window

When the **Reference** button is selected a sub-menu overlay window is displayed providing assistance information to the technician on how to proceed to re-reference the platform at a charger or other location (Figure 12). A cancel command option is also offered.

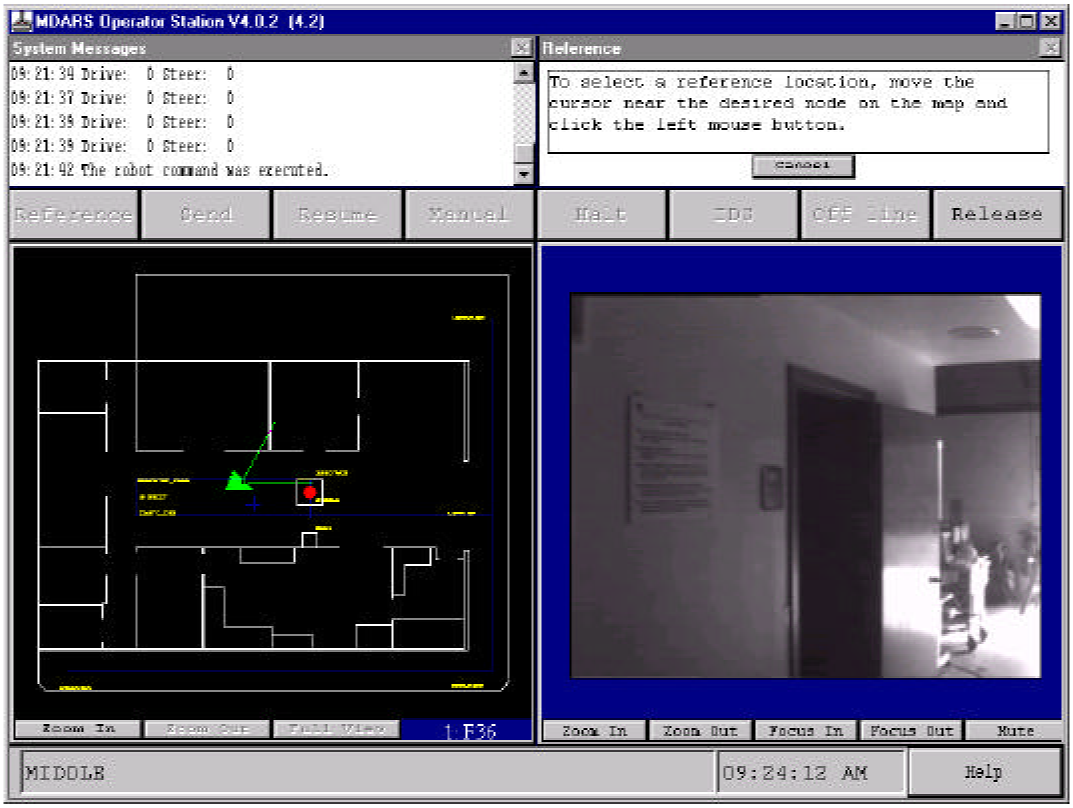



**Figure 12. Reference Overlay Window**

5.1.2.2.6.3 Offline Overlay Window

When the ***Offline*** button is selected, a sub-menu overlay window is displayed. The window text queries the guard about desiring to place the robot in either *Off-line Mode* or *Power-Down Mode* (Figure 13). Button menu options ***Power Up***, ***Power Down***, and ***On-line*** are offered.

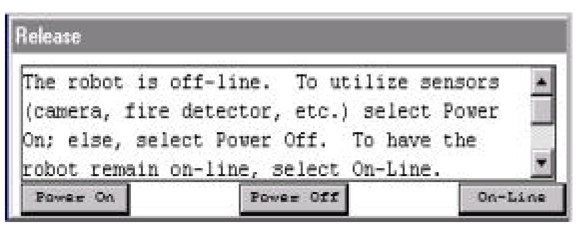



**Figure 13. Offline Overlay Window**

5.1.2.2.6.4 Release Overlay Window

When the ***Release*** button is selected and the guard has left the robot in *Intruder Detection Mode*, a sub-menu overlay window is displayed. The window text queries the guard about desiring to leave the robot in *Intruder Detection Mode* (Figure 14). Button menu options ***IDS On*** and ***IDS Off*** are offered.

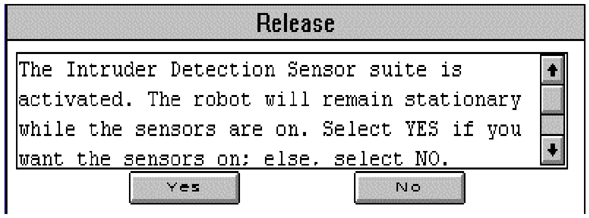



**Figure 14. Release-IDS Overlay Window**

#### 5.1.2.2.7 Helpbar Window

The *Helpbar Window* is located at the bottom of the screen below the *Button Menu Window* (Figure 8). The *Helpbar Window* is used to display text messages that provide information about selectable items on the Operator Station display (buttons, virtual nodes, scroll bars). The appropriate text message is displayed when the cursor is within the boundary of a selectable item. To the right of the assistance information the current time in hours, minutes, and seconds is displayed. At the far right of the *Helpbar Window* is a ***Help*** button. Selecting this button, activates an on-line assistance facility.

### 5.1.3 Command Functions

On-screen controls allow the guard to input commands to the assigned robot as well as control several aspects of the map and video displays. The screen cursor used on the Operator Station will have an arrow shape. The guard can interact with on-screen controls in the following windows:

- *Unassigned Display Screen*
- *Standard Display Screen*
  - *Message Window*
  - *Status Window*
  - *Map Display Window*
  - *Video Display Window*
  - *Button Menu Window*
  - *Helpbar Window*

#### *5.1.3.1 Unassigned Window Controls*

A ***Request_Platform*** momentary button is available to allow the guard to request assignment for a specific platform. To request a platform, the guard selects the ***Request_Platform*** button, then the desired platform.

#### *5.1.3.2 Message Window Controls*

System messages are time stamped and stored in chronological order in the format of a scrolling log with new messages being added to the bottom of the log. A horizontal scroll bar is located along the right hand side of the *Message Window*. The user can use the input device (i.e., mouse, and trackball) to activate the scroll bar controls to review past messages in the log.

##### 5.1.3.2.1 Critical System Message Overlay Window

Critical system messages are generated by the system to inform the guard of an extraordinary situation or event. These messages are posted in a overlay window over the message log (Figure 8) and are accompanied by an audible beep. They require guard input that ranges from an acknowledgment (***OK*** button select) to a decision selected from presented button options. If a guard response is not received for a pre-set period of time, another audible beep is generated.

#### *5.1.3.3 Status Window Controls*

The *Status Window* is located in the upper right area of the screen adjacent to the *Message Window* on the Operator Station. The window has an identifying caption at the top that includes the platform identification number, the platform type (interior, exterior), and the patrol area. The *Status Window* is used to convey to the guard detailed information on the status of the assigned platform. The status information is visually organized into two “cells” (rectangular boxes) displayed side-by-side in the window area. The cell on the left displays a graphical representation of the platform and mode information. The cell on the right contains a textual representation of current system status information listed in order of importance, information in a scrollable format. A horizontal scroll bar is located along the right hand side of this cell. The user can use the input device (i.e., mouse, and trackball) to activate the scroll bar controls to review the detailed status information. In addition, any item in the status window can be selected to solicit more information.

##### 5.1.3.3.1 Task Status Overlay Window

Task status messages are generated by the system to inform the guard of an event or action that is in the process of being completed. These messages are posted in a overlay window over the *Status Window* (Figure 8). They contain a textual message describing the event and a graphical status bar that fills with green shading as the event nears completion. Task status messages require no guard response but offer a ***Cancel*** button option.

#### *5.1.3.4 Map Window Controls*

The map display may be controlled by scrolling the viewport. These commands are entered via scroll bars. Horizontal and vertical scroll bars are located along the bottom and right hand side of the *Map Display Window*, respectively. The scroll bars are actuated with a cursor point-and-click input device (i.e., mouse, and trackball). The horizontal bar controls the horizontal map viewport and the vertical bar controls the vertical map viewport. The scroll bar thumbs indicate the displayed portion of the map relative to the entire map.

Below the horizontal scroll bar, four equally sized cells exist. The first three cells (from left to right) are map display control buttons--***Zoom In***, ***Zoom Out***, and ***Full View***.

Map ***Zoom In*** and ***Zoom Out*** are auto repeat buttons, located in cells one and two respectively (Figure 8). These functions allow the guard to zoom in or out the displayed view of the map.

The map may be manually or automatically scrolled. Horizontal and vertical scroll bars will be located along the bottom and right hand side of the *Map Display Window*, respectively. Automatic scrolling occurs when the platform is in motion; the displayed portion of the map scrolls, as the platform moves, to always maintain the platform in the displayed region. When the platform is halted, manual scrolling is enabled. The scroll bars are actuated with a cursor point-and-click input device (i.e., mouse, and trackball).

***Full View*** is a momentary button located in cell three (Figure 8). Depressing this button centers the map in the *Map Display Window*, zooms the view all the way out, and displays the map at the lowest detail level (if applicable).

The fourth cell contains a text display indicating the robot identification associated with the map and the name of the patrol area.

### *5.1.3.5 Video Window Controls*

Below the video image are five control buttons-- ***Focus In***, ***Focus Out***, ***Zoom In***, ***Zoom Out***, and ***Mute***..

Camera lens ***Focus In*** and ***Focus Out*** are autorepeat buttons (Figure 8). These functions allow the guard to actuate the lens focus mechanism on the remote camera, supporting intruder assessment.

Camera lens ***Zoom In*** and ***Zoom Out*** are autorepeat buttons (Figure 8). These functions allow the guard to actuate the lens telephoto mechanism on the remote camera, supporting intruder assessment.

***Mute*** is a toggle button (Figure 8). Depressing this button deactivates the audio link from the platform to the control station.

The remote camera view is controlled by commanding a remote pan and tilt mechanism. These commands are entered via an external hardware joystick device, see the *User Interface* section. Fore and aft motion of the joystick will cause the camera to be tilted down and up, respectively. Left and right motion of the joystick will cause the camera to be panned horizontally to the left-side and right-side, respectively. Buttons located on the joystick device provide focus in and focus out and center camera controls.

In addition to manual control of the camera position, the system pans the camera to view the area where the highest intruder threat has been detected. Automatic intruder tracking occurs when the a new alarm is detected from the IDS system. Tracking continues until a no-alarm condition exists or until the guard utilizes the joystick device for positioning the camera.

### *5.1.3.6 Button Menu Window Controls*

There is a horizontal row of menu buttons below the *Map* and *Video Display Windows* as shown in Figure 8. These buttons allow the guard to input commands to the system; available commands include the following:

| | | |
|---|---|---|
| ***Reference*** | - | Reset the platform location based on execution of a referencing procedure |
| ***Send*** | - | Send platform to a specified destination |
| ***Resume*** | - | Restore motion after a platform is halted |
| ***Manual*** | - | Teleoperate (manually drive) the platform |
| ***Halt*** | - | Halt the motion of the platform |
| ***Off-line*** | - | Powers down the platform and removes it as a resource |
| ***IDS*** | - | Place a platform in intruder detection mode |
| ***Release*** | - | Release control of platform and free Operator Station |

#### 5.1.3.6.1 Reference

The ***Reference*** button is used to initiate a re-referencing action in the event of a lost robot. When the ***Reference*** button is selected an overlay window is displayed providing assistance information to the technician on how to proceed to re-reference the platform at a charger or other location (Figure 10). A ***Cancel*** button option is also offered. The various re-reference locations will be displayed on the map and the names will appear in the Reference Overlay Window as the cursor is brought into close proximity. The guard will be able to re-reference the platform by selecting a re-reference node with the input device. A reference action typically takes place with the aid of a navigation beacon, such as the one used on the recharging stations.

#### 5.1.3.6.2 Send

When the ***Send*** button is selected a sub-menu overlay window is displayed providing assistance information to the guard on how to proceed to move the platform to a new location. A ***Cancel*** command option is also offered. The various virtual point destinations will be displayed on the map and the names will appear in the Interface Assistance Window as the cursor is brought into close proximity. The desired destination is selected by clicking the input device. The Planner then generates the appropriate *virtual path* and downloads it to the platform via the Link Server.

#### 5.1.3.6.3 Resume

The ***Resume*** button is used to restore platform operation after a *Halt* command or after an emergency halt action. When the platform is halted during the execution of a virtual path, the platform can resume execution of that path as long as it has not been physically moved or operated in another mobility mode (i.e., *Reflexive*) during the time between the *Halt* and *Resume* commands.

#### 5.1.3.6.4 Manual

The ***Manual*** button is used to place the platform in Telereflexive mode for direct teleoperation with collision avoidance assistance from platform sensors.

#### 5.1.3.6.5 Halt

The ***Halt*** button is used to immediately stop the motion of the assigned platform.

#### 5.1.3.6.6 IDS

The ***IDS*** button is used to place the platform in *Intruder Detection Mode*; it will remain in *Intruder Detection Mode* until the toggle button is released. Security sensor information is posted in *the Status Window* on the Operator Station.

The platform will continue in *Survey Mode* even after the guard clicks on the ***Release*** menu button and verifies the *Survey* exit option. This frees the Operator Station, but the platform will not be sent on patrols by the Supervisor. *Directed Survey* will be listed in the Supervisor *Event Window* as a *Non-assignable Event*. If the *Survey* exit option is not selected, the security sensors will be powered down and the platform will begin patrolling after release.

#### 5.1.3.6.7 Off-line

The ***Off-line*** button is used to place a platform in a non-functioning, low power consumption mode. The platform will continue in *Off-line Mode* even after the guard clicks on the ***Release*** menu button and verifies the ***Off-line*** exit option. This frees the Operator Station, but the platform will not be sent on patrols by the Supervisor. *Robot Off-line* will be listed in the Supervisor *Event Window* as a *Non-assignable Event*. If the ***Off-line*** exit option is not selected, the robot will be returned to a ready state and will be sent on patrols by the Supervisor.

#### 5.1.3.6.8 Release

This menu button is used to exit the Operator Station resource. When the ***Release*** button is selected and the guard has left the robot in *Intruder Detection Mode*, an overlay window is displayed. The window text queries the guard about desiring to leave the robot in *Intruder Detection Mode*. Leaving the platform in *Intruder Detection Mode* is appropriate when it is desired to protect a specific limited area or asset or when the mobility system has suffered a diagnostic failure.

### *5.1.3.7 Helpbar Window Controls*

The *Helpbar Window* is used to display text messages that provide information about items on the Operator Station display (buttons, virtual nodes, icons). The appropriate text message is displayed when the cursor is within the boundary of a selectable item. At the far right of the *Helpbar Window* is a ***Help*** button. Selecting this button, activates an on-line assistance facility.

### *5.1.3.8 Input Devices*

Input devices associated with the Operator Station include: 1) a keyboard, 2) cursor controls, and 3) a joystick.

#### 5.1.3.8.1 Keyboard

The keyboard is only used for development and debugging purposes, it will not be used by the end-user.

#### 5.1.3.8.2 Cursor Control Input Device

The Operator Station supports the use of a pointing device to designate screen actions. The device must provide a Microsoft Windows-compatible interface. The device must provide an X-Y screen location, and a left-button to select screen items. Currently, a three button trackball is used, but other devices could be used, like a touch screen or mouse.

#### 5.1.3.8.3 Joystick Input Device

The Operator Station provides an auto-centering two-axis joystick for controlling the remote camera view by commanding the camera pan and tilt mechanism mounted on the platform. Fore and aft motion of the joystick causes the camera to be tilted down and up, respectively. Left and right motion of the joystick causes the camera to be panned horizontally to the left-side and right-side, respectively. A button located on the joystick device provides a center camera control.

In addition to manual control of the camera position, the system pans the camera to view the area where the highest intruder threat has been detected. Automatic intruder tracking occurs when the a new alarm is detected from the IDS system. Tracking continues until a no-alarm condition exists or until the guard utilizes the joystick device for positioning the camera.

The joystick is also used for telereflexively (manually) controlling the drive movements of a platform. When the platform is in *Manual Mode* and the joystick trigger is depressed, fore and aft motion of the joystick commands the platform to more forward and backward, respectively. Left and right motion of the joystick commands the platform to turn. If the trigger is released the platform is halted.

## 5.1.4 Housekeeping Functions

The Operator Station monitors the status of MRHA resources that are required for working with the currently assigned platform. It sets timers on communications with these resources and reports to the guard faults or failures that would affect control of the robot.

The Operator Station does not maintain platform initialization information; this information is sent by the Supervisor (see Supervisor: Housekeeping Functions).

### *5.1.4.1 Command Line Parameters*

The behavior of the Operator Station can be altered through the use of parameters entered on the command line when the main program is executed. Command line parameters are used to modify program operation during test and development. Typically, commands are provided for turning on and off certain debug capabilities that are not used during normal (i.e., fielded) operation. For normal operation, no command line parameters are used. Parameters are entered from the Windows NT Program Manager **File** menu **Properties** command, and appear after the program name in the **Command Line:** edit box.

The general command line parameter syntax is given below:

*-command [parameter[, parameter...]]*

Where:

*command* is a single ASCII character representing the program command.
*parameter* is an input to the previous command.

The following rules apply:

1. At least one space must appear between commands and their parameters.
2. Commands are not case sensitive so that, for example, “-A” is the same as “-a”.
3. Parameter strings containing spaces must be enclosed in double quotes as in “Banner Message.”

### 5.2 Current Status

A prototype Operator Station has been coded, tested, and debugged in Ada for the Windows NT operating system environment in accordance with Category E-III functional level in Appendix B.

Current documentation status for the Operator Station is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the Operator Station CSCI of the MDARS MRHA (SSC SD, 2000)

# 6. Link Server

The Link Server CSCI provides a communications gateway between host processors on the LAN and the remote robotic platforms.

## 6.1 Link Server Functions

The following general functions have been identified for the Link Server CSCI:

- Initialization
- Display
- Message Routing
- Status Polling
- Emergency Halt
- Data Logging/Eavesdropping
- Housekeeping
- User Interface
- Diagnostics

These will be discussed in the following subsections. Specific functionalities falling under these general function areas are presented in Appendix B.

### 6.1.1 Message Router

The primary function of the Link Server is to act as a message router for directed communications between processors on the host LAN and the remote platforms. The Link Server provides a virtual point-to-point connection between any of the resources on the network (i.e., Planner, Operator Station) and the various platforms via an RF modem network.

The RF modem network consists of one or more Ethernet RF modems stationed within and around the site connected remotely to the guard station via fiber optics or other high bandwidth media. The modems provide standard 10BASE2/5/T interface connectors and are Ethernet 802.3 compatible. Several configurable frequency bands within the 902-928 MHz range and the 2.4-2.4835 GHz range are available for simultaneous use so that multiple independent networks can be established. The modem network is dynamic and operates similar to a cellular telephone network.

Each platform is also equipped with an Ethernet RF modem and is assigned a unique IP address. Platforms are individually IP addressable and communicate with the Link Server using an IP address and a socket port number. Port number 5001 is the default for communications to the MRHA. The UDP IP protocol is used to communicate with both interior and exterior platforms. Communications to a platform involves establishing a UDP/TCP connection between each platform and the Link Server. Once the connection is made, a virtual (serial) data stream is provided for host-platform communications.

Messages are passed from host computer resources to the Link Server over the LAN, with each message having an associated function that the Link Server executes. A primary function is to simply pass the information contained within the body of the message to a platform via the RF modem network. The Link Server will maintain a data structure in the form of a routing table that associates an IP address/port number with a particular platform ID. Messages that are destined for a remote platform will be transmitted to the IP address and port number contained in the routing table that matches (i.e., is indexed by) the indicated platform ID.

Related to message routine, the Link Server is also tasked with forwarding differential global positioning system (DGPS) data to exterior platforms. A serial connection exists between the DGPS base station and the Link Server over which the base station transmits periodic (1 Hz) differential corrections. The serial stream is captured by the Link Server, packetized, and sent to each exterior platform on the UDP connection. The DGPS data is essential to exterior navigation.

### 6.1.2 Status Polling

In order to offload from the Supervisor the tedium of constantly requesting status information from the individual remote platforms, the Link Server will, at pre-determined intervals, poll each platform for critical data such as battery voltage, position, heading, etc. Status polling is second in priority to directed communications as discussed in Section 6.1.1.

Note: Due to increased number of platforms, and the aforementioned polling priority, the Survey report packets from each platform may need to be locally integrated for some finite period to ensure a temporary alarm condition is not missed between updates to the host. Polling of platforms in a recharging status could be performed at a slower rate if required to lighten RF link loading.

Like the Supervisor, the Link Server will maintain a data structure in the form of a blackboard that will contain up-to-date status information on each platform. A mechanism will be provided to ensure that the Link Server does not transfer data to the Supervisor that is “changing,” or only partially updated. Since the status information represents a relatively small number of bytes, it will be routinely transferred over the host LAN as a block and not as individual fields (i.e., requests for individual fields will not be supported).

### 6.1.3 Emergency Halt

A System Emergency Halt button is interfaced to the Link Server computer in such a fashion to send a collective emergency halt command to all platforms. Direct connection to the Link Server makes this feature independent of the functional status of the Supervisor, Operator Station, the various Planner, or the LAN itself. An Emergency Halt action is treated as a non-assignable event by the Supervisor and logged accordingly.

The actual user interface is a non-momentary button-type toggle switch mounted just below the display monitors. This way the guard does not have to look for the correct mouse (Supervisor vs. Operator Station), and then place the cursor on the correct graphic icon to stop all the platforms. See also Section 6.1.6 below.

Emergency Halt commands will be transmitted to all platforms until the button is physically reset by the guard, in the event momentary interference or signal blockage precluded successful reception by one or more platforms. Once the button is physically reset, the Supervisor will be tasked with sequentially assigning each platform to an Operator Station in response to the queued Emergency-Halt-Restore assignable events.

### 6.1.4 Data Logging/Eavesdropping

Raw data log files (exclusive of the event log maintained by the Supervisor) would be generated by the Link Server if such are required during development. This is a non-guard diagnostic feature only, and will probably be deleted after initial development/debugging of the system is completed.

The Link Server should provide a configurable packet-eavesdropping capability for debugging and diagnostic purposes. This feature would allow the system technicians and software support personnel to monitor communication flows for selected platforms, or to intercept and display specific packet types on a temporary diagnostic display.

### 6.1.5 Housekeeping

The primary housekeeping tasks performed by the Link Server include maintaining the UDP connections to the platforms, and processing network requests from other CSCIs on the MRHA LAN.

It is assumed that the virtual connections to the platforms are fragile and may be lost at any time. The Link Server continually checks to see if each platform connection is valid, and attempts to re-establish connections that have been lost or that were never made. As part of its standard operating procedure, the Link Server looks for network requests from other CSCIs and processes those requests. Requests are categorized according to the amount of processing time required to complete the requested function. Function requests that can be executed quickly are processed immediately, while functions that require communications with a platform, for example, are queued for later processing.

### 6.1.6 User Interface

The system Emergency Halt button (Section 6.1.3), is physically located between and just below the Supervisor and Operator Station monitors, within easy reach of the guard.

Other than the diagnostic features addressed above, which would be only a temporary connection, no other user interfaces are required at the Link Server.

## *6.2 Current Status*

The Link Server has been converted to operate under the Windows NT operating system, and is completely written in the Ada programming language. It is currently Category E-III capable with full Ethernet modem and serial modem support.

Current documentation status for the Link Server is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the Link Server CSCI of the MDARS MRHA (SSC SD, 2000)

# 7. Product Assessment System

The Product Assessment System, depicted in Figure 15, tracks the location of selected items in warehouse inventory. The Product Assessment System consists of one or more Database Access Computers (DACs), a Database Administration System (DAS), a Product Database Computer (PDC), a Product Assessment Computer (PAC), radio frequency identification (RFID) tags and tag reader. Specifically, the Product Assessment System will provide the means for warehouse personnel to:

- Automatically inventory tagged items on a personnel-defined periodic basis.
- Inventory tagged items on an as-needed basis.
- Identify missing or moved items or items not before catalogued but identified during an inventory.
- Display expected and actual locations of tagged items in map and text form.
- Manually and semi-automatically enter tag item information into the system.

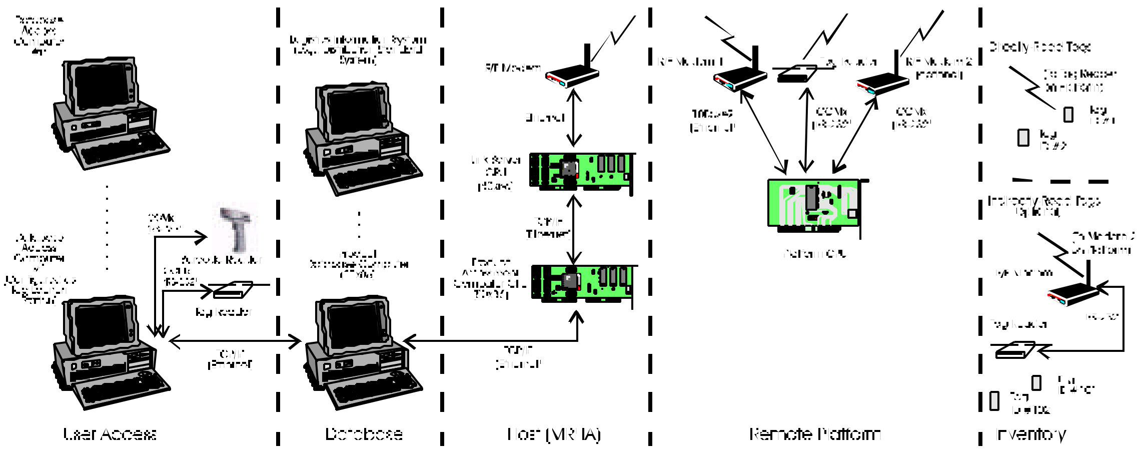


**Figure 15. Overall Block Diagram (PAS)**

The Product Assessment System can be functionally divided into five distinct subsystems:

- The **Inventory Subsystem** located throughout the inventory storage area, and consisting of RFID tags and, possibly, one or more tag readers attached to one or more modems.
- The **Remote Platform Subsystem** - located on the remote platform, and consisting of one tag reader and one or more RF modems connected to the platform's computer systems.
- The **Host (MRHA) Subsystem** - located at the host console as part of the MRHA, and consisting of the Product Assessment Computer (PAC).
- The **Database Subsystem** - located anywhere within the warehouse environment and consisting of the Product Database Computer (PDC) and the Database Administration System (DAS). At some point the database subsystem will also include connection(s) to existing logistics information systems (e.g., Distribution Standard System).

- The **User Access Subsystem** - located at various points within the warehouse environment and consisting of the Database Access Computers (DACs).

Under the Product Assessment System (PAS), RFID tags (each with a unique tag ID) are placed on inventory items. Remote Platform Subsystems perform tag collection and pass the information on tag IDs and their locations to the Database Subsystem (PDC) via the Host Subsystem (PAC) in the MRHA. The PDC also receives input on tag IDs entered via data entry on the User Access Subsystem (DACs). Information on tagged items and their locations is made available to users via reports and maps on a DAC.

The various subsystems of the PAS use different means of communications with one another as depicted in Figure 16. The Remote Platform Subsystem collects tag information remotely using a tag reader, and passes data to the Host Subsystem via a radio frequency (R/F) modem. The Host, Database, and User Access Subsystems communicate with each other via an Ethernet LAN.

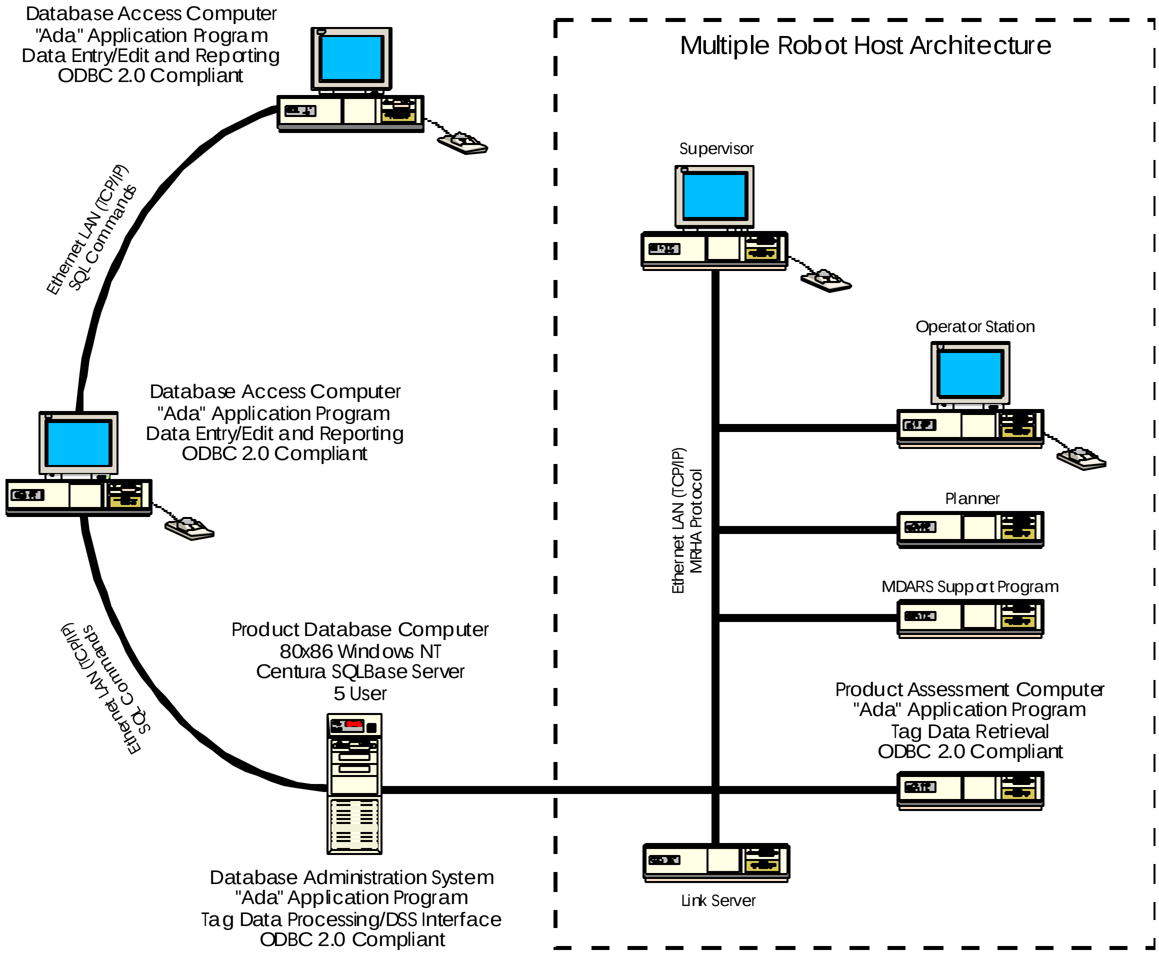



**Figure 16. Connectivity Diagram (PAS)**

## 7.1 Remote Platform Subsystem

### 7.1.1 Remote Platform Subsystem Functions

The functions of the Remote Platform Subsystem include:

- Tag reading operations.
- Transferring the tag information to the PAC when requested.

#### *7.1.1.1 Tag Reading*

Tag reading is the first step in the product assessment process. Tag reading consists of the robot traversing the warehouse or storage depot and executing tag reads at predefined points (events) on warehouse paths. The tag information is collected and transmitted to the PAC on the Host Subsystem. Tag reading is initiated by issuing an "inventory on" command via the Supervisor console.

The tag information is collected from tags, which are attached to various items within the warehouse. A variety of tags, and their associated tag readers, can be used. Tags currently being used are RF Code Spider Tags. Tags are selected based on the ranges at which they can be read, their price, and other factors. The suitability of tags for different environments is decided by PM-PSE.

Once an "inventory on" command has been issued by the Supervisor, the Planner instructs the platform to go into "inventory mode". While in inventory mode, the platform communicates with tag reader(s) at designated stops as the robot vehicle patrols. Depending on the type of tag reader being used and the site layout, the platform may communicate with tag readers in one of several ways:

- It may instruct the reader to perform a tag collection.
- It may request data already collected by a tag reader as the reader and tag communicate independently from the platform.
- It may pass commands and tag data to /from a reader via wireless modems where one modem is on the platform and one modem is attached to the tag reader. This method of indirectly reading tags is used when tags are placed in areas, such as storage buildings, which are out of the range of the tag reader onboard a platform.

After a tag collection is completed, the Remote Platform Subsystem then packetizes the tag data into its onboard memory for transmission to the PAC.

#### *7.1.1.2 Transferring Tag Information*

The PAC gets a status packet from the Link Server that indicates if tag information is available, and if so, how much information is available. When a sufficient amount of tag information is available, the PAC will request that the tag information be uploaded from the platform to the PAC. The PAC will continue reading tag information in this manner until the platform indicates that there is no more data.

### 7.1.2 Remote Platform Subsystem Current Status

Current Remote Platform Subsystem development has reached the completion of Category E-III capabilities, as outlined in Appendix B.

## *7.2 Host Subsystem - Product Assessment Computer (PAC)*

The PAC resides on a Pentium PC on the MRHA rack, and its operation is transparent to a user.

### 7.2.1 PAC Functions

The PAC performs the following general functions:

- Receives tag information from the PAS Remote Platform Subsystem
- Inserts tag information into the MDARS database in the form of temporary survey tables.

#### *7.2.1.1 Receiving Tag Information*

The PAC receives packets of tag information data from the Remote Platform Subsystem, validates the data and logs and displays data errors. The tag information received from each tag reader consists of the tag ID, received signal strength (if available), and the X,Y position of the robot platform when this tag data was received.

#### *7.2.1.2 Insert Into The Database*

The preliminary function of the PAC is to store tag information in the MDARS database. When the PAC receives tag information from the platform, it performs the following steps:

- Connects to the MDARS database.
- Creates a survey table in the MDARS database, with the name of the form "tblTYYYYMMDDHHMMSS" (where YYYYMMDDHHMMSS is the current year, month, day hour, minutes, and second).
- Parses the tag information into SQL-compatible variables.
- For each tag ID in the tag information received, inserts one row into the survey table.
- Once the survey table is completed, renames the form to "tblSYYYYMMDDHHMMSS" (this name is the form expected by the Update function of the Database Administration System (DAS)).
- Disconnects from the MDARS database.

### 7.2.2 PAC Current Status

PAC development has reached the completion of Category E-III capabilities, as outlined in Appendix B.

Current documentation for the PAC is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)
Design Document for the PAC CSCI of the MDARS MRHA (SSC SD, 2000)

## *7.3 Database Subsystem - Product Database Computer (PDC)*

The PDC is an SQL relational database server. Its purpose is to store data on tag IDs and their locations. The computer is a Pentium PC running the Windows NT operating system. The database server software is the multi-user SQLBase from Centura Software Corporation. The PDC also hosts the DAS software. The PDC communicates with the PAC and one or more DACs using the Ethernet LAN (TCP/IP) SQL communications protocol. The PAC and DACs are *clients* of the PDC in this client/server database architecture.

### 7.3.1 Background

Selection of the database server software and user access application development software (which runs on the PAC, PDC and the DACs) was made following a database tradeoff analysis. The purpose of this analysis, conducted by Computer Sciences Corporation (Eatontown), was to compare three commercial-off-the-shelf (COTS) database products for integration into the MRHA to fulfill the depot inventory control mission requirement. Requirements included:

- SQL Requirement: In accordance with HQDA message, 031309Z August 1987, SQL will be used for relational databases as the interface between programs and the supporting DBMS. A waiver is not required for any systems using an SQL-compliant DBMS in conjunction with Ada.
- Ada Requirement: The database selected for the PAS should support the Ada language as an application program interface/precompiler. A waiver is not required for non- developmental software application packages and off-the shelf software not modified by or for DoD.
- Tagging Strategy: The tagging strategy to be employed on the MDARS system must be considered as a factor influencing the database for the PAS.
- DoD Standardization: MDARS must remain fully compatible with standardized RFID systems employed throughout DoD.
- Host Hardware: The PAS database should preferably run on hardware identical to other CSCIs in the Multiple Resource Host Architecture.
- Cost: Acquisition and development costs, as well as site licensing fees, if applicable, must be reasonable and in keeping with the MDARS budget.

Information was gathered from product brochures and technical briefs, as well as telephone conversations with technical support personnel from the respective manufacturers. Several COTS products were reviewed and discarded based on known systems requirements and factors of influence cited above. Only three leading candidates were more extensively evaluated due to time and funding constraints. To summarize the preliminary findings:

- The Centura Structured Query Language (SQL) was better suited for the PC network environment.
- Informix On-line was better suited for the medium corporate-flavored environment.
- Oracle 7 was better suited for the Management Information Systems (MIS) level computing environment.

It is important to note that for this evaluation, which was limited in scope, no one particular product could be definitively selected as the database of choice given the lack of formalized PAS requirements. Computer Sciences Corporation (Eatontown) recommended a follow-on effort further defining a solid set of PAS performance characteristics to narrow the scope of database choices. Only when cost, expandability, and performance is determined can criteria and tradeoffs be established with a given level of confidence and assurance. The Centura multi-user SQLBase database server, C language applications programming interface (for use on the PAC), as well as SQLTalk/Windows and the SQLWindows programming interface (for use on the DACs) were selected. During development, the use of the C language applications programming interface and the use of the SQLWindows programming interface were discontinued in favor of the use of Ada. Also, the use of the Open Database Connectivity (ODBC) standard was selected as a higher level interface to SQL, to allow for more independence from the particular server software being used. It should be noted that, if necessary, a change could be made to different database server software (from among a number of commercial products) without requiring a change to the type of software used on the PAC and the DACs.

### 7.3.2 PDC Functions

The functions of the PDC are:

- Host the Centura SQLBase server software, which accepts and executes SQL commands from the client applications.
- Host the MDARS database, which is the database of tag IDs and their locations. The generic name of the database is dbMDARS, although its actual name will vary by site (e.g., dbRIA for Rock Island Arsenal).
- Host the DAS CSCI.

For Category III, database administration and maintenance functions are performed on the PDC using the SQLTalk user interface included with SQLBase. In Category IV, database administration and maintenance functions will be included in the DAS user interface on the PDC.

#### *7.3.2.1 MDARS Database*

The MDARS database (dbMDARS) contains three permanent tables and a variable number of temporary tables. The permanent tables are the user table, the update table, and the update status table. The user table (named tblUser_Data) contains information, which may be entered and/or updated by human users. Data is organized with one row per unique tag ID, and includes (as a sample):

- Tag ID
- National Stock Number
- Description of Item
- Expected Location of Item
- Condition Code

The update table (named tblUpdate_Data) contains information about each tag ID’s location and status. Data in the table is organized with one row per unique tag ID, and includes (as a sample):

- Tag ID
- Location of platform when tag ID was “read” by tag reader
- Strength of signal when tag ID was “read”
- Assessed location of tag ID
- Flags to indicate whether tag ID is Found, Missing or Moved

The update status table (tblUpdate_Status) contains information about the Update function of the DAS. If Update is currently running, the table indicates current progress. If Update is not running, the table indicates when it ran most recently and some statistics on the results. The DAS stores information in the update status table and the DAC reads information for its Update Status user display.

The update status table contains only one row, and includes (as a sample):

- A flag indicating whether update is currently in progress.
- A designation of phase currently in progress if update is in progress.
- The date/time when the most recent update ended.
- The percent of tags marked missing or moved in the most recent update.

There are also temporary tables created by the PAC (with names of the form ‘TBLSYYYYMMDDHHMMSS’). Each table contains data collected from the platform. The PAC creates these tables and places tag location information in them (see Section 7.2.1.2). These tables are read as part of the update (product location estimation) function of the DAS (see Section 7.3.2.3.1). Once data in a temporary table has been read and processed, the table is deleted.

### *7.3.2.2 SQL Commands from PAC and DACs*

The PAC and DACs perform their database operations by sending SQL commands to the PDC via the Ethernet LAN. The PDC executes these commands and returns result codes and/or data to the appropriate PAC or DAC.

### *7.3.2.3 Database Administration System CSCI (DAS)*

The Database Administration System (DAS) CSCI resides on the PDC and provides an interface for a user, who would generally serve as database administrator, to handle these operations:

- Run the update process (automatically or manually), which performs these functions:
  - Incorporates data stored by the PAC into permanent tables.
  - Computes assessed locations for each tag ID.
  - Marks tag IDs Found, Missing, or Moved as appropriate.
  - Makes the rows of the user and update tables consistent so that each contains rows for all tag IDs.
- Manually clears any or all tables.

#### 7.3.2.3.1 Update Process

Figure 17 shows the main screen for the DAS. Users select functions from Windows-style pull-down menus using keyboard hot-keys, a mouse, or a combination of the two.

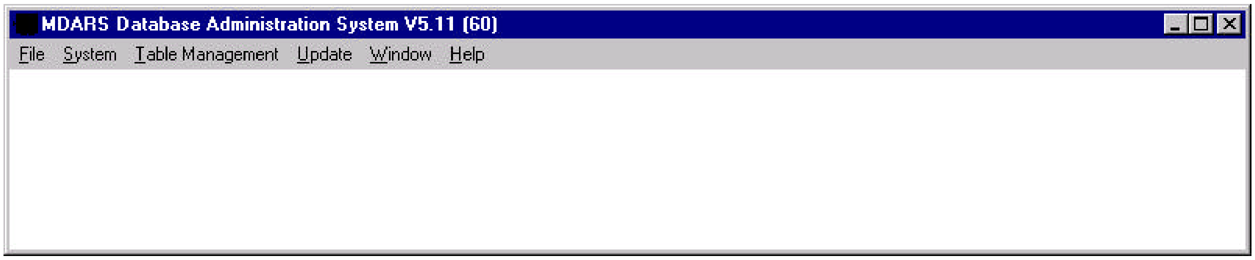



**Figure 17. Main Screen (DAS)**

The update process, which runs as part of the DAS, performs the function of processing the temporary survey tables created by the PAC and incorporating those tables into the permanent update table. This process would generally be run automatically at regular times of day or intervals. The desired times of day or intervals are specified in the DAS initialization file and will generally vary by site. When the DAS is running, it will schedule and run the update process at the specified times of day or intervals. If desired, however, the update process can also be started manually. Figure 18 shows the update pull-down menu for the DAS. The user does not need to login to the database to manually start the update function.

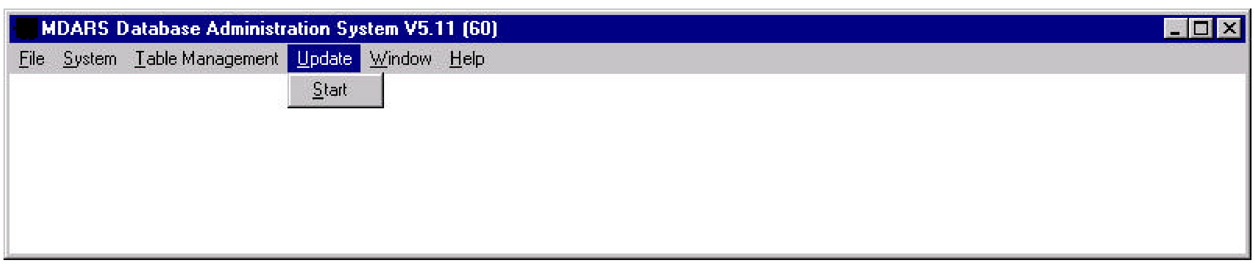



**Figure 18. Update Pull-Down Menu (DAS)**

#### 7.3.2.3.2 Table Management

Although it would generally be an unusual situation, there may be a need to clear some or all of the database tables of their contents. In this case, the database administrator could login to the MDARS database (via the system menu) and perform a table management function. Figure 19 shows the System pull-down menu.

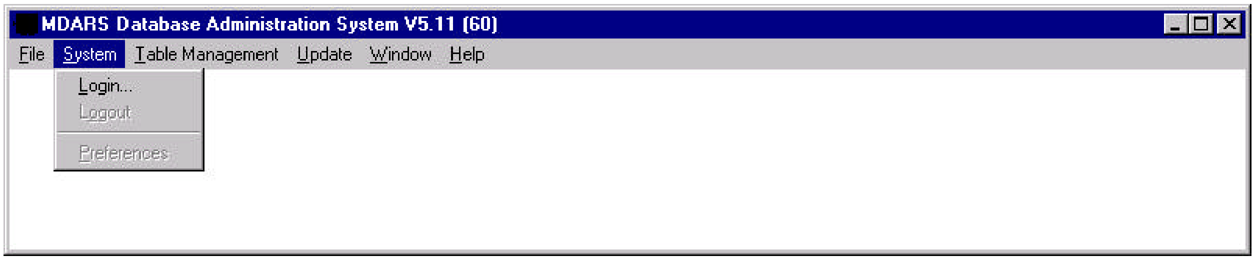



**Figure 19. System Pull-Down Menu (DAS)**

Once logged in, the database administrator can access the Clear Tables menu item under the Table Management pull-down menu. Clear Tables allows the user to select any or all of the user table, the update table and any current survey tables to be cleared (deleted in the case of the survey tables). Figure 20 shows the Clear Tables selection window.

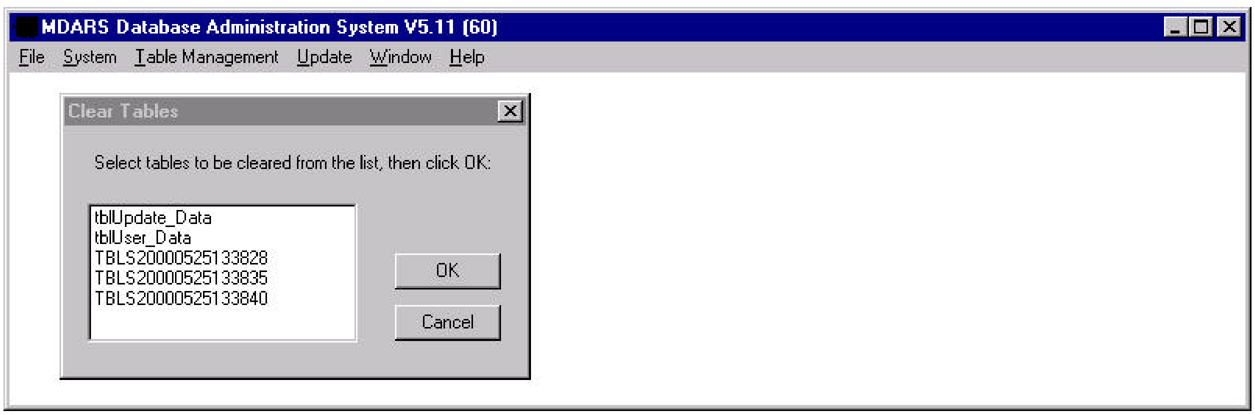



**Figure 20. Table Management/Clear Tables Function (DAS)**

#### 7.3.2.3.3 DAS Current Status

DAS development has reached the completion of Category E-III capabilities, as outlined in Appendix B.

Current documentation for the DAS is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the DAS CSCI of the MDARS MRHA (SSC SD, 2000)

### *7.4 User Access Subsystem - Database Access Computers (DACs)*

The User Access Subsystem allows users to access the MDARS Database via a windows-style graphical user interface on the Database Access Computer (DAC). Future interface of the PAS with the Distribution Standard System (DSS) is being investigated and is planned for Category IV.

### 7.4.1 DAC Functions

Multiple DACs may be attached to the Product Assessment System (PAS). These computers can be desktop or laptop systems.

They are all used by inventory management personnel to ship and receive tagged items, produce reports, locate tagged items and perform data entry.

The functions of a DAC are:

- **Inventory Management:** users can add, modify and delete data from the database. Inventory management can be done semi-automatically using a combination of bar code scanner, modified tag reader and typing of information, or manually by typing information.
- **Reporting:** users can generate item reports and locate items on site maps.
- **Product Location:** users can view the expected and the assessed locations of specific items in both map and text format and can cause tags to beep (if tags are able to beep).
- **Monitoring Update Status:** users can monitor the status of the Database Administration System (DAS) update process.

#### *7.4.1.1 Inventory Management, Reporting, Product Location and Update Status Functions*

Figure 21 shows the main screen for the DAC. Users select functions from windows-style pull-down menus using keyboard hot-keys, a mouse, or a combination of the two. Figure 22 shows the pull-down menu for the System functions.

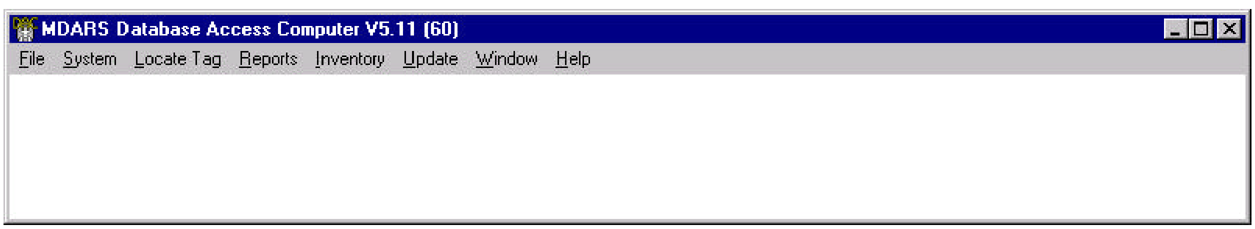



**Figure 21. Main Screen (DAC)**

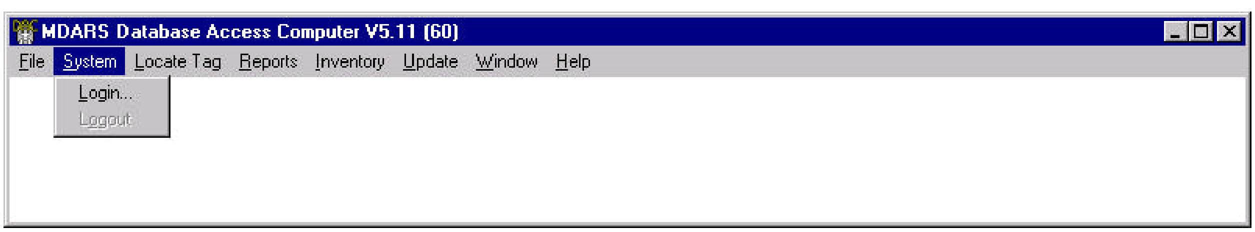



**Figure 22. System Pull-Down Menu (DAC)**

The user has access to context sensitive help using the various Help menu functions. Figure 23 shows the pull-down menu for Help.


**Figure 23. Help Pull-Down Menu (DAC)**

### 7.4.1.1.1 Inventory Management

Information on tags in the database can be added, deleted or modified using two different basic methods. One method is using "launch station" features of the DAC; the other is manual data entry. A launch station is a DAC equipped with a bar code reader capable of reading at least Code 39 bar codes (a Symbol Corporation Model LS-3603-1200A reader is currently used), a modified tag reader compatible with the tags which are being attached to items, and a keyboard for typed entries. The bar code reader and tag reader are connected to serial communications ports on the DAC. A DAC equipped as a launch station allows warehouse personnel to process tagged items into and out of the warehouse with just a few steps and a minimum of typing. Manual data entry is done strictly by typing information on tags and their associated inventory items into an on-screen data entry form. The DAC Inventory menu, shown in Figure 24 provides both launch station and manual data entry features.

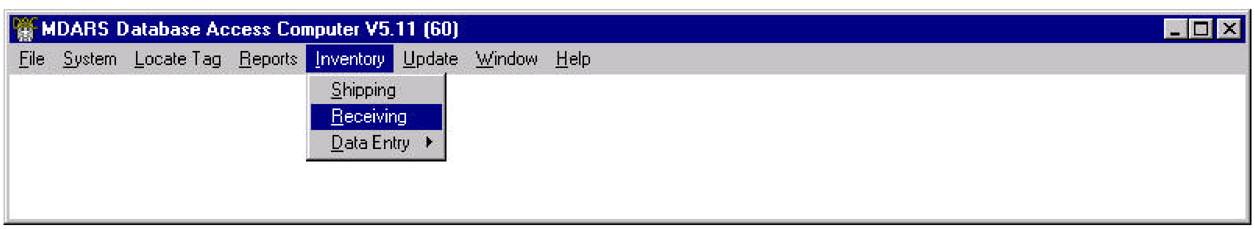



**Figure 24. Inventory Pull-Down Menu (DAC)**

#### 7.4.1.1.1.1 Launch Station Features

A DAC equipped as a launch station allows a user to semi-automatically process tagged items into and out of the database. The Inventory menu is used for these functions. Inventory menu launch station sub-options are:

Receiving - used to semi-automatically add new tag IDs to the database.
Shipping - used to semi-automatically remove tag IDs from the database.

When a user selects the Inventory/Receiving menu option, a message box appears asking the user to place the tag that is to be attached to the inventory on the launch station. Once the user presses the OK button, the tag reader reads the tag IDs from any tags within its range. Assuming only one tag ID is read, the user is prompted to enter the rest of the information to be associated with that tag. If multiple tags are read, the user is prompted for which tag ID to use before proceeding. Figure 25 shows the form that appears in which the user will enter the remaining information. The Container ID field shown in the form can be filled in by scanning the bar code on the container to which the tag will be attached. Once the bar code is scanned, the scanned alphanumeric characters will be automatically entered into the field. The rest of the form is to be filled in manually.

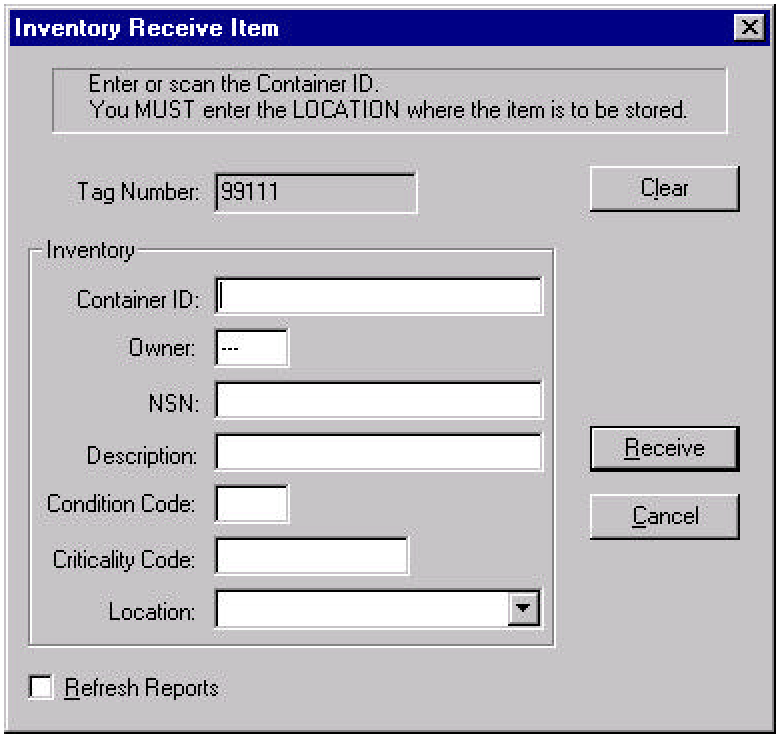



**Figure 25. Inventory/Receiving Function (DAC)**

When a user selects the Inventory/Shipping menu option, a message box appears prompting the user for the Container ID of the item to be shipped. The Container ID can be entered by scanning the bar code on the container. Once the user presses the OK button, a form appears, shown in Figure 26, which contains several pieces of information about the container and its associated tag. The user is asked to validate the removal of the item from the database.

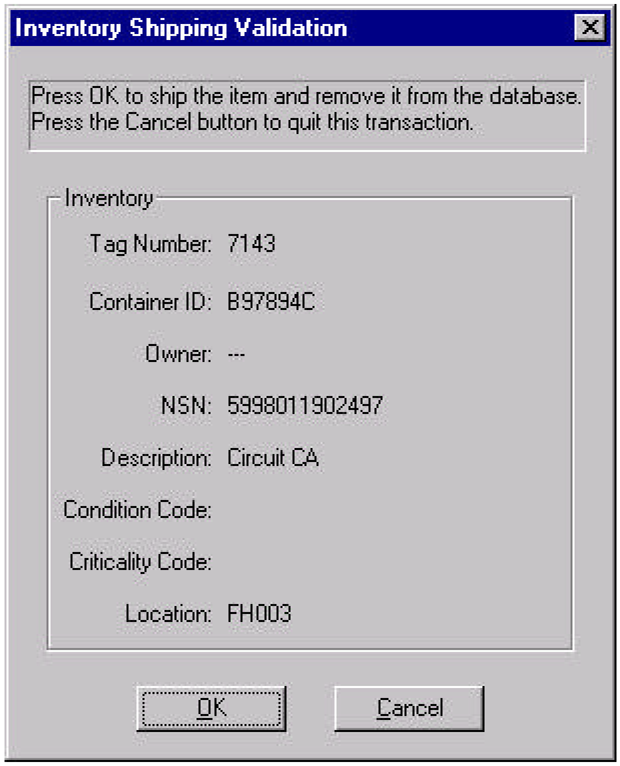



**Figure 26. Inventory/Shipping Validation Form (DAC)**

7.4.1.1.1.2 Manual Data Entry

Manual entry and editing of data in the database is done using functions under the Inventory/Data Entry menu option. Data Entry menu sub-options are:

Add Item - used to add new tag IDs to the database.
Edit Item - used for manual update of data associated with a tag ID.
Delete Item - used to remove tag IDs from the database.

Figure 27 shows the screen layout for the Add Item function, which is similar to the layout for all of these functions.

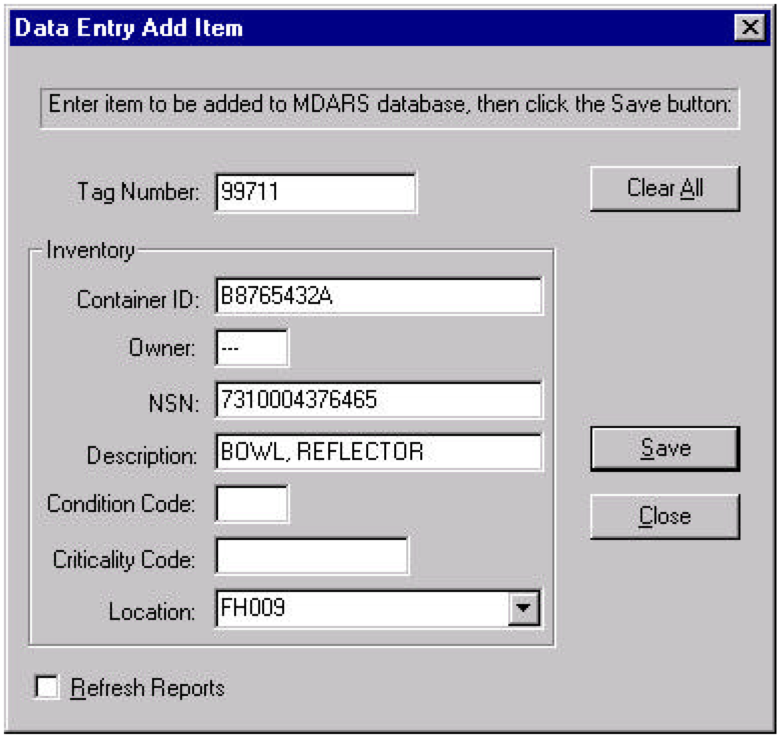
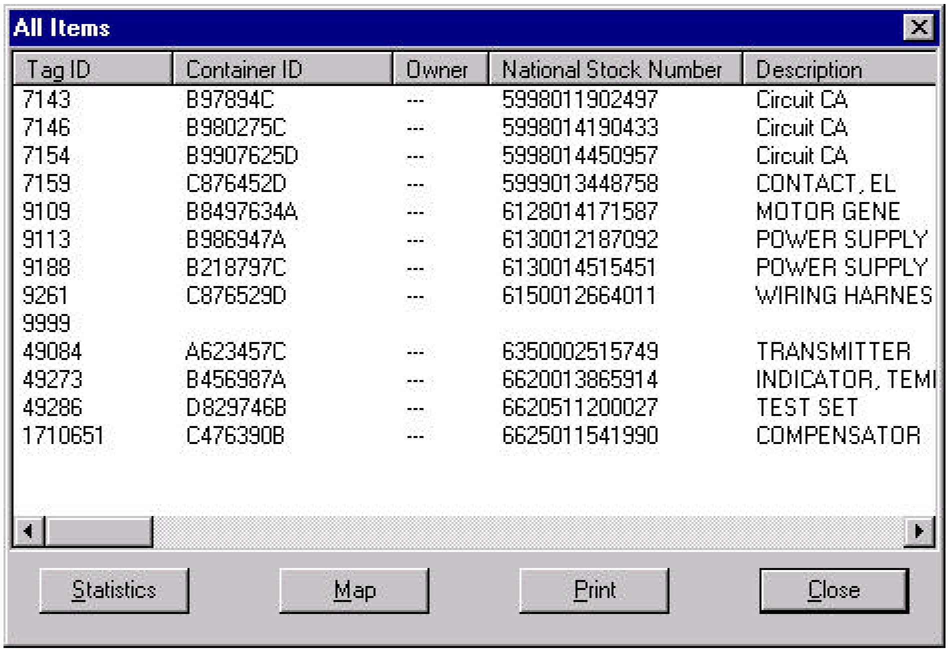



**Figure 27. Inventory/Data Entry/Add Item Function (DAC)**

7.4.1.1.2 Reports

Currently, some standard reports are available under the Reports menu option for querying and displaying all database inventory and database inventory exceptions. Exception reports display conditions which are considered unusual and which may need follow-up action by a human. Ad hoc query capability is planned for Category IV and will be implemented under a Customize menu option. Currently, available reports are:

- All Items Report: report of all tag IDs in the database, as well as their related information. Figure 28 shows a portion of a sample All Items report. Other report formats are similar.
- Found Items Report: report of tag IDs found by platforms where the tag IDs are not already entered in the tag-id table.
- Missing Items Report: report of any tag IDs not located by any platform within a defined time period (e.g. 24 hours).
- Moved Items Report: report of any tag IDs located by any platform at a greater than allowable distance from their expected location(s).



All Items

| Tag ID | Container ID | Owner | National Stock Number | Description |
|---|---|---|---|---|
| 7143 | B97894C | --- | 5998011902497 | Circuit CA |
| 7146 | B980275C | --- | 5998014190433 | Circuit CA |
| 7154 | B9907625D | --- | 5998014450957 | Circuit CA |
| 7159 | C876452D | --- | 5999013448758 | CONTACT, EL |
| 9109 | B84976334A | --- | 6128014171587 | MOTOR GENE |
| 9113 | B986947A | --- | 6130012187092 | POWER SUPPLY |
| 9188 | B218797C | --- | 6130014515451 | POWER SUPPLY |
| 9261 | C876529D | --- | 6150012664011 | WIRING HARNES |
| 9999 | | | | |
| 49084 | A623457C | --- | 6350002515749 | TRANSMITTER |
| 49273 | B456987A | --- | 6620013865914 | INDICATOR, TEMI |
| 49286 | D829746B | --- | 6620511200027 | TEST SET |
| 1710651 | C476390B | --- | 6625011541990 | COMPENSATOR |



**Figure 28. Sample All Items Report (DAC)**

7.4.1.1.2.1 Database Reports Locate

Site mapping of an entire report is available (via the Map button on the report) for all available reports. This brings up a map with all tags in the report identified on the map.

### 7.4.1.1.3 Product Location

The DAC can be used to help locate any individual tag. The Locate Tag menu, shown in Figure 29, provides one sub-options:

Map - used to display both the expected and assessed locations of a tag ID in map and text form. Figure 30 shows a location map for a tag ID.

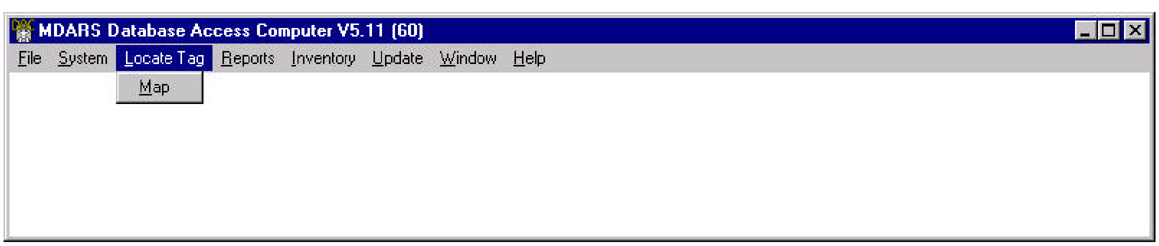



**Figure 29. Locate Tag Pull-Down Menu (DAC)**

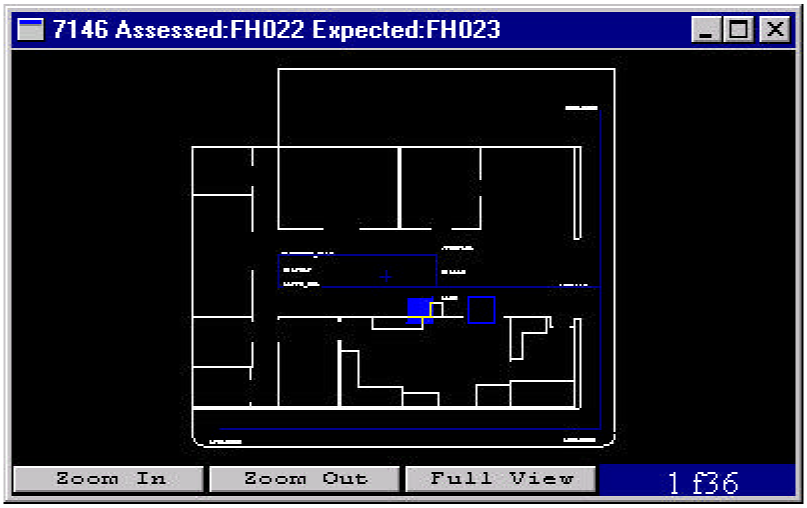



**Figure 30. Locate Map Display (DAC)**

7.4.1.1.4 Update Status

A user can monitor the DAS update process via the Update Status menu option on the DAC. If an update is currently in progress, this status window will indicate the phase and percentage complete of the update process. Otherwise, the status window displays the data and time of the last update and related statistics.

### 7.4.2 DAC Current Status

DAC development has reached the completion of Category E-III capabilities, as outlined in Appendix B.

For Category E-IV, the user interface will be similar to that used in Category III but with added capabilities (Reports Ad Hoc Queries and interface with the Distribution Standard System).

Current documentation for the DAC is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the DAC CSCI of the MDARS MRHA (SSC SD, 2000)

# 8. MDARS Support Program

As part of the MRHA, the MDARS Support Program (MSP) is an executable program that runs as a dedicated process. The MSP provides a DTMF (Dual Tone Multi-Frequency) phone user interface and robot video tracking using stationary warehouse cameras.

The MSP executable program is loaded onto the MSP Hardware Configuration Item (HWCI) that is physically connected to the remaining MRHA HWCIs via an Ethernet LAN. Interprocessor communications are carried out over the LAN.

## *8.1 MSP Functions*

The following general functions have been identified for the MSP CSCI:

- Initialization
- Phone User Interface
- Video Switching

These will be discussed in the following subsections. Specific functionalities falling under these general function areas are presented in Appendix B.

### 8.1.1 Initialization Functions

The MSP must perform the following operations before it is available as an MRHA resource:

- Process command-line arguments (i.e., these typically control debug mode and packet logging).
- Read the initialization file containing site-specific parameters.
- Connect to other computers on the MRHA network.

### 8.1.2 Phone User Interface

The DTMF phone user interface supports both incoming phone calls (requests for robot status or for MSP configuration changes) and outgoing phone calls (emergency event notification).

### 8.1.3 Video Switching

The MSP controls all of the active stationary warehouse cameras input into a video switch. The MSP switches to output video from a camera that is covering the selected robot that the video is tracking.

## *8.2 Current Status*

MSP development has reached the completion of Category E-III capabilities (see Appendix B).

Current documentation status for the MSP is as follows:

Interface Design Document for the MDARS MRHA (SSC SD, 2000)

Design Document for the MSP CSCI of the MDARS MRHA (SSC SD, 2000)

# 9. Local Area Network

A high-speed local area network (LAN) is used as the command, control, and communications link between the distributed computing systems of the host. Based loosely upon the ISO OSI model (reference, date), the LAN consists of the network interface hardware and the supporting software layers as shown in Figure 31. The physical layers are implemented using Ethernet hardware configured in a bus topology. Ethernet provides a 10-Mbps network and is widely supported.

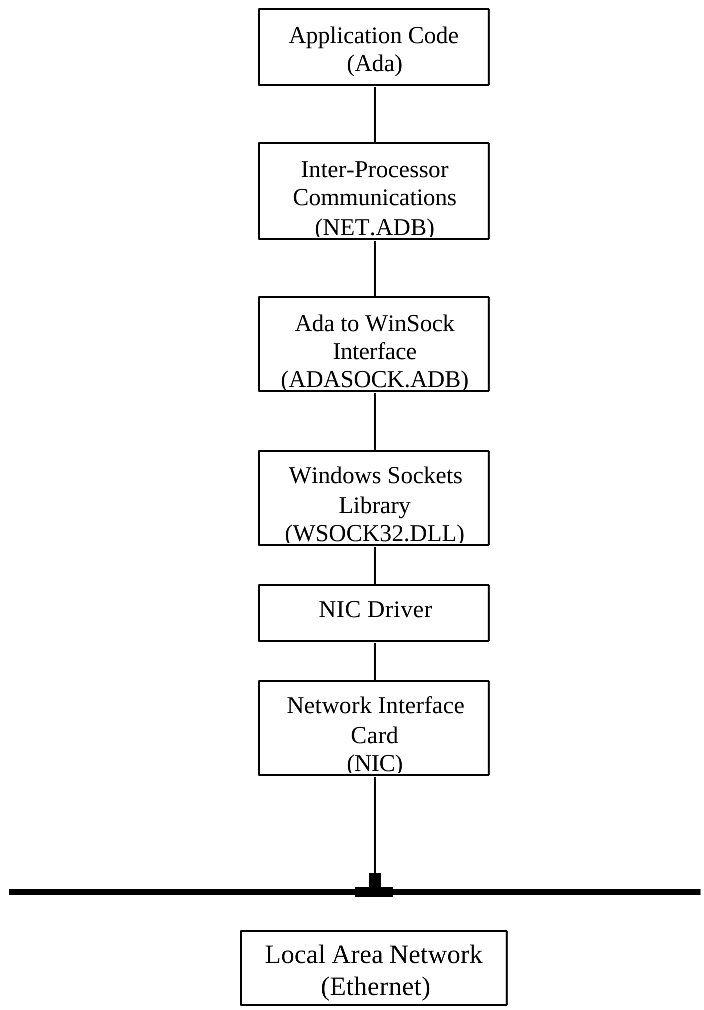



**Figure 31. Block Diagram of Local Area Network Communications Interface**

The network software layers will provide basic communication services between distributed processors from the higher-level programming languages (i.e., Ada) using the TCP/IP protocol. TCP/IP, an established industry standard and an integral part of the Windows NT operating system, provides peer-to-peer communication services as callable library functions that can be interfaced to Ada via a pragma directive. TCP/IP also provides hardware independence through the use of a common low-level Windows socket.

## 10. References

Albert, A. E., "The effect of graphic input devices on performance in a cursor positioning task," Proceedings of the Human Factors Society 26th Annual Meeting, Santa Monica, CA, pp. 54-58 (1982).

Borenstein, J., Everett, H.R., Feng, L., *Navigating Mobile Robots: Systems and Techniques*, A.K. Peters, Ltd., Wellesley, MA, 1996.

Brooks, R.A., "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Trans. on System, Man, and Cybernetics, Vol. SMC-13, No. 3 (1983).

Computer Sciences Corporation, "Software Development Plan for the Supervisor Computer Software Configuration Item of Mobile Detection, Assessment, and Response System (MDARS)," Contract No. DAA807-90-A035, CDRL #1, Task Order Number 4-92C, Task Title: MDARS Multiple Robot Prototype, May 14, (1992a).

Computer Sciences Corporation, "Software Design Document for the Supervisor Computer Software Configuration Item of Mobile Detection, Assessment, and Response System (MDARS)," Contract No. DAA807-90-A035, CDRL #2, Task Order Number 4-92C, Task Title: MDARS Multiple Robot Prototype, September 14, (1992b).

Computer Sciences Corporation, "Interface Design Document for the Supervisor Computer Software Configuration Item of Mobile Detection, Assessment, and Response System (MDARS)," Contract No. DAA807-90-A035, CDRL #2 (Continued), Task Order Number 4-92C, Task Title: MDARS Multiple Robot Prototype, Nov 20, (1992c).

Computer Sciences Corporation, "Software Development Plan for the Operator Display Computer Software Configuration Item," Purchase Order No. WFP-55-92, CDRL # A001, Task Title: MDARS Multiple Robot Prototype, Dec 31, (1992d).

Computer Sciences Corporation, "Software Development Plan for the Planner/Dispatcher Computer Software Configuration Item," Purchase Order No. WFP-55-92, CDRL # A002, Task Title: MDARS Multiple Robot Prototype, Dec 31, (1992e).

Computer Sciences Corporation, "Software Development Plan for the Link Server Computer Software Configuration Item," Purchase Order No. WFP-55-92, CDRL # A003, Task Title: MDARS Multiple Robot Prototype, Dec 31, (1992f).

Computer Sciences Corporation, "Detail Design Document for the Supervisor Computer Software Configuration Item of Mobile Detection, Assessment, and Response System (MDARS)," Contract No. DAA807-90-A035, CDRL #2, Task Order Number 4-92C, Task Title: MDARS Multiple Robot Prototype, April, (1993a).

Everett, H. R., "Security and Sentry Robots", in INTERNATIONAL ENCYCLOPEDIA OF ROBOTICS APPLICATIONS AND AUTOMATION, ed. Dorf, R.C., John Wiley and Sons, March (1988).

Everett, H.R., Gilbreath, G.A, Tran, T.T., Nieusma, J.M., "Modeling the Environment of a Mobile Security Robot," NOSC Technical Document 1835, Naval Ocean Systems Center, San Diego, CA, (1990).

Everett, H.R., Sensors for Mobile Robots: Theory and Application, ISBN 1-56881-048-2, A.K. Peters, Ltd., Wellesley, MA, June, 1995.

Everett, H.R., Laird, R.T., Gilbreath, G.A., Heath-Pastore, T.A., “Technical Development Strategy for the Mobile Detection Assessment and Response System (MDARS),” NCCOSC Technical Note 1776, Naval Command Control and Ocean Surveillance Center, RDT&E Division, San Diego, CA, August, 1996.

Fryxell, R.C., "Navigation Planning Using Quadtrees," Mobile Robots II, W.J. Wolfe, W.H. Chun, Eds., Proc. SPIE 852, pp. 256- 261 (1988).

Gilbreath, G.A., "Software Test Description for the Planner/Dispatcher CSCI of the Mobile Detection Assessment Response System," MDARS-PLANNER-STD-CAT-ROC1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 6 July 1993.

Grant, K.J., "Software Test Description for the Supervisor CSCI of the Mobile Detection Assessment Response System," MDARS-SUPER-STD-CAT-ROC1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 6 July 1993.

Heath-Pastore, T.A, "Software Test Description for the Operator CSCI of the Mobile Detection Assessment Response System," MDARS- OPERATOR-STD-CAT-ROC1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 6 July 1993.

Helander, M. (ed.), "Handbook of Human-Computer Interaction," Elsvier Science Publishing Company, Inc., New York, NY, pp. 495- 519 (1988).

Holland, J., Everett, H.R., Gilbreath, G.A., "Hybrid Navigational Control Scheme for Autonomous Platforms," SPIE Mobile Robots V, (1990).

Karat, J., McDonald, J. E., and Anderson, M., "A Comparison of Menu Selection Techniques: Touch Panel, Mouse, and Keyboard," International Journal of Man-Machine Studies, Vol. 25, pp. 73-78 (1986).

Laird, R.T., Everett, H.R., Gilbreath, G.A., "A Host Architecture for Multiple Robot Control," ANS Fifth Topical Meeting on Robotics and Remote Handling, Knoxville, TN, April, (1993a).

Laird, R.T., Gilbreath, G.A., Heath-Pastore, T.A., Smurlo, R.T., Tran, T.A., "Software Test Plan for the Mobile Detection Assessment Response System," MDARS-MRHA-STP-CAT-ROC1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 6 July 1993.

Laird, R.T., "Software Test Description for the Link Server CSCI of the Mobile Detection Assessment Response System," MDARS- LINKSERV-STD-CAT-ROC1, Naval Command, Control and Ocean Surveillance Center, San Diego, CA, 6 July 1993.

Lee, C.Y., "An Algorithm for Path Connections and its Applications," IRE Transactions on Electronic Computers, Vol. EC-10, September, pp. 346-365 (1961).

Lozano-Perez, T., and M.A. Wesley, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Communications of the ACM, Vol. 22, No. 10, pp.560-570 (1979).

Moravec, H.P., "Certainty Grids for Mobile Robots," Proceedings of the Workshop on Space Telerobotics, JPL, Pasadena, CA, January (1987).

Rubin, F. "The Lee Path Connection Algorithm," IEEE Transactions on Computers, Vol. C-23, No. 9, September, pp. 907-914 (1974).

Sears, A. and Shneiderman, B., "High Precision Touchscreens: Design Strategies and Comparisons with a Mouse," International Journal of Man-Machine Studies, Vol. 34, pp. 593-613 (1991).

Smurlo, R.P., Everett, H.R., "Intelligent Security Assessment for a Mobile Robot", Sensors Expo, Chicago, September, (1992).

Winston, P.H., ARTIFICIAL INTELLIGENCE, Addison-Wesley, Reading, MA (1984).

# 11. Bibliography

Everett, H.R., "A Computer Controlled Sentry Robot", ROBOTICS AGE, March/April, 1982.

Everett, H.R., "A Microprocessor Controlled Autonomous Sentry Robot", Master's Thesis, Naval Postgraduate School, October, 1982.

Everett, H.R., "Summary of Navy Applications of Robotics Sponsored by the Naval Sea Systems Command", ASME Computer Technology Committee, CAD/CAM Meeting, San Antonio, TX, 17-18 June, 1984.

Everett, H.R., "NAVSEA Integrated Robotics Program: Annual Report", FY-84, NAVSEA Technical Report No. 450-90G-TR-0002, Naval Sea Systems Command, Washington, DC, December 1984.

Everett, H.R., "Robotics in the Navy: Industrial Development Efforts", ROBOTICS AGE, November, 1985.

Everett, H.R., "Robotics in the Navy, Part II: Non-industrial Development Efforts", ROBOTICS AGE, December, 1985.

Everett, H.R., "A Second Generation Autonomous Sentry Robot", ROBOTICS AGE, April, 1985.

Everett, H.R., "A Multi-Element Ultrasonic Ranging Array", ROBOTICS AGE, July, 1985.

Everett, H.R., "Robotics Technology: Areas of Needed Research and Development", White Paper presented to ONR/ONT, Ser 90G/119, Naval Sea Systems Command, 6 September 1985.

Everett, H. R., "NAVSEA Integrated Robotics Program: Annual Report, FY 85", NAVSEA Technical Report No. 450-90G-TR-0003, Naval Sea Systems Command, Washington, DC, December 1985.

Everett, H.R., Flynn, A.M., "A Programmable Near-Infrared Proximity Detector for Mobile Robot Navigation", Proceedings, SPIE Mobile Robots I, Cambridge, MA, 26-31 October 1986.

Everett, H.R., "Non-Contact Ranging Systems for Mobile Robots", SENSORS, Vol. 4, No. 4, April 1987, pp. 9-19.

Everett, H.R., Bianchini, G.L., "ROBART II: An Intelligent Security Robot", U.S. Army Training and Doctrine Command, Artificial Intelligence and Robotics Symposium, Norfolk, VA, June 1987.

Everett, H.R., "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots", NOSC Technical Report No. 1194, Naval Ocean Systems Center, San Diego, CA, March 1988.

Everett, H.R., Gilbreath, G.A., Bianchini, G.L., "Robart II: An Autonomous Sentry Robot", Interim Report, FY-87, NOSC Technical Document 1230, Naval Ocean Systems Center, San Diego, CA, March 1988.

Everett, H.R., Gilbreath, G.A., Alderson, S.L., "Intelligent Security Assessment for a Mobile Sentry Robot", Proceedings, 29th Annual Meeting, Institute for Nuclear Materials Management, Las Vegas, NV, June 1988.

Gilbreath, G.A., Everett, H.R., "Path Planning and Collision Avoidance for an Indoor Security Robot", Proceedings, SPIE Mobile Robots III, Cambridge, MA, November 1988.

Everett, H.R., Gilbreath, G.A., "A Supervised Autonomous Security Robot", ROBOTICS AND AUTONOMOUS SYSTEMS, North-Holland, 1988.

Everett, H.R., Gilbreath, G.A., "ROBART II: A Robotic Security Testbed", NOSC Technical Document 1450, Naval Ocean Systems Center, San Diego, CA, January 1989.

Everett, H.R., "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots", ROBOTICS AND AUTONOMOUS SYSTEMS, North- Holland, 1989.

Everett, H.R., Gilbreath, G.A., Gage, D.W., "Environmental Modeling for Mobile Security Robots", CPIA Publication 517,Proceedings, Second Annual Navy IR/IED Symposium, The John Hopkins University APL Laurel, MD., 20707-6099 June 1989.

Moser, J., Everett, H.R., "Wide Angle Optical Ranging System", Proceedings, SPIE Mobile Robots III, Philadelphia, PA, November 1989.

Everett, H.R., Laird, R.T., "Reflexive Teleoperated Control", Proceedings, Association For Unmanned Vehicles Symposium, AUVS- 90, Dayton, OH, July-August 1990.

Everett, H.R., Gilbreath, G.A., Tran, T.T., "Modeling the Environment of a Mobile Security Robot", NOSC Technical Document 1835, Naval Oceans Systems Center, August, 1990.

Everett, H.R., Gilbreath, G.A., Holland, J.M., "Hybrid Navigational Control Scheme", SPIE Mobile Robots V, November 1990.

Hughes, T.W., Everett, H.R., et al, "Issues in Mobile Robotics: Unmanned Ground Vehicle Program Teleoperated Vehicle", SPIE Mobile Robots V, November 1990.

Everett, H.R., Gilbreath, G.A., Laird, R.T., "Multiple Host Robotic Architecture", NCCOSC Technical Note 1710, February, 1992.

Metz, C.D., Everett, H.R., Myers, S., "Recent Developments in Tactical Unmanned Ground Vehicles", Association for Unmanned Vehicles Symposium, Huntsville, AL, June, 1992.

Everett, H.R., Stitz, E.H., "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots", NCCOSC Technical Report No. 1194, Revision 1, San Diego, CA, draft.

Everett, H.R., DeMuth, D.E., Stitz, E.H., "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots," NCCOSC Technical Report No. 1194, Revision 1, Naval Command Control and Ocean Surveillance Center, RDT&E Division, San Diego, CA, December, 1992.

Smurlo, R.P., Everett, H.R., "Intelligent Security Assessment for a Mobile Robot,"Sensors Expo, Chicago, IL, pp. 125-132, September, 1992.

# 12. Appendix A

**TN-1710/TD-3026 DOCUMENT CONFIGURATION CONTROL**

| | |
|---|---|
| <u>21 November 91</u> | "Multiple Robot Host Architecture", Preliminary Draft. |
| <u>12 December 91</u> | "Multiple Robot Host Architecture", NOSC Technical Note 1710, Draft. |
| <u>18 February 92</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710. |
| <u>1 November 92</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 1, Draft. |
| <u>09 February 93</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 1. |
| <u>21 April 93</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 2, Draft. |
| <u>08 July 93</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 2, 2nd Draft |
| <u>20 December 93</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 2, 3rd Draft. |
| <u>01 April 94</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 2. |
| <u>22 August 96</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 3, Preliminary Draft. |
| <u>15 October 96</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 3, Final Draft. |
| <u>01 April97</u> | "Multiple Robot Host Architecture", NRaD Technical Note 1710, Revision 3. |
| <u>21 January 98</u> | "Multiple Resource Host Architecture", NRaD Technical Note 1710 Revision 4. |
| <u>August 98</u> | "Multiple Resource Host Architecture", SSC San Diego, Technical |

Document 3026.

# 13. Appendix B

## SYSTEM FUNCTIONAL REQUIREMENTS

Required functions for all system Computer Software Configuration Items (CSCIs) are normalized with respect to the Supervisor into the following nine categories:

| Category | Milestone | DOS Target Date | NT Target Date |
|---|---|---|---|
| I | Prototype Development | 7/12/93 | 8/26/96 |
| II | F-36 Test | 6/13/94 | 8/26/96 |
| III | Camp Elliott Test | N/A | 1/27/97 |
| IV | Early User Assessment (EUA) | N/A | 4/01/97 |
| E-I | Category E-I Test | N/A | 5/26/99 |
| E-II | Category E-II Test | N/A | 11/8/99 |
| E-III | Category E-III Test | N/A | 2/10/00 |
| E-IV | Category E-IV Test | N/A | TBD |
| P3I | Pre-Planned Program Improvements | N/A | TBD |

The following codes are employed at the end of each functionality to denote development status:

| | |
|---|---|
| NS | Not Started |
| IP | In Progress |
| CP | Completed |

| Category | Major Function | # | SUPERVISOR* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| I | Initialization | 1 | Display program usage and version information when the program is started with "-h" command line parameter | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Read configuration file on startup for site-specific parameters | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Initialize from cold start to on-line configuration | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Display | 1 | Display high level information in form of color-coded icons | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Generally indicate robot heading through icon orientation | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Display prioritized event conditions in the Event Window | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 4 | Display amplifying info and prompts in Supervisor Message Window | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 5 | Depict time of event notification in Event Window | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 6 | Alert guard via Event Window of highest priority need queued | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 7 | Automatically center map display on the priority event | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 8 | Manually center map display on a selected robot | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 9 | Provide a means to select a robot by clicking on displayed icon | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 10 | Provide a means to select a robot from a group listing overlay | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 11 | Provide a means to select a robot from the event window listing | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 12 | Display time of day in Time Window | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 13 | Display date in Date Window | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Command | 1 | Routinely poll the Link Server for updated status information | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 2 | Maintain a blackboard-type data structure representing status | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Provide detailed status information on selected robot | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 4 | Manually assign Planner and Operator resources as directed by guard | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | " | 5 | Maintain assignment status information in blackboard | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 6 | Provide a split-screen display capability for four maps | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 7 | Zoom and Scroll map displays as instructed by guard | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | Event Processing | 1 | Assess status information for exceptional event conditions | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | " | 2 | Prioritize any determined exceptional event conditions | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | " | 3 | Log events as they occur to Supervisor hard drive | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | " | 4 | Automatically assign resources in response to a limited set of prioritized events: random patrol, blocked, trapped, lost | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | " | 5 | Identify available resources on the host LAN | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 6 | Detect non-responsive resource reports | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 7 | Detect Emergency Halt condition and display as non-assignable event | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 8 | Support Manual Assignment for Emergency Halt recovery | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | Housekeeping | 1 | Temporarily store K2A programs downloaded to individual robots | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 2 | Periodically perform time synchronization for all PCs on LAN | SW | CP | 7/9/93 | N/A | N/A |
| I | User Interface | 1 | Allow the use of a Microsoft-Mouse compatible pointing device | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Provide an audible beep to guard as event status changes | SW | CP | 7/9/93 | N/A | N/A |
| I | " | 3 | Activate VCR when video assigned for selected events | SW | CP | 8/15/93 | CP | 5/26/99 |
| I | Diagnostics | 1 | Generate predefined universal health check messages | SW | CP | 7/2/93 | CP | 8/26/96 |
| II | Initialization | 1 | Eliminate command line options for normal operation | SW | CP | 10/29/93 | CP | 8/26/96 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| II | “ | 2 | Read Event information from .INI file. Include Priority, Event Text, and Log to Disk/Screen/Both/Neither | SW | CP | 11/1/93 | CP | 8/26/96 |
| II | “ | 3 | Read Font name, size, and usage settings from the .INI file | SW | CP | 7/2/93 | N/A | N/A |
| II | “ | 4 | Process command line options to turn Network On/Off, and to choose which configuration file to use | SW | CP | 11/1/93 | CP | 8/26/96 |
| II | “ | 5 | Allow sufficient settings in the .INI file to run the Supervisor at different display resolutions without coding changes | SW | CP | 7/2/93 | N/A | N/A |
| II | Display | 1 | Depict IDS threat level in Event Window | SW | CP | 10/21/93 | N/A | N/A |
| II | “ | 2 | Suitably annotate the icon of robot assigned video/audio link | SW | CP | 11/1/93 | CP | 8/26/96 |
| II | “ | 3 | Graphically depict the IDS threat vector for individual icons | SW | CP | 11/1/93 | CP | 8/26/96 |
| II | “ | 4 | Provide icon drawing routines to better display platforms during zoom/scroll operations | SW | CP | 9/10/93 | CP | 8/26/96 |
| II | “ | 5 | Deactivate blank buttons (events with no text, etc.) | SW | CP | 7/2/93 | CP | 8/26/96 |
| II | “ | 6 | Modify Map Drawing package to read and display MDARS Map Format files | SW | CP | 12/1/93 | CP | 8/26/96 |
| II | “ | 7 | Add Scroll Bars to the Map display, and remove Arrow keys from Menu Window. | SW | CP | 12/2/93 | CP | 8/26/96 |
| II | “ | 8 | Modify Robot Status display to more accurately depict the display a guard would use | SW | CP | 5/10/94 | N/A | N/A |
| II | “ | 9 | Process display activity more intelligently; don't turn off mouse cursor if the screen update is away from the mouse cursor location | SW | OH | N/A | CP | 8/26/96 |
| II | Command | 1 | Monitor progress of HWCIs to set reasonable time-outs when we believe a certain operation should be finished, i.e., 30 seconds for a random patrol assignment, etc. | SW | CP | 5/12/94 | CP | 8/26/96 |
| II | Event Processing | 1 | Automatically assign resources in response to prioritized remaining Events: Directed_Survey, Lost_Communications, New_Object_Encountered, Robot_Failed_Diagnostic, Emergency_Halt_Recover | SW | CP | 1/3/94 | CP | 8/26/96 |
| II | “ | 2 | Log appropriate events as they occur to hard copy printer | SW | CP | 4/28/94 | CP | 11/1/96 |
| II | “ | 3 | Provide automatic restore function after an Emergency Halt | SW | CP | 4/28/94 | CP | 8/26/96 |
| II | “ | 4 | Deselect non-functional resources with graceful degradation. Log all HWCI and platform failures | SW | CP | 4/29/94 | CP | 8/26/96 |
| II | “ | 5 | Provide a new Event, STATUS_VERIFY, causing the Supervisor to verify PLATFORM_STATUS requests are sent on schedule. The Event posts at initialization and re-posts each time it is processed | SW | CP | 4/29/94 | CP | 8/26/96 |
| II | “ | 6 | Process limited environmental alarms. | SW | CP | 5/22/94 | CP | 8/26/96 |
| II | “ | 8 | Provide support for SPI module. | SW | CP | 5/22/94 | CP | 1/27/97 |
| II | “ | 9 | Handle new CSCI-Completion Status codes. | SW | CP | 2/1/95 | CP | 8/26/96 |
| II | “ | 10 | Handle new Operator Station return codes, Disable and Ignore. | SW | IP | N/A | N/A | N/A |
| II | Housekeeping | 1 | Provide for limited canned-path execution - inventory mode | SW | CP | 5/18/94 | CP | 8/26/96 |
| II | “ | 2 | Perform configuration file management for individual robots | SW | CP | 3/11/94 | CP | 8/26/96 |
| II | “ | 3 | Verify a platform's health if no status is received for a platform and mark the platform as off-line, if necessary (only get status for health platforms). Check the platform status information for off-line platforms, but do not generate further Events | SW | CP | 6/1/94 | CP | 8/26/96 |
| II | “ | 4 | Check HWCI health only on valid connections | SW | CP | 3/23/94 | CP | 8/26/96 |
| II | “ | 5 | Log appropriate non-Event information | SW | CP | 4/29/94 | CP | 8/26/96 |
| II | “ | 6 | Generate a video message when video source is changed | SW | CP | 4/26/94 | CP | 8/26/96 |
| II | “ | 7 | Generate an abandon message for timed-out events | SW | CP | 5/1/94 | CP | 8/26/96 |
| II | “ | 8 | Generate a TASK_STATUS message to check on HWCI status | SW | CP | 5/1/94 | CP | 8/26/96 |
| II | User Interface | 1 | Print on demand from the Supervisor event log | SW | CP | 5/2/94 | CP | 8/26/96 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| II | “ | 2 | Provide synthesized voice output to advise/alert guard | SW | CP | 10/3/93 | CP | 8/26/96 |
| II | “ | 3 | Provide an auto-repeat feature for appropriate screen buttons | SW | CP | 5/15/94 | CP | 8/26/96 |
| III | Display | 1 | Use new Platform status bit to display when the platform is actually performing a tag read operation. | SW | NS | N/A | CP | 9/26/96 |
| III | “ | 2 | Read and display tag reader status. | SW | NS | N/A | CP | 9/26/96 |
| III | Command | 1 | Provide limited canned path (script file) execution | SW | NS | N/A | CP | 11/27/96 |
| III | Event Processing | 1 | Log all platform events, not just assigned events | SW | NS | N/A | CP | 9/26/96 |
| III | “ | 2 | Support on-demand printing for events. | SW | N/A | N/A | CP | 9/26/96 |
| III | “ | 3 | Invoke recall when CS fails. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 4 | Plan to nearest node when Operator returns halted non-resumable platform. | SW | N/A | N/A | CP | 1/27/97 |
| III | Housekeeping | 1 | Convert S/W to Windows NT Operating System to alleviate memory restrictions. | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 2 | Assign new Planner to Operator upon request. | SW | N/A | N/S | CP | 1/27/97 |
| IV | Initialization | 1 | Automatically reference robots at charger on startup if at dock. | SW | NS | N/A | NS | |
| IV | Display | 1 | Display one-line help message in help bar when mouse moves over menu buttons. | SW | NS | N/A | NS | |
| IV | “ | 2 | Provide capability to control/display maps of up to 8 robots. | SW | NS | N/A | CP | 5/26/99 |
| IV | Event Processing | 1 | Display multiple events per platform | SW | NS | N/A | NS | |
| IV | “ | 2 | Allow assignment of Operator Station without Planner; assign Planner later. | SW | NS | N/A | NS | |
| IV | “ | 3 | Generate messages for Operator when higher priority event is waiting. | SW | NS | N/A | NS | |
| IV | “ | 4 | Log pass-through events/conditions from other CSCIs. | SW | NS | N/A | CP | 1/27/97 |
| IV | Housekeeping | 1 | Provide support for multiple Link Servers. | SW | NS | N/A | CP | 5/26/99 |
| IV | “ | 2 | Provide full canned-path execution. | SW | NS | N/A | NS | |
| IV | “ | 3 | Provide support for multiple Operator Stations | SW | NS | N/A | NS | |
| IV | “ | 4 | Degrade gracefully/recover when another CSCI or LAN cable is disconnected | SW | NS | N/A | CP | 5/26/99 |
| IV | “ | 5 | Provide on-line (windows) help capability. | SW-3.5w* | NS | N/A | CP | 7/15/97 |
| IV | User Interface | 1 | Provide full printing capability. | SW-2w | NS | N/A | NS | |
| IV | “ | 2 | Provide on-line (Windows) help capability | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Initialization | 1 | Recognize platform type | SW | NS | N/A | CP | 5/26/99 |
| E-I | “ | 2 | Initialize platform | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Command | 1 | Process emergency halt | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Event Processing | 1 | Assign resource for events | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Housekeeping | 1 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-I | “ | 2 | Support random patrols | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Display | 1 | Display location for X/Y | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Command | 1 | Support camera select | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Housekeeping | 1 | Add file names to INI file | SW | N/A | N/A | CP | 11/8/99 |
| E-III | Command | 1 | Support degraded operation | SW | N/A | N/A | CP | 2/10/00 |
| E-IV | Display | 1 | Support exterior map format | SW | N/A | N/A | NS | |
| E-IV | “ | 2 | Support map layers | SW | N/A | N/A | CP | 5/26/99 |
| E-IV | “ | 3 | Display stored item description | SW | N/A | N/A | NS | |
| E-IV | Command | 1 | Support lock get data | SW | N/A | N/A | IP | |
| E-IV | User Interface | 1 | Support enhanced path scripting | SW | N/A | N/A | NS | |
| P3I | Initialization | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Display | 1 | ICIDS integration | SW | NS | N/A | NS | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| P3I | Command | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Event Processing | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Housekeeping | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | User Interface | 1 | ICIDS integration | SW | NS | N/A | NS | |

** POC for the Supervisor is Kelly Grant - (619) 553-0850*

| Category | Major Function | # | PLANNER* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| I | Initialization | 1 | Process version and program usage options from command line | SW | CP | 7/9/93 | CP | 6/3/96 |
| I | " | 2 | Process and display configuration file information prior to network initialization | SW | CP | 7/9/93 | CP | 6/3/96 |
| I | " | 3 | Connect to other computers on the network during system startup | SW | CP | 7/2/93 | CP | 6/17/96 |
| I | Random Patrols | 1 | Direct the platform to random virtual nodes | SW | CP | 7/2/93 | CP | 6/17/96 |
| I | " | 2 | The platform will randomly pause at virtual nodes to enter Survey mode | SW | CP | 7/10/93 | CP | 6/17/96 |
| I | Obstacle Avoidance | 1 | In preparation for obstacle avoidance planning, the Planner will update the locations of transient obstacles in the map by incorporating sonar history data from the narrow beam sonar computer | SW | CP | 7/2/93 | CP | 1/27/97 |
| I | " | 2 | An obstacle avoidance path will be planned and executed around objects, guiding the platform to the originally intended destination | SW | CP | 7/2/93 | CP | 1/27/97 |
| I | Directed Movement | 1 | A Reference action will be downloaded to the platform upon receipt of a reference packet from the Operator station | SW | CP | 7/2/93 | CP | 6/17/96 |
| I | " | 2 | When sent to a specific destination by the Operator station, the Planner will not insert random survey stops. | SW | CP | 7/2/93 | CP | 6/17/96 |
| I | Housekeeping | 1 | The Planner will have the ability to send the current path information to the Supervisor. | SW | CP | 7/10/93 | CP | 6/17/96 |
| I | Diagnostics | 1 | The Planner will respond to Health Check packets | SW | CP | 7/9/93 | CP | 6/3/96 |
| II | Initialization | 1 | Eliminate command line operations for normal operation | SW | CP | 12/5/93 | | |
| II | Random Patrols | 1 | Patrol only to nodes with random bit set | SW | CP | 6/6/94 | CP | 6/17/96 |
| II | " | 2 | Provide support for "LARGE" maps | SW | CP | 6/6/94 | CP | 6/3/96 |
| II | Obstacle Avoidance | 1 | Continue to monitor robot during CA maneuver when robot is in selective halt node. | SW | N/A | N/A | CP | 1/27/97 |
| III | Display | 1 | Convert S/W to Windows NT Operating System to alleviate memory restrictions. | SW | NS | N/A | CP | 9/26/96 |
| III | Obstacle Avoidance | 1 | Improve collision avoidance algorithm to model improved sensor suite (i.e. interpret added sensors to detect 8 transducers). | SW | OH | N/A | CP | 9/26/96 |
| III | " | 2 | Implement "plan to nearest point on path" for collision avoidance. | SW | NS | N/A | CP | 1/27/97 |
| III | " | 3 | Support recall plan type. | SW | N/A | N/A | CP | 1/27/97 |
| III | Directed Movement | 1 | Modify Planner to read learning database from robot before downloading new program. | SW | OH | N/A | CP | 8/5/96 |
| III | " | 2 | Support limited mixed virtual and unrestricted paths. | SW | N/A | N/A | CP | 1/27/97 |
| III | " | 3 | Support plan to nearest node plan type. | SW | N/A | N/A | CP | 1/27/97 |
| III | " | 4 | Support interrupted plan type. | SW | N/A | N/A | CP | 1/27/97 |
| III | Housekeeping | 1 | Modify robot commands for new Cybermotion memory maps and computer numbers. | SW | NS | N/A | CP | 6/17/96 |
| III | " | 2 | Support task status. | SW | N/A | N/A | CP | 1/27/97 |
| IV | Random Patrols | 1 | Allow disabling of paths. | SW | NS | N/A | CP | 6/3/96 |
| IV | Obstacle Avoidance | 1 | Accommodate Cybermotion's CIRCUMNAVIGATION. | SW | NS | N/A | CP | 1/27/97 |
| IV | " | 2 | Implement automated recovery routine. | SW | NS | N/A | N/A | N/A |
| IV | Directed Movement | 1 | Mix virtual and unrestricted paths. | SW | NS | N/A | N/A | N/A |
| IV | Housekeeping | 1 | Degrade gracefully/recover when another CSCI or LAN cable is disconnected | SW | NS | N/A | CP | 5/26/99 |
| IV | " | 2 | Determine if any path segment has not been recently traversed | SW | NS | N/A | N/A | N/A |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IV | User Interface | 1 | Provide on-line (windows) help capability | SW | NS | N/A | CP | 5/26/99 |
| E-I | Random Patrols | 1 | Support random patrols | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Directed Movement | 1 | Support directed sends | SW | NS | N/A | CP | 5/26/99 |
| E-I | Housekeeping | 1 | Process platform status | SW | NS | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Diagnostics | 1 | Support debug capability | SW | N/A | N/A | CP | 11/8/99 |
| E-II | “ | 1 | Support diagnostic capability | SW | N/A | N/A | CP | 11/8/99 |
| E-IV | User Interface | 1 | Enhance node selection | SW | N/A | N/A | NS | TBD |
| P3I | Initialization | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | Display | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | Random Patrols | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | Housekeeping | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | User Interface | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | Diagnostics | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |
| P3I | Diagnostics | 1 | ICIDS integration | SW | NS | N/A | N/A | N/A |

** POC for the Planner is Gary Gilbreath - (619) 553-3669*

| Category | Major Functions | # | OPERATOR STATION* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| I | Initialization | 1 | Display program usage and version information as a command line option | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 2 | Read and process configuration file | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | Display | 1 | Display date and time information in date window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 2 | Display assigned platform status information in status window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 3 | Display map in map window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 4 | Display color coded platform icon in map window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 5 | Display assigned platform identification number in map window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 6 | Display threat vector an alarm state in map window during | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 7 | Display camera field-of-view and bearing icon in map window | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | Command | 1 | Halt the platform, upon user command | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 2 | Resume action after a halt, upon user command | SW | CP | 7/2/93 | CP | 7/15/96 |
| I | “ | 3 | Send a platform to a specified virtual point, upon user command | SW | CP | 7/2/93 | CP | 7/15/96 |
| I | “ | 4 | Initiate a referencing action, upon user command | SW | CP | 7/2/93 | CP | 7/15/96 |
| I | “ | 5 | Place a platform in survey mode, upon user command | SW | CP | 7/2/93 | CP | 8/7/96 |
| I | “ | 6 | Release platform and free Operator Station, upon user command | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | Housekeeping | 1 | Handle assignment from Supervisor | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 2 | Process time synchronization packets from Supervisor | SW | CP | 7/2/93 | N/A | N/A |
| I | User Interface | 1 | Provide user selectable zoom in/zoom out map display | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 2 | Provide user selectable map display scroll | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | “ | 3 | Provide user selectable command buttons | SW | CP | 7/2/93 | CP | 7/1/96 |
| I | Diagnostics | 1 | Process Health Check packets | SW | CP | 7/2/93 | CP | 7/1/96 |
| II | Initialization | 1 | Provide a diagnostics mode and a normal operating mode for the Operator Station. | SW | CP | 7/15/94 | CP | 8/15/96 |
| II | “ | 2 | Eliminate command line options for normal operation | SW | CP | 12/5/93 | CP | 8/15/96 |
| II | Display | 1 | Display node names in the help bar during a send or reference activity | SW | CP | 7/15/94 | CP | 8/15/96 |
| II | “ | 2 | Display MRHA module name and version in the upper center title window | SW | CP | 12/7/94 | CP | 7/1/96 |
| II | “ | 3 | Display camera FOV icon to represent video link assignment | SW | CP | 7/15/94 | N/A | N/A |
| II | “ | 4 | Integrate new map display module (*.lmp files) | SW | CP | 7/15/94 | CP | 7/1/96 |
| II | “ | 5 | Gray command buttons when they are not an appropriate command selection software | SW | CP | 7/15/94 | CP | 8/25/96 |
| II | Command | 1 | Provide release options for off-line and survey | SW | CP | 7/15/94 | CP | 8/25/96 |
| II | “ | 2 | Provide Cancel options for SEND and REFERENCE commands | SW | CP | 8/15/93 | CP | 8/21/96 |
| II | “ | 3 | Provide manual control option (interface with Telereflexive control software) | SW | CP | 7/15/94 | CP | 1/27/97 |
| II | “ | 4 | Provide camera function control (interface with camera control software) | SW | CP | 10/18/94 | CP | 1/27/97 |
| II | “ | 5 | Implement a “smart” resume (check to see that the robot is in a resumable state) | SW | CP | 1/12/94 | CP | 7/15/96 |
| II | “ | 6 | Modify resume for Category II Platform compatibility | SW | CP | 10/18/94 | CP | 7/15/96 |
| II | “ | 7 | Handle collision avoidance maneuvers | SW | CP | 10/18/94 | CP | 8/25/96 |
| II | Housekeeping | 1 | Handle Emergency Halt Recover | SW | CP | 10/18/94 | CP | 8/23/96 |
| II | “ | 2 | Implement Survey mode when initiated by another MRHA module | SW | CP | 7/15/94 | CP | 8/25/96 |
| II | “ | 3 | Handle time-outs in connection with other CSCIs (Planner) | SW | CP | 7/15/94 | CP | 8/25/96 |
| II | “ | 4 | Utilize Platform status bit “path interrupted” to assist with robot state determination. | SW | CP | 10/15/94 | CP | 8/15/96 |
| II | “ | 5 | Handle new Planner Completion Status Codes | SW | CP | 2/1/95 | CP | 8/5/96 |
| II | “ | 6 | Log packets | SW | CP | 7/15/94 | CP | 8/15/96 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| II | “ | 7 | Provide command line option for putting an alternative initialization file name | SW | N/A | N/A | CP | 8/15/96 |
| II | “ | 8 | Display build number in title caption | SW | N/A | N/A | CP | 8/15/96 |
| II | “ | 9 | Support new Operator Assign packet with the Sub_Mode information | SW | N/A | N/A | CP | 8/23/96 |
| II | User interface | 1 | Provide preliminary Telereflexive platform control | SW | CP | 3/29/94 | CP | 1/27/97 |
| II | “ | 2 | Provide preliminary user-selectable camera on/off control | SW | CP | 7/15/94 | N/A | N/A |
| II | “ | 3 | Provide preliminary camera pan, tilt | SW | CP | 7/15/94 | CP | 1/27/97 |
| II | “ | 4 | Provide a command cancel capability | SW | CP | 7/15/94 | CP | 8/21/96 |
| II | “ | 5 | Provide button selection feedback | SW | CP | 7/15/94 | CP | 8/19/96 |
| II | “ | 6 | Provide HMI feedback in line with human response time guidelines | SW | CP | 7/15/94 | CP | 8/19/96 |
| III | Initialization | 1 | Check for access to robot configuration information (maps and databases) during initialization process. | SW | NS | N/A | CP | 8/15/96 |
| III | “ | 2 | Process video link configuration information from Supervisor | SW | NS | N/A | N/A | N/A |
| III | Display | 1 | Implement pop-up message windows for system critical information. | SW | NS | N/A | CP | 8/25/96 |
| III | “ | 2 | Implement pop-up message windows for process status information. | SW | NS | N/A | CP | 8/25/96 |
| III | “ | 3 | Display platform video on Operator display. | SW | NS | N/A | CP | 2/10/00 |
| III | Command | 1 | Provide Release options menu when the robot is released in a resumable state. | SW | NS | N/A | N/A | N/A |
| III | “ | 2 | Provide Release options menu when the robot is released in intruder detection mode. | SW | NS | N/A | CP | 8/25/96 |
| III | “ | 3 | Modify robot commands for new Cybermotion memory maps and computer numbers. | SW | NS | N/A | CP | 8/15/96 |
| III | “ | 4 | Provide stay/retreat options menu when CA attempt fails. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 5 | Request combination plan when appropriate (from X-Y to node). | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 6 | Request interrupted plan when appropriate to a new virtual node target. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 7 | Provide Release options menu when the robot is released in off-line mode. | SW | N/A | N/A | CP | 1/27/97 |
| III | Housekeeping | 1 | Handle K2A E-Stop. | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 2 | Convert S/W to Windows NT Operating System to alleviate memory restrictions. | SW | NS | N/A | CP | 9/26/96 |
| III | “ | 3 | Improve software maintainability by converting software to Ada programming language. | SW | NS | N/A | CP | 9/26/96 |
| III | “ | 4 | Perform automatic node ID if platform stops for any reason and is not at Target node | SW | NS | N/A | CP | 8/15/96 |
| III | “ | 5 | Load applicable database when platform assigned | SW | NS | N/A | CP | 8/15/96 |
| III | “ | 6 | Incorporate video link status communication with Supervisor | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 7 | Request assign of new Planner if Planner unresponsive. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 8 | Inquire and report on Planner task status. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 9 | Support User-generated halt as Selective_Halt mode. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 10 | Auto resume the robot if released in a resumable state. | SW | N/A | N/A | CP | 1/27/97 |
| III | “ | 11 | Incorporate new completion status, plan type, and event codes. | SW | N/A | N/A | CP | 1/27/97 |
| III | User Interface | 1 | Provide on-screen control of camera focus and zoom functions | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 2 | Provide device control of camera pan, tilt, center functions | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 3 | Provide device control of Telereflexive functions | SW | NS | N/A | CP | 1/27/97 |
| III | “ | 4 | Provide Operator Station immunity to superfluous input from the normal operating mode-input devices. | SW | IP | N/A | CP | 1/27/97 |
| III | “ | 5 | Provide Entrance Window to inform guard of reason for assignment and brief description of possible actions. | SW | NS | N/A | CP | 1/27/97 |
| III | Diagnostics | 1 | Handle lost (Network) communications | SW | NS | N/A | CP | 8/25/96 |
| IV | Display | 1 | Provide “balloon help” for all buttons/windows | SW | NS | N/A | N/A | N/A |
| IV | “ | 2 | Display the planned path notifications. | SW | NS | N/A | N/A | N/A |
| IV | “ | 3 | Display higher priority Supervisor messages in the message window | SW | NS | N/A | NS | |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| IV | “ | 4 | Converge on final verbiage for text and status information | SW | NS | N/A | NS | |
| IV | “ | 5 | Implement “help line” window to display information on user controls when the cursor is within control boundaries. | SW | NS | N/A | IP | |
| IV | “ | 6 | Implement site-specific X-Y location look-up table. | SW | NS | N/A | CP | 2/10/00 |
| IV | “ | 7 | Change Operator Station designation to Directed Control Station (DCS). | SW | NS | N/A | NS | |
| IV | “ | 8 | Add Operational Time Remaining Status Element. | SW | NS | N/A | CP | 2/10/00 |
| IV | Command | 1 | Incorporate Request Platform function. | SW | NS | N/S | NS | |
| IV | Housekeeping | 1 | Pass information to Supervisor for logging. | SW | NS | N/A | CP | 5/26/99 |
| IV | “ | 2 | Request and upload paths from Planner | SW | NS | N/A | N/A | N/A |
| IV | “ | 3 | Degrade gracefully/recover when another CSCI or LAN cable is disconnected | SW | NS | N/A | CP | 1/27/97 |
| IV | User Interface | 1 | Provide on-line (windows) help. | SW | NS | N/A | CP | 5/26/99 |
| E-I | Command | 1 | Process emergency halt | SW | N/A | N/A | CP | 5/26/99 |
| E-I | “ | 2 | Support directed sends | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Housekeeping | 1 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Display | 1 | Display location for X/Y | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Command | 1 | Support teleoperation | SW | N/A | N/A | CP | 11/8/99 |
| E-II | “ | 2 | Support camera move | SW | N/A | N/A | CP | 11/8/99 |
| E-II | “ | 3 | Support camera control | SW | N/A | N/A | CP | 11/8/99 |
| E-III | Command | 1 | Support degraded operation | SW | N/A | N/A | CP | 2/10/00 |
| E-IV | Command | 1 | Handle tamper alarm | SW | NA | N/A | CP | 4/24/00 |
| E-IV | “ | 2 | Support lock get data | SW | NA | N/A | CP | 4/24/00 |
| E-IV | “ | 3 | Handle low fuel status | SW | NA | N/A | NS | |
| E-IV | Display | 1 | Support Meteor II digital video | SW | NA | N/A | CP | 4/24/00 |
| E-IV | “ | 2 | Support Indigo digital video | SW | NA | N/A | CP | 4/24/00 |
| E-IV | “ | 3 | Support exterior map format | SW | NA | N/A | NS | |
| E-IV | “ | 4 | Support map layers | SW | NA | N/A | NS | |
| E-IV | “ | 5 | Display stored item description | SW | NA | N/A | IP | |
| E-IV | User Interface | 1 | Support two-way audio link | SW | NA | N/A | NS | |
| E-IV | “ | 2 | Enhance node selection | SW | NA | N/A | NS | |
| P3I | Initialization | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Display | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Command | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Housekeeping | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | User Interface | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Diagnostics | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | “ | 2 | Perform detailed platform diagnostic analysis when necessary | SW | NS | N/A | NS | |

** POC for the Operator Station is Kelly Grant- (619) 553-0850*

| Category | Major Functions | # | LINK SERVER* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| I | Initialization | 1 | Provide program version and usage information to the user from the operating system command line through the standard options '-?', '-H', and '-h' | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Read configuration information from a disk file that specifies the physical serial I/O connection for each robot in the system | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Dynamically (at start-up time) determine connectivity of logical robots with physical serial I/O ports | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Display | 1 | Provide debug and maintenance displays for:<br>a) incoming/outgoing platform message and network packet information<br>b) time/date and program version information<br>c) platform communication status<br>d) package initialization debug messages<br>e) user help at each operational level | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Message Routing | 1 | Provide reliable (R/F) communications between the host and each platform in the system (i.e., message re-transmission, re- routing, etc.) | SW | CP | 7/11/93 | CP | 8/26/96 |
| I | " | 2 | Maintain a local (routing table) data structure that specifies connectivity of logical platforms with physical serial I/O ports | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Status Polling | 1 | Maintain a local (blackboard) data structure to hold current status for each platform in the system | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Periodically request status from each platform in the system | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Store status information locally for later retrieval by other computer resources | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Emergency Halt | 1 | Monitor the status of an external switch (emergency halt button) and report activation of the switch to the rest of the system (i.e., H/W emergency halt network message) | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 2 | Command each platform in the system to halt upon detecting the activation of the external switch (emergency halt button) | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Generate S/W emergency halt network message (as determined by network failure) | SW | CP | 7/8/93 | CP | 8/26/96 |
| I | Data Log/Eavesdrop | 1 | Provide data logging capabilities for both external serial I/O and local area network communications traffic | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Housekeeping | 1 | Process time synchronization packets from Supervisor | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 2 | Be capable of communicating between CSCIs on the LAN and remote resources | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Be capable of querying specific robots for their operational status and reporting that information to other CSCIs on the LAN | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 4 | Be capable of determining the "health" of a specific platform and reporting that information to other CSCIs on the LAN | SW | CP | 7/7/93 | CP | 8/26/96 |
| I | " | 5 | Be capable of assigning a video channel to a specific platform and releasing the video channel assigned to a platform, manage platform assignment data | SW | CP | 7/10/93 | CP | 8/26/96 |
| I | User Interface | 1 | Provide an emergency halt switch for activation of the emergency halt function | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Provide support for a debug and maintenance keyboard | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Diagnostics | 1 | Be capable of responding to the pre-defined universal network messages (health check) | SW | CP | 7/9/93 | CP | 8/26/96 |
| I | " | 2 | Provide debugging/monitoring capabilities for both external serial I/O and local area network communications traffic along with operational statistics (e.g., error count, message count) | SW | CP | 7/2/93 | CP | 8/26/96 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| I | " | 3 | Provide built-in diagnostics for each hardware subsystem (e.g., serial I/O subsystem and attached modems) | SW | CP | 7/2/93 | CP | 8/26/96 |
| II | Initialization | 1 | Eliminate command line options for "normal" operation | SW | CP | 12/3/93 | CP | 8/26/96 |
| II | " | 2 | Detect when more than one platform reports the same platform ID | SW | CP | 12/7/93 | CP | 1/2797 |
| II | " | 3 | Pass platform ID information to other CSCIs on LAN | SW | CP | 3/4/94 | CP | 8/26/96 |
| II | Message Routing | 1 | Provide non-lockstep simultaneous communications between multiple (8) robots | SW | IP | 3/29/94 | CP | 8/26/96 |
| II | Status Polling | 1 | Poll only those platforms identified in configuration file | SW | CP | 2/23/94 | CP | 8/26/96 |
| II | Data Log/Eavesdrop | 1 | Log platform message traffic to individual files based upon platform ID | SW | CP | 4/28/94 | CP | 9/26/96 |
| II | " | 2 | Provide filtering capabilities for platform message logging | SW | CP | 5/11/94 | CP | 1/27/97 |
| II | " | 3 | Provide filtering capabilities for network packet logging | SW | CP | 5/11/94 | CP | 1/27/97 |
| II | " | 4 | Add command line option to begin operation with data logging turned on | SW | CP | 3/10/94 | CP | 9/26/96 |
| II | Housekeeping | 1 | Report status only for robots that are identified in system at start-up time | SW | CP | 12/5/94 | CP | 8/26/96 |
| II | " | 2 | Accommodate inventory management functions originating on remote platform | SW | CP | 7/15/94 | N/A | N/A |
| II | User Interface | 1 | Bullet-proof vest for keyboard input | SW | CP | 7/15/94 | CP | 8/26/96 |
| III | Initialization | 1 | Provide support for multiple serial I/O cards (2 cards minimum, 8 robots) | SW | N/A | N/A | CP | 8/26/96 |
| III | Display | 1 | Convert S/W to Windows NT Operating System to alleviate memory restrictions | SW | N/A | N/A | CP | 8/26/96 |
| III | Message Routing | 1 | Verify platform status integrity when received from platform | SW | N/A | N/A | CP | 8/26/96 |
| III | " | 2 | Improve communications throughput by upgrading to new Ethernet-addressable modems | SW | N/A | N/A | CP | 8/26/96 |
| III | " | 3 | Integrate platform communications link with control station remoting electronics | SW | N/A | N/A | CP | 6/1/97 |
| III | " | 4 | Implement Cybermotion "#" abbreviated messages | SW | NS | N/A | N/A | N/A |
| III | Housekeeping | 1 | Degrade gracefully/recover when another CSCI or LAN cable is disconnected | SW | NS | N/A | NS | N/A |
| III | Diagnostics | 1 | Quantify modem communications integrity (i.e., reliability information). | SW | IP | CP | CP | 6/1/97 |
| IV | Status Polling | 1 | Send platform status to each CSCI at regular intervals | SW | NS | NS | NS | |
| IV | User Interface | 1 | Provide on-line (windows) help | SW | NS | N/A | CP | 5/26/99 |
| E-I | Initialization | 1 | Recognize platform type | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 2 | Initialize platform | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Message Routing | 1 | Support DGPS message relay | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Status Polling | 1 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Emergency Halt | 1 | Process emergency halt | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Housekeeping | 1 | Disconnect gracefully | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | NS | N/A | CP | 11/8/99 |
| E-II | Message Routing | 1 | Support camera select | SW | N/A | N/A | CP | 11/8/99 |
| P3I | Initialization | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Display | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | " | 2 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Data Log/Eavesdrop | 1 | Provide sophisticated network packet and platform message filtering (i.e., field filters, log bad data) | SW | NS | N/A | NS | |
| P3I | " | 2 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Housekeeping | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Diagnostics | 1 | ICIDS integration | SW | NS | N/A | NS | |

** POC for the Link Server is Daniel Carroll - (619) 553-7636*

| Category | Major Functions | # | PRODUCT ASSESSMENT COMPUTER* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| II | Initialization | 1 | Provide program and usage information to the user from the operating system command line through the standard options '?', '-H', and '-h' | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 2 | Provide '-d' debug command line option | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 3 | Provide '-n' network present command line option | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 4 | Read and process configuration file | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | Display | 1 | When in debug mode, display appropriate debug messages in debug window | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 2 | When in debug mode, display appropriate LAN traffic information in LAN window | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 3 | When in debug mode, display appropriate tag information in tag window | SW | CP | 7/15/94 | CP | 6/1/97 |
| II | Housekeeping | 1 | Process time synchronization packets from Supervisor | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | " | 2 | Respond to Health Checks from Supervisor | SW | CP | 1/31/94 | CP | 6/1/97 |
| II | Tag Info Processing | 1 | Periodically (continuously) request status information from all robots to determine if tags are available | SW | CP | 7/15/94 | CP | 6/1/97 |
| II | " | 2 | Get tag information from TRC when available | SW | CP | 7/15/94 | CP | 6/1/97 |
| II | " | 3 | Provide S/W interface between PAC and Product Database Computer (transaction based database requests) | SW | CP | 7/15/94 | CP | 6/1/97 |
| III | Initialization | 1 | Check to see if Tag Reader Computer is present and responding; if not, continue checking until found (do not send requests to robot that is not responding) | SW | NS | N/A | CP | 6/1/97 |
| III | Display | 1 | Convert S/W to Windows NT Operating System to alleviate memory restrictions | SW | NS | N/A | CP | 6/1/97 |
| III | Housekeeping | 1 | Modify robot commands for new Cybermotion memory maps and computer numbers | SW | NS | N/A | CP | 6/1/97 |
| III | Tag Info Processing | 1 | Modify PAC interface to database to accommodate new Database format | SW | NS | N/A | CP | 6/1/97 |
| III | " | 3 | Write Ada pragmas to C/API and write functions in Ada | SW | NS | N/A | CP | 6/1/97 |
| E-I | Housekeeping | 1 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 2 | Disconnect gracefully | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Command | 1 | Process emergency halt | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Tag Info Processing | 1 | Communicate to 8 robots | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Command | 1 | Support tag get data | SW | N/A | N/A | CP | 11/8/99 |
| E-IV | Tag Info Processing | 1 | Support Spider tag | SW | N/A | N/A | CP | 04/24/99 |
| E-IV | " | 2 | Support production Spider tag | SW | N/A | N/A | NS | TBD |

** POC for the Product Assessment Computer is Daniel Carroll - (619) 553-7636*

| Category | Major Functions | # | DATABASE ADMINISTRATION SYSTEM* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| II | User Interface | 1 | Provide a means for assessing inventory (comparing expected items and detected items) | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | Tag Info Processing | 1 | From tag list, determine and separately store best estimate of each tags X,Y location | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | " | 2 | Utilize database server locking/unlocking mechanisms to allow concurrent access to database by multiple users. | SW | CP | 7/15/94 | CP | 2/7/97 |
| III | Tag Info Processing | 1 | Separate update function and manual inventory maintenance function | SW | N/A | N/A | CP | 2/7/97 |
| III | " | 3 | Provide improved tag localization strategy. | SW/HW | N/A | N/A | CP | 2/7/97 |
| III | " | 4 | Provide logging of user and error messages | SW | N/A | N/A | CP | 2/7/97 |
| E-I | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-IV | Tag Info Processing | 1 | Support production Spider tag | SW | N/A | N/A | NS | |
| E-IV | Tag Info Processing | 1 | Support production Spider tag | SW | N/A | N/A | NS | |
| E-IV | Tag Info Processing | 2 | Improve tag localization scheme | SW | N/A | N/A | NS | |
| P3I | User interface | 1 | Provide administration/maintenance function: Adding/removing users | SW | N/A | N/A | NS | |
| P3I | User Interface | 2 | Provide administration/maintenance function: Changing passwords | SW | N/A | N/A | NS | |
| P3I | User Interface | 3 | Provide administration/maintenance function: Database backups | SW | N/A | N/A | NS | |
| P3I | User Interface | 4 | Provide administration/maintenance function: Database "crash" recovery | SW | N/A | N/A | NS | |

** POC for the Database Administration System is Doriann Jaffee - (619) 553-6915*

| Category | Major Functions | # | DATABASE ACCESS COMPUTER* Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| II | Initialization | 1 | Provide software interface to product database | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | Display | 1 | Provide capability of generating product database reports | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | User Interface | 1 | Allow user to log in to the Product Database Computer | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | " | 2 | Provide pull-down menus and entry screens for manual database data entry and manipulation | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | " | 3 | Provide capability of user to add items to product database | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | " | 4 | Provide capability of user to delete items from product database | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | " | 5 | Provide capability of user to modify (update) items in product database | SW | CP | 7/15/94 | CP | 2/7/97 |
| II | Tag Info Reading | 1 | Utilize database server locking/unlocking mechanisms to allow concurrent access to database by multiple users | SW | CP | 7/15/94 | CP | 2/7/97 |
| III | Display | 1 | Display feedback to user (in the form of a pop-up window) when an attempt is made to access a record, which is locked by another user, and allow user capability to "cancel" the requested operation. | SW | N/A | N/A | CP | 6/4/97 |
| III | User Interface | 1 | Provide on-line help capability | SW | CP | 7/15/94 | CP | 06/11/97 |
| III | Tag Info Processing | 1 | Separate survey data and tag data into separate databases (or database tables) | SW | N/A | N/A | CP | 8/26/96 |
| III | " | 2 | Separate update function and manual inventory maintenance function | SW | N/A | N/A | CP | 2/7/97 |
| III | " | 3 | Shorten lock time-outs and add logic for handling time-outs | SW | N/A | N/A | CP | 6/4/97 |
| III | " | 4 | Provide logging of user and error messages | SW | N/A | N/A | CP | 2/7/97 |
| E-I | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-IV | Tag Info Processing | 1 | Support Spider tag | SW | N/A | N/A | CP | 4/24/00 |
| E-IV | " | 2 | Support production Spider tag | SW | N/A | N/A | NS | |
| E-IV | User Interface | 1 | Export report to file | SW | N/A | N/A | NS | |

** POC for the Database Access Computer is Doriann Jaffee - (619) 553-6915*

| Category | Major Functions | # | MDARS SUPPORT PROGRAM* Functionality / Requirement | HW or SW | DOS Status | NT CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| III | Display | 1 | Convert S/W to window NT Operating System | SW | N/A | N/A | CP | 8/26/96 |
| III | User Interface | 1 | Provide Rhetorex 9432 device support for remote user interface | SW | N/A | N/A | CP | 8/26/96 |
| III | " | 2 | Allow user to reconfigure emergency calling, emergency phone list, and platform tracking via GUI inputs | SW | N/A | N/A | CP | 8/26/96 |
| III | " | 3 | Allow user to reconfigure emergency calling, emergency phone list, and platform tracking via Rhetorex-supported phone inputs | SW | N/A | N/A | CP | 8/26/96 |
| III | Configuration | 1 | Support both NRaD and Kramer video switchers | SW | N/A | N/A | CP | 8/26/96 |
| IV | User Interface | 1 | Provide on-line (windows) help capability | SW | N/A | N/A | CP | 6/1/97 |
| E-I | Command | 1 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |

** POC for the MDARS Support Program is Kelly Grant (619) 553-0850*

| Category | Major Functions | # | PLATFORM SIMULATOR*<br>Functionality / Requirement | HW or SW | DOS Status | DOS CP Date | NT Status | NT CP Date |
|---|---|---|---|---|---|---|---|---|
| I | Initialization | 1 | Display program usage/version and provide user selectable parameters as a command line option | SW | CP | 7/2/93 | CP | 6/1/96 |
| I | Display | 1 | Provide Help display mode which displays the help menu | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 2 | Provide Blackboard display mode which displays appropriate headers, continuously update memory blackboard and robot status window | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 3 | Provide Program Path Decoding mode which displays the robot status window and the current downloaded program in an understandable format similar to the platform | SW | CP | 7/2/93 | N/A | N/A |
| I | Command | 1 | Provide on screen option menu | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 2 | Blackboard manipulation: provide several capabilities of manipulating memory blackboard either from direct keyboard or option menu | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 3 | Provide robot status window display | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | Housekeeping | 1 | Provide simulated communication using communication protocol between the Link Server and the simulated computers on board robot | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 2 | Decoding all the instructions which currently are implemented | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 3 | Simulate platform while in patrol mode | SW | CP | 7/2/93 | CP | 8/26/96 |
| I | " | 4 | Simulate most of the instructions described in the platform control language section | SW | CP | 7/2/93 | N/A | N/A |
| I | User Interface | 1 | Provide user-controlled blackboard memory and decoding program path scroll | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 2 | Provide automatic scrolling in the program path decoding mode while the simulator robot is in patrol mode | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 3 | Provide keyboard input capabilities | SW | CP | 7/2/93 | N/A | N/A |
| I | " | 4 | Provide advance highlighted byte using arrow keys | SW | CP | 7/2/93 | N/A | N/A |
| II | Command | 1 | Provide emergency halt and resume | SW | CP | 12/20/93 | CP | 8/26/96 |
| II | " | 2 | Provide teleoperation/manual mode simulation | SW | CP | 1/20/94 | N/A | N/A |
| II | " | 3 | Provide AVAM functionality simulation in all modes | SW | CP | 9/25/94 | N/A | N/A |
| II | Housekeeping | 1 | Simulate 8 parallel output bit for the DB-02 Beacon Control for 30 docks | SW | CP | 7/2/93 | N/A | N/A |
| II | " | 2 | Support Planner enhancements with new Platform | SW | CP | 10/30/93 | N/A | N/A |
| II | " | 3 | Support for Product Assessment System (baseline) | SW | CP | 2/25/94 | N/A | N/A |
| II | " | 4 | Support for Product Assessment System (as required for actual Tag Reader Computer) | SW | CP | 7/15/94 | N/A | N/A |
| II | " | 5 | Simulate hardware errors (e.g., battery voltage drops) | SW | CP | 11/2/93 | N/A | N/A |
| II | " | 6 | Simulate platform operating in survey mode | SW | CP | 3/29/94 | N/A | N/A |
| II | " | 7 | Simulate Tag Reader Computer | SW | CP | 3/29/94 | N/A | N/A |
| II | " | 8 | Simulate program path execution with varying speed | SW | CP | 10/94 | CP | 8/26/99 |
| II | " | 9 | Convert to Windows NT | SW | NS | N/A | CP | 8/26/96 |
| II | " | 10 | Support Ethernet communications protocol | SW | N/A | N/A | CP | 8/26/99 |
| II | " | 11 | Simulate AUTOMATIC, HALT, RESUME, MANUAL, TELE-REFLEXIVE mode | SW | NS | N/A | CP | 8/26/96 |
| II | " | 12 | Simulate MANUAL, TELE-REFLEXIVE, PATROL, EMERGENCY_HALT, SELECTIVE_HALT modes | SW | N/A | N/A | CP | 8/26/96 |
| II | " | 13 | Support IDD protocol | SW | N/A | N/A | CP | 8/26/96 |
| II | User Interface | 1 | Bullet-proof keyboard input | SW | CP | 3/1/94 | CP | 8/26/99 |
| II | " | 2 | Provide additional variable modification capabilities | SW | CP | 2/28/94 | CP | 8/26/96 |
| II | " | 3 | Provide a function key to flip through display blackboards | SW | CP | 10/10/94 | N/A | N/A |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| II | " | 4 | Provide a command line option (-l) to log I/O data between host and platform | SW | CP | 1/20/95 | N/A | N/A |
| III | Command | 1 | Provide log file play back with user-controlled playback speed | SW | CP | 7/2/93 | N/A | N/A |
| III | Housekeeping | 1 | Simulate RECALL mode. | SW | NS | N/A | N/A | N/A |
| III | " | 2 | Simulate SURVEY mode. | SW | NS | N/A | CP | 10/30/96 |
| III | " | 3 | Simulate Platform INVENTORY mode. | SW | NS | N/A | CP | 12/30/96 |
| III | " | 4 | Simulate diagnostic failures | SW | NS | N/A | CP | 5/26/99 |
| III | " | 5 | Degrade gracefully/recover when Link Server CSCI or LAN cable is disconnected | SW | NS | N/A | CP | 1/8/97 |
| III | " | 6 | Simulate SPI Pan/Tilt | SW | N/A | N/A | CP | 5/30/97 |
| III | User Interface | 1 | Provide menu selection for setting Platform mode and status. | SW | N/A | N/A | CP | 8/15/97 |
| III | " | 2 | Provide menu selection of various diagnostic failures | SW | N/A | N/A | CP | 5/26/99 |
| III | " | 3 | Implement pull-down menus for user interface | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Housekeeping | 1 | Recognize platform type | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 2 | Initialize platform | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 3 | Process platform status | SW | N/A | N/A | CP | 5/26/99 |
| E-I | Command | 1 | Process emergency halt | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 2 | Support random patrols | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 3 | Support directed sends | SW | N/A | N/A | CP | 5/26/99 |
| E-I | " | 4 | Support DGS message relay | SW | N/A | N/A | CP | 5/26/99 |
| E-II | Initialization | 1 | Display error/usage window | SW | N/A | N/A | CP | 11/8/99 |
| E-II | Command | 1 | Support teleoperation | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 2 | Support camera select | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 3 | Support camera move | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 4 | Support camera control | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 5 | Support debug capability | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 6 | Support diagnostic capability | SW | N/A | N/A | CP | 11/8/99 |
| E-II | " | 7 | Support tag get data | SW | N/A | N/A | CP | 11/8/99 |
| E-III | Command | 1 | Support degraded operations | SW | N/A | N/A | CP | 2/10/00 |
| E-IV | Command | 1 | Support lock get data | SW | N/A | N/A | CP | 4/24/00 |
| E-IV | User Interface | 1 | Support two-way audio link | SW | N/A | N/A | NS | TBD |
| P3I | Command | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | Housekeeping | 1 | ICIDS integration | SW | NS | N/A | NS | |
| P3I | User Interface | 1 | ICIDS integration | SW | NS | N/A | NS | |

** POC for the Platform Simulator is Daniel Carroll - (619) 553-7636*