TECHNICAL DOCUMENT 3026
August 1998

Multiple Resource Host Architecture for the Mobile Detection Assessment and Response System

H. R. Everett
R. T. Laird
G. Gilbreath
T. A. Heath-Pastore
R. S. Inderieden
K. Grant
D. M. Jaffee

Approved for public release;
distribution is unlimited.

SPAWAR
Systems Center
San Diego

Space and Naval Warfare Systems Center
San Diego, CA 92152–5001

SPACE AND NAVAL WARFARE SYSTEMS CENTER
San Diego, California 92152–5001

Commanding Officerl **Executive Director**

ADMINISTRATIVE INFORMATION

The work detailed in this report was performed for the U. S. Army R&D Center Physical Security Equipment Division by the Advanced Systems Division, Code D37, Space and Naval Warfare (SPAWAR) Systems Center, San Diego (SSC San Diego).

Released by

Associate for Robotics
Advanced Systems
Division

Under authority of

Advanced Systems
Division

This technology is related to the subject matter of one or more U. S. Patents assigned to the U. S. Government, including Patent No. 5,659,779. Licensing inquiries may be directed to:

Legal Counsel for Patents
Space and Naval Warfare Systems Center, Code D0012
San Diego, CA 92152–5765
(619) 553–3001

CONTENTS

Figures

Tables

1. BACKGROUND

The Mobile Detection Assessment and Response System (MDARS) Program is a non-standard program managed by the U.S. Army Physical Security Equipment Management Office (PSEMO), Fort Belvoir, VA. Mr. Jerry L. Edwards, PSEMO, is the MDARS Lead Project Officer responsible for overall program management.

1.1 DEVELOPMENT TEAM ORGANIZATION

The MDARS Team Organization (figure 1) was structured to foster program efficiency to meet the aggressive development schedule.

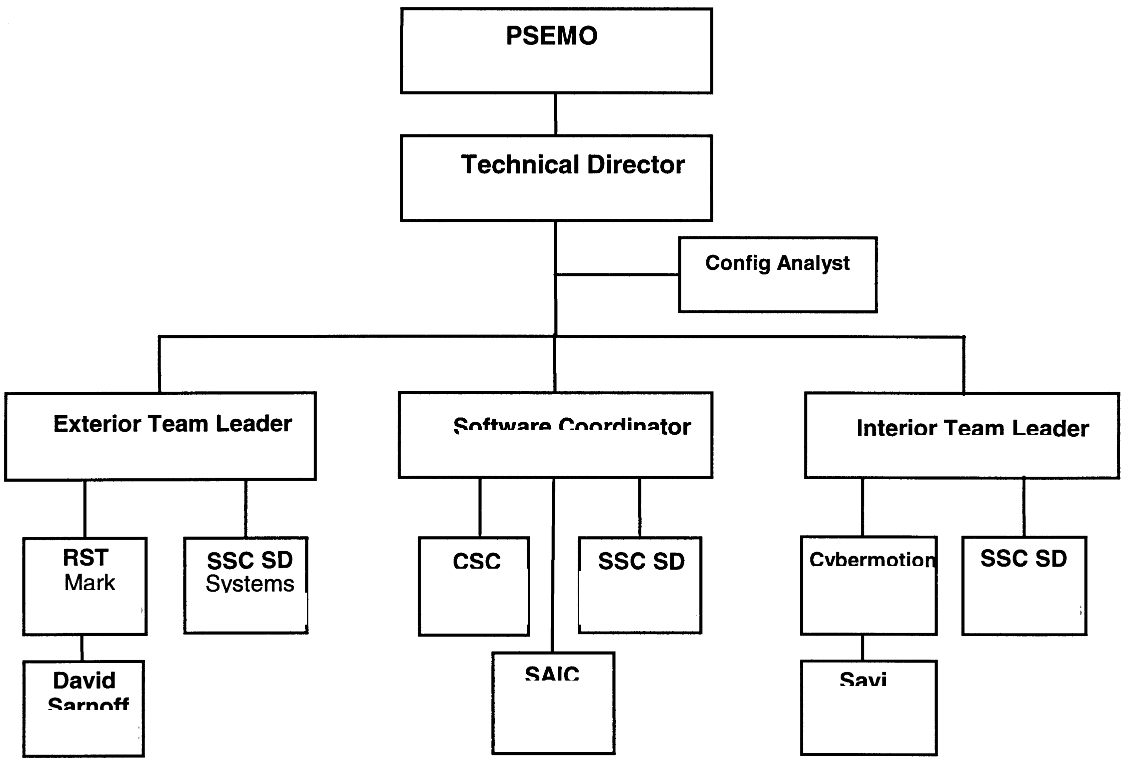

Figure 1. MDARS Development Team organization.

1.2 MULTIPLE PLATFORM CONTROL PHILOSOPHY

The initial effort during Phase I of the program was aimed at demonstrating the feasibility of a single robotic security platform operating under the high-level control of a remote host, with the direct supervision of a human operator. The coordinated control of multiple platforms poses some significant design concerns that must be addressed through appropriate trade-off analysis.

One of the first questions that 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 implementation schemes are considered: whether to include distribution of computational resources at both the host and remote ends, which communication strategies to use between the two, and which man-machine interfaces for real-time multiple-platform operation are required.

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 man-in-the-loop control. The tradeoff involved is simple: real-time response is going to suffer as the quantity and complexity of operator interactions is 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 human guard deals with a higher priority situation involving another platform.

To eliminate this potential problem, one obvious alternative might 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 fails to provide the necessary navigational and assessment tools required to support this degree of autonomy.

- Political and legal considerations dictate a man-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 host architecture geared towards a single guard (figure 2) controlling up to eight 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.

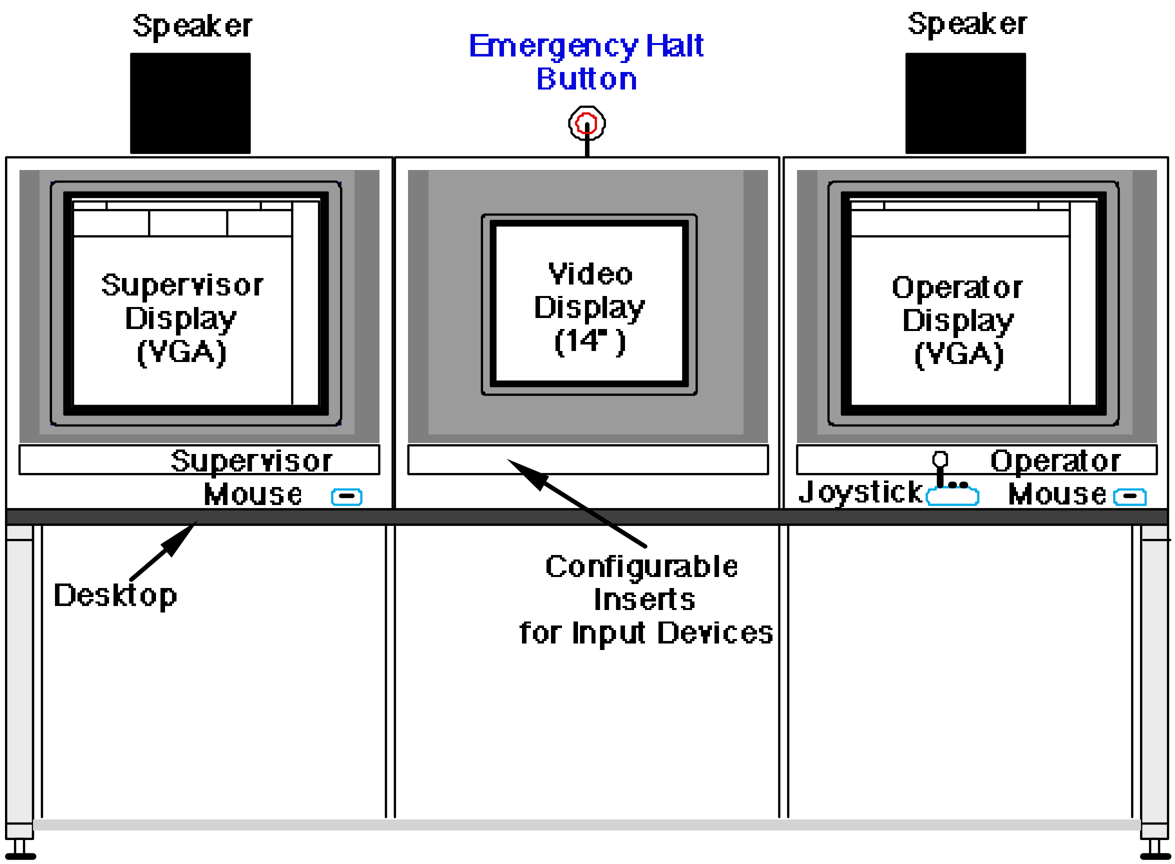

Figure 2. Possible layout of Guard Control Console.

1.3 DEVELOPMENTAL APPROACH

MDARS development approach involved phased rapid-prototyping 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, which are presented in appendix B of this document.

These required functionalities have been broken out into four distinct categories to facilitate phased development and in-house testing and to provide a standardized mechanism for tracking and reporting. 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 completion dates for the new Windows NT-based MRHA in addition to the original completion dates for the earlier DOS-based implementation. The following summaries are included to provide a high-level appreciation for the overall intent of each of the four categories.

1.3.1 Category I

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. All listed functionalities (appendix B) were successfully completed on schedule.

1.3.2 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 has been completed in the area of robust navigation in a semi-structured environment. Cybermotion has incorporated many of these features in later versions of their onboard software.

The capability of reading radio frequency (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 were 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 since 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 informed of this situation in February 1995, and were 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 4.1.

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

1.3.3 Category III: Inventory Tagging and Assessment

The principle focus of Category III was to develop and demonstrate a successful and cost-effective inventory assessment strategy that can demonstrate value-added during Early User Appraisal (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 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 attempted to work with PSEMO 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.

1.3.4 Category IV: Early User Assessment

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

1.4 SYSTEM REDESIGN

Toward the end of 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.

1.4.1 Operating System Conversion

The memory problem occurred because 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, we performed an informal survey 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, as the use of Ada had been mandated. 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 worldwide, hosts a number of validated Ada compilers, and executes on the current target hardware (industrial rack-mounted PCs).

In summary, necessary growth of the MRHA forced conversion to a new operating system—this was not a decision we took casually, nor did we view this as an opportunity to explore the latest in Microsoft technology. Windows NT has been installed and operational for several months. The operating system itself poses very minimal additional technical risk. However, conversion of existing software to run under Windows NT does pose a formidable technical and programmatic risk. Unfortunately there was no reasonable alternative.

1.4.2 Software Conversion

Conversion from MS-DOS to Windows NT required that we modify the existing MRHA software to run under the new operating system, as the old code simply would not execute under Windows NT. To effectively manage the conversion effort, we planned for a phased approach whereby we would first implement Category II functionality under Windows NT and, then, after the system had been successfully tested against a baselined test plan and known trouble report set, we would proceed with Category III functionality development. 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). Our problems would have been compounded if we were to simultaneously change operating systems and attempt to add new code to address Category III requirements. This is known as "crawl-before-you-walk" software engineering.

Since we faced converting all existing MRHA software to operate under Windows NT (a task that would require redesign of several major components), it was decided that all software would be converted to the Ada programming language in keeping with the verified direction provided by CECOM. This impacted the Operator, Planner, Database Access Computer, and Robot Simulator CSCIs, but resulted in a more robust system with significantly reduced maintenance costs. 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. Software maintenance in-house was difficult at best, since no single individual was well versed in any two languages, let alone all four. 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% of the MRHA software will be shared among the eight CSCIs. This factor alone contributed to long-term software maintenance savings on the order of several hundred thousand dollars, given that we have reduced the number of required maintenance programmers whose annual cost is minimally $75K.

The conversion effort did not add capability to the system beyond what was previously attained under MS-DOS at the end of Category II; it simply brought us back to where we were except that we now have a new (expandable) operating system and a common software language for improved supportability. Once Category II functionality under Windows NT was completed and successfully tested; we had an established baseline upon which to build Category III functionality.

reduce our software maintenance costs in the long-term. Once the software conversion was completed, we faced the very large task of systems integration.

1.5 SYSTEMS INTEGRATION

Following each phase of Category development was a period of system-level integration for both hardware and software components (CSCIs/HWCIs). Specifically, following conversion of the MRHA to the Windows NT operating system, we began systems integration with the existing interior K2A platforms. We used the "Category II" interior platform to perform initial systems integration in order to minimize the number of variables during what is primarily integration of re-engineered software systems—we needed to keep the platform configuration constant as the software changes. Once the Category II Windows NT software was successfully integrated with the old platform, we retested the entire system against the existing Category II test plan in order to re-establish a solid baseline. From that point we proceeded with confidence to Category III software development that targeted the new (BAA) interior and exterior platforms (see figure 3).

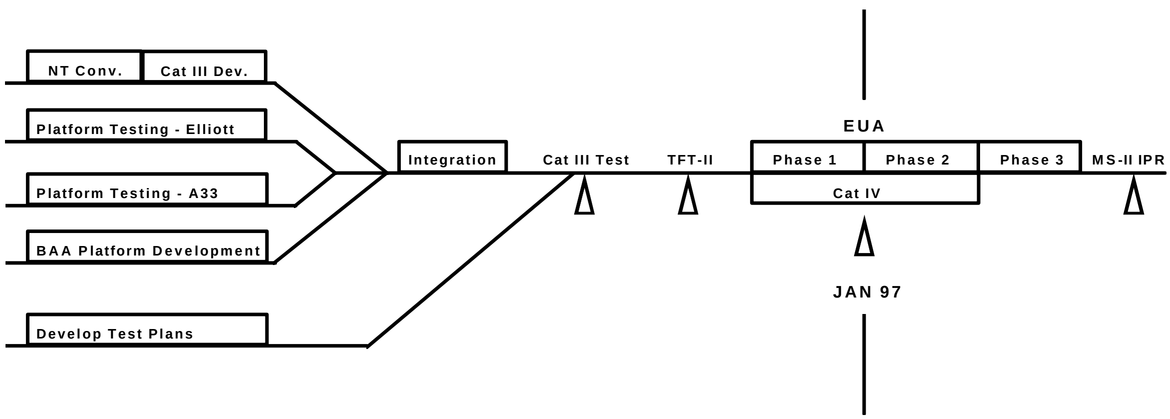

Figure 3. MDARS interior timeline showing the integration efforts following Category II rework and Windows NT conversion, followed by Category III development and testing and then TFT-II.

Integration activities took place throughout the entire Category development cycle. However, final systems integration took place over a number of weeks, during which time all of the major subsystems were brought together and their interfaces tested and debugged. Only after integration was complete did we proceed with Category testing, which typically took only a few days. Category testing was performed against a formal set of test plans, with the results recorded and published.

Systems integration did not in itself contribute to increased operational capability; it simply brought together the pieces that embody functionality into something that could then be tested and subsequently operated reliably while fulfilling the user's expectations. It was a very necessary step in a software-intensive development program that was often overlooked or seriously underestimated. The bottom line was this: we could provide a piece-wise capability to meet all the stated requirements in the MDARS Operational Requirements Document (ORD) yet still not have a system that is practical or acceptable to the user, unless adequate attention and resources are paid to effective systems integration.

This philosophy did not mean that the system had to be perfect in all respects, nor that it should have been overdeveloped to the point of excess. It simply meant that care must have been 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 could have been fatal to an otherwise healthy program. The optimal solution was somewhere in the middle, and it took 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 was 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 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 does not wind up with something that meets the letter of the ORD but not the intent. The bottom line here is it must be soldier-proof; if it takes an engineer to run it, it is of little value to anyone, even if it did 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 does not 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 effect.

2. SYSTEM OVERVIEW

A high-level block diagram of the Multiple Resource Host Architecture (MRHA) is presented in figure 4. 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 resource 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.

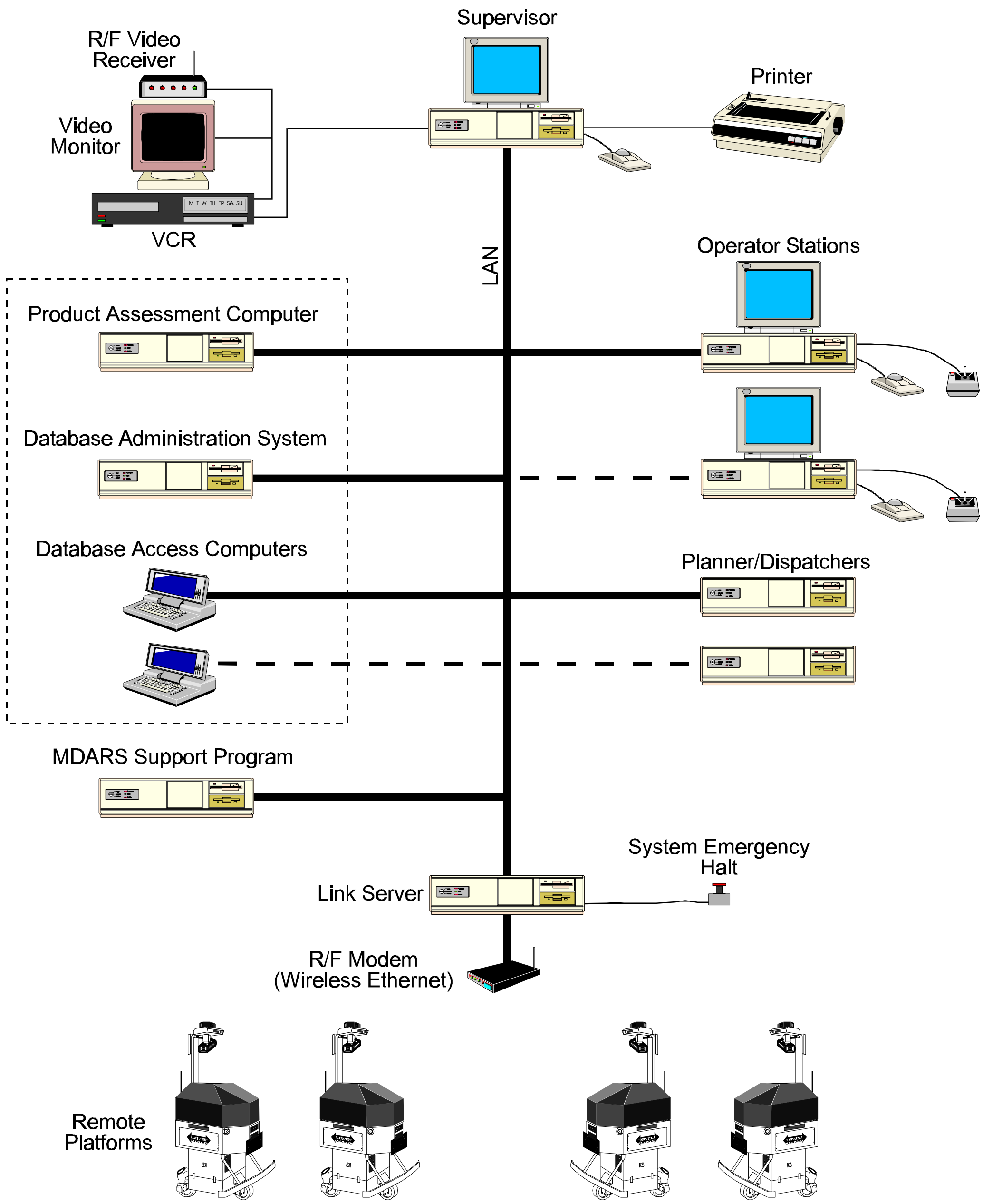

Figure 4. Expandable host architecture block diagram.

Figure 5 shows the Supervisor display monitor located on the left side of the prototype guard console. 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 require installation at an existing workstation, which implies usage of desktop-style VGAs. Rack-mounted computer equipment such as shown in figure 6 would then be installed in an adjacent room.

Figure 5. Prototype of Guard Station Console used for development purposes at SSC San Diego.

As shown in the block diagram, the Supervisor can access a number of Planner/Dispatcher computers linked over a common high-speed local area network (LAN). 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 figures 5 and 6. A globally shared world model is maintained by these modules to provide a real-time collision avoidance capability complementing the Cybermotion *virtual path* navigation scheme employed on their K2A robotic platform. The Planner/Dispatchers, in essence, perform the current *virtual path* planning functions of the Cybermotion Dispatcher (Holland et al., 1990) on an as-needed basis.

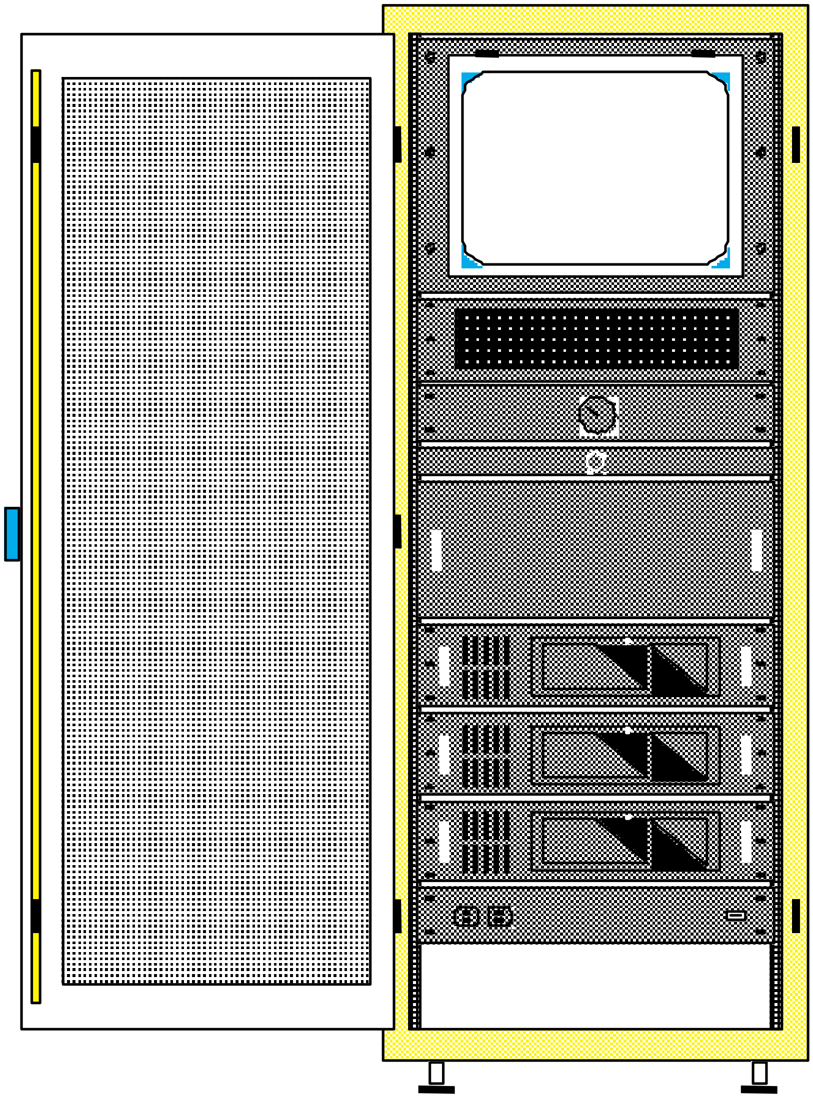

Figure 6. Optional 19-inch rack for MRHA hardware.

Similarly, the Supervisor has access over the LAN to one or more Operator Stations, as shown in figure 4. 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 several remote platforms.

All the distributed resources within the host architecture communicate with the various remote platforms via a RF 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/Dispatcher, Operator Station) on the host LAN can, thus, simultaneously communicate, as needed, in real-time with their assigned remote platforms.

The number of Planner/Dispatcher and Operator Station modules resident on the host LAN can be varied as implied in figure 3 in proportion to the number of remote platforms employed. Preliminary considerations, as discussed in the initial MDARS Work Package Review Meeting at the Armament Research Development Engineering Center (ARDEC) on 12 September 1991, called for the integration of a number of embedded Planner/Dispatcher 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, for reasons discussed below:

- A Dispatcher *virtual path* planning operation takes only a fraction of a second to generate a sequence of route segments, which is then downloaded to the remote K2A computer via the Scheduler onboard the robot.

- The SSC San Diego real-time security assessment algorithm, which previously ran on the Planner, has been ported down to the platform and incorporated into the Cybermotion *SPI* computer.

- The only other function of the Planner would be to depict the platform's position and orientation as the path was traversed. This position display function, however, is to be performed under the proposed multiplatform scheme by a separate display computer (i.e., the Supervisor).

Accordingly, it was decided that the host architecture depicted in the block diagram of figure 3 represented the optimal solution as a minimal hardware configuration without significant tradeoffs in performance and response time. The initial prototype systems developed by SSC San Diego will be configured with a Supervisor, two Planner/Dispatchers, one Operator Station, and one Link Server for coordinated control of up to eight patrol units (Laird et al., 1993).

The major components of the MRHA (the Supervisor, the Planner/Dispatcher, the Operator Station, the Link Server, the Product Assessment Database, and the 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.

Automatic assignment of resources (Planner/Dispatcher, 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 a Planner/Dispatcher, or both a Planner/Dispatcher 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 five 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/Dispatcher 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 section 3.1.1.1), followed by initialization file entries (see section 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 pass down to a Planner/Dispatcher 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 (figure 7):

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 displays 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 7. 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.

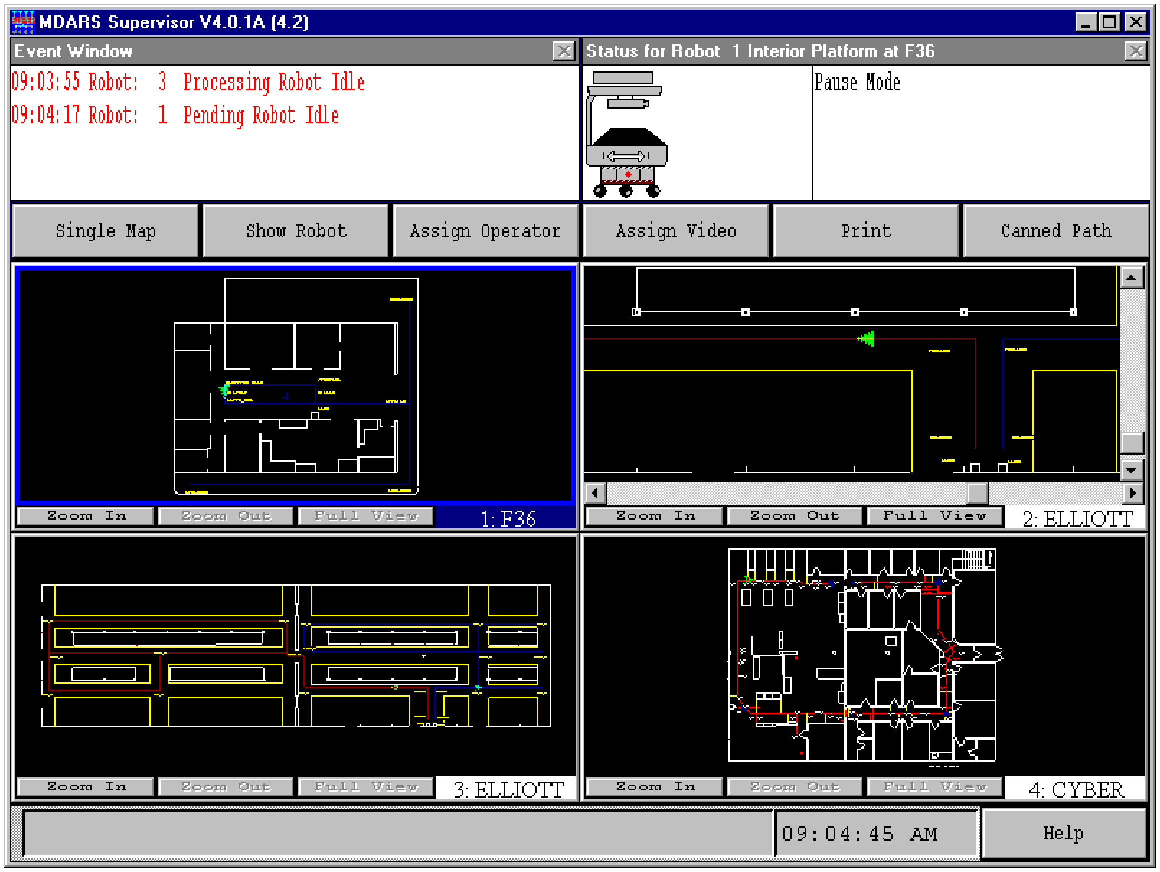

Figure 7. Actual screen dump of prototype Supervisor display.

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 that requires a subsequent platform selection, a *Platform Select Listing* is overlaid on the *Status Window* as shown in figure 7. 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. Clicking on the Cancel button in the Robot Select Listing cancels the unit selection process.

The Robot Select Listing will provide a Cancel button to terminate the selection process for any command. A button titled Turn All 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 the 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 7) derives its information for display purposes from the blackboard data structure, as partially illustrated below in table 1.

The title bar of the Status Window displays 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.

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	3.1	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 right side of the display displays 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 messages with black text on a yellow background, and normal messages 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 automatically displays 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. Clicking in the map area of the desired map also provides map selection.

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 cannot 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 Window pane. The following buttons are provided in all Map Window panes 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

Note: This limited color-coding scheme makes no distinction between platform modes, such as IDS versus Patrol.

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 a Planner/Dispatcher, or both a Planner/Dispatcher and an Operator Station. Such events can be of two types: 1) assignable events, for which the Supervisor can automatically allocate 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 in table 2.

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 7). 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 (figure 8).

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.)

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/Dispatcher
Yes	Platform Blocked	Platform
Yes	Low Fuel	Platform
Yes	Low Battery	Platform
No	Directed IDS	Operator Station
Yes	Platform Idle Condition	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/Dispatcher for automatic handling, the status will be Processing. If assigned to an Operator Station, the status is listed as Assigned For. This status coding helps the user understand that 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: THREAT LEVEL 82
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 8. Sample Event Window, with fifth (lowest priority) line blank. A perceived threat score of '82' is reported for platform number 1. 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 will be printed out on the printer at the end of each guard shift, or when 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 way 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 (see 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/Dispatcher 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 guard uses the Assign Video menu button 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 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 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 Scheduler 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 does not 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 Operators, multiple link configurations will have a conflict resolution mechanism that 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 toggles 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 the 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* occur 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 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 appears in red and a perceived threat vector displays on the screen. An *Alarm* status appears in the Supervisor *Event Window*, and an audible alert sounds (section 3.1.3). The platform automatically assigns a Planner/Dispatcher and an Operator Station, as discussed in section 3.2.

3.1.2.7.2.3 Platform Blocked. If a platform is blocked, a Planner/Dispatcher is automatically assigned, and *Blocked* status appears in the Supervisor *Event Window* (section 3.1.3).

3.1.2.7.2.4 Failed Diagnostic. A diagnostic failure appears in the *Event Window*, and the guard can use the mouse to select the icon or ID listing that has indicated a problem. When the guard clicks on the *Show Robot* button and then on the icon, the *Status Window* depicts the condition of the various critical elements for that particular platform. (The *event* listed in the *Event Window* describes the actual diagnostic failure, i.e., *E_STOP_CIRCUIT_OPEN*, not the cryptic Failed Diagnostic). A Planner/Dispatcher and Operator Station are automatically assigned.

3.1.2.7.2.5 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 K2A computer 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 are cleared by the K2A computer (i.e., reset to 0,0,0), such as when the robot is first powered up.

A *Platform Lost* condition is most likely caused by an accumulation of dead-reckoning errors. A common situation occurs when the platform is closer to a wall than its dead reckoning says it should be. This can typically happen in one of two ways:

- The wall is ahead of the platform and probably used as the navigational reference for an *Approach* instruction.

- The wall is to the side of the platform and is probably 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 Unreferenced condition typically results (in addition to a cold start) if the onboard K2A computer somehow loses its X, Y, and Heading parameters. This necessitates an embedded diagnostic check running on the Scheduler of each platform that constantly checks delta-X, delta-Y, and delta-theta for radical discontinuities, as well as a powerup reboot flag if the entire remote system glitches.

A Platform Lost or Platform Unreferenced condition may be detected in one of two ways:

- Automatically by the Supervisor, when the *Scheduler Status Robot Lost* bit indicates that the position is invalid.

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

In the first case above, the Scheduler sets a flag on the robot indicating that the robot is lost. The platform will be automatically assigned to a Planner and an Operator Station with Robot Lost status. The guard must then teleoperate the robot to the vicinity of a referencing location and reference the robot using the Reference command.

The second case typically occurs after the robot has been teleoperated extensively or sent on a long, unrestricted path, either one of which will induce significant dead reckoning errors. When this happens, the system assigns a Planner and Operator as before, but with a Robot Trapped condition. It is then up to the guard to assess the situation and determine whether the robot is trapped or lost.

An unreferenced platform is halted by its own Scheduler if moving in Automatic Mode, and then shows up on the Supervisor Display as Off-line. The guard must then click on the Show Robot button to find out why the platform is Off-line (i.e., it is unreferenced.) The Supervisor, however, is already aware of the situation and, when able, will assign a Planner/Dispatcher as well as an Operator to the task of re-referencing the platform so it can continue patrolling.

3.1.2.7.2.6 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 that 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*, the Supervisor, based on additional information contained in the blackboard, automatically assigns resources. This information represents the detailed status of every platform in the system, as well as the availability of the resources themselves. The example illustrated in table 3 shows where platform5, with the highest priority need, has been assigned Planner/Dispatcher No. 2, and Operator Station No. 1. Platform3 has been assigned Planner/Dispatcher 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. Proposed layout of that portion of the blackboard data structure supporting the assignment function. Assignment priority is taken from table 1.

Platform ID	Assignment Priority	Assigned Planner	Assigned Operator	K2A Program	Map Name
Platform1	4				B100
Platform2					B203
Platform3	3	1		Robot2.k2a	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/Dispatcher 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/Dispatcher 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/Dispatcher 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 needs 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 fails, a Guard should be notified, and a Planner/Dispatcher and Operator Station assigned.
- Planner/Dispatchers 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/Dispatcher is available, the lower priority *events* that only require a Planner/Dispatcher will be assigned to the available resource ahead of the queued *events* that require both a Planner/Dispatcher and an Operator Station.
- There should be at least one more Planner/Dispatcher 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/Dispatcher, 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/Dispatcher
- The assigned Operator Station (if applicable)

- The problem (event code) or reason for assignment
- The last K2A program in execution onboard the platform, if appropriate

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 the 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 the Planner completion data as a component of the completion message. 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. If 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.
- Late Planner Completion. Some lengthy paths may be started by an Operator who then relinquishes control of the resources, even though the Planner and the robot are still working together. The Supervisor must track this and accommodate a Planner completion arriving after the Operator completion.
- 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 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.

3.1.2.8 Housekeeping Functions

3.1.2.8.1 Robot Program Storage. Each time a Planner/Dispatcher downloads a program to a platform, the Planner/Dispatcher 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 Planner/Dispatchers use the previous program 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 maintains 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) from the individual *virtual path* programs. Only service personnel will perform creation and maintenance of this database.

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. Menu items and platform icons are selected from 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 and 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.

Table 4 provides an evaluation summary of each device.

Table 4. Interface devices.

Device	Mouse
Advantages	Low Cost
	Accurate
Disadvantages	Requires some training and practice
	Susceptible to contamination (food, dirt)
	Susceptible to abuse
Unit Cost	$40 to $300
Device	Trackball
Advantages	Low Cost
	Accurate
	Stationary
	More durable than mouse
Disadvantages	Requires some training and practice
	Susceptible to contamination (food, dirt)
Unit Cost	$40 - $400
Device	Touch Screen
Advantages	Intuitive interface (minimal training)
	Accurate on targets larger than 0.33" x 0.5"
	Smaller targets with software assistance
	Rapid selection
	Requires no additional desk space

Table 4. Interface devices. (Continued)

Device	Touch Screen
Disadvantages	High unit and development costs Some screen technologies susceptible to scratches Accidental activation of controls Finger partially obstructs display Potential user arm fatigue
Unit Cost	$600-$5000

At this point in the evaluation, several conclusions can be made. 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 input/output (I/O) port communicates with an RS-232-controllable VCR.

3.1.2.9.2.3 Sound Card. A Sound Blaster sound board 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 table 5). Ordinary events (such as Robot Idle) will be handled routinely without disturbing the user. Each *event* is configurable (via the initialization file (see section 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 can control and display 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, and obstacles will automatically be avoided using Planner/Dispatcher assignments. 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:

Software Development Plan (CSC, 1992a)

Initial Software Design Document (CSC, 1992b)

Initial Interface Design Document (CSC, 1992c)

Detailed Design Document (CSC, 1998)

Software Test Plan (Laird et al., 1996)

Software Test Description (Grant, 1996)

Software User's Manual (Grant, 1998)

User Manual and Training Guide (Grant, 1998)

4. PLANNER/DISPATCHER

The current MDARS navigation scheme is basically an integration of the Cybermotion *Dispatcher* and the Space and Naval Warfare (SPAWAR) Systems Center (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 FY 92 under a Cooperative Research And Development Agreement (CRADA) with Cybermotion, thus giving rise to the term "Planner/Dispatcher." For simplicity, the term "Planner" will be used throughout this section in reference to the "Planner/Dispatcher."

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.

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 that determine its appropriate application, as 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 that reflexively reacts to the sensed position of some external guiding reference. Industrial vehicles have been guided by physical paths including slots, buried wires, optical stripes, and other methods for almost 30 years. 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 that sense and follow the AF or RF field from a closed-loop wire embedded in the floor; (2) those that sense and follow a magnetic tape in or on the floor; and (3) those that 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 that track a special reflective tape illuminated by a near-infrared source, and a chemical stripe that glows when irradiated by ultraviolet energy.

Advantages of guidepath control are seen primarily in the improved efficiency and reduction of manpower that 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 that 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 if changes to product flow lines in assembly plants, or in the case of a security robot that must investigate a potential break-in at a designated remote location.

4.1.2 Unrestricted Path Planning

The term unrestricted path planning implies the ability of a free-roaming platform to travel anywhere desired, subject to nominal considerations of terrain traversability. Many potential applications await an indoor mobile robot that could move in a purposeful fashion without following a set guidepath, with the intelligence to avoid objects and, if necessary, choose alternative routes of its own planning. Most of the path planning work to date was done on the premise that the ultimate navigation system would be capable of mapping out its environment with sensors, and then plan routes accordingly. While such systems have much appeal, they encounter several significant difficulties in practice.

The most significant problem associated with building a world model is the poor quality of most sensor data. There are many choices available to the designer of such a navigation system, but in every case good data are expensive. In practice, reflective sensors (ultrasonic rangefinders and near-infrared proximity detectors) have predominated. All reflective sensors are subject to the problems of noise, specular and secondary reflections, and signal absorption to one extent or another.

Of all the data provided from a sensor system at a given location, only a small percentage will be true, pertinent, and accurate. Furthermore, the position of objects viewed from different positions will be distorted by any errors in the vehicle's dead reckoning accuracy as it moves between vantage points. Template matching of sensor data can, thus, be very difficult. Finally, some areas that appear clear to sensors may, in fact, be dangerous or impassable for other reasons. This fact places an additional "data-editing" responsibility on the vehicle's programmer (Holland et al., 1990).

Specialized sensors must be coupled with some type of world modeling capability in order to provide a mobile platform with sufficient awareness of its surroundings to support intelligent movement in unstructured environments (Everett, 1995). The model represents a two-dimensional mapping of the absolute locations of surrounding objects, and is refined in a continuous fashion as the platform moves about its workspace. The accuracy of this model is directly dependent upon the validity of the platform's own perceived location and orientation. Accumulated dead-reckoning errors soon render the information entered into the model invalid in that the associated geographical reference point for data acquired relative to the platform's position is incorrect.

As the accuracy of the model degrades, platform’s ability to successfully navigate and avoid collisions diminishes rapidly, until it fails altogether. A robust navigational scheme that preserves the validity of the world model for free-roaming platforms has remained an elusive research goal and, for this reason, many proposed applications of truly autonomous mobile robots are yet to be implemented.

Providing an autonomous capability to support nonrestricted motion, therefore, involves the implementation of an appropriate map representation, the acquisition of information regarding ranges and bearings to nearby objects, and the subsequent interpretation of that data in building and maintaining the world model.

4.1.2.1 Selecting a Map Representation. Several different map representation schemes have been devised, including polyhedral objects (Lozano-Perez, 1979), generalized cones (Brooks, 1983), certainty grids (Moravec, 1987), and quadtrees (Fryxell, 1988). The simplest scheme is a two-dimensional array of cells; each cell corresponds to a square of fixed size in the region being mapped. The map can be accessed and updated quickly, which is extremely important for real-time operation. Free space is indicated with a cell value of zero; a nonzero cell value indicates an object. The most compact form of a cell map consists of 1 bit per cell, and thus indicates only the presence or absence of an object. By using multiple bits per cell, additional descriptive information can be represented in the map, such as identification of structural walls and doorways. In addition, the probability of a given square being occupied can be easily encoded, which turns the map into a form of certainty grid (Moravec, 1987). This statistical approach is especially useful when the precise location of objects is unknown.

4.1.2.2 SSC San Diego Unrestricted Path Planning Algorithm. A wide variety of path planning techniques have been developed over the years, each having various advantages and disadvantages. The actual path planner employed in a given application is often dictated by which world modeling scheme has been chosen. For a certainty grid representation, the most straightforward planner is derived from the Lee maze router (Lee, 1961), with the cell coding enhancements suggested by Rubin (1974). The basic search algorithm begins by "expanding" the initial cell corresponding to the robot's current position in the floor map (i.e., each unoccupied neighbor cell is added to the "expansion list"). Then each cell on the expansion list is expanded, the process continuing until the destination cell is placed on the expansion list, or the list becomes empty, in which case, no path exists. This algorithm will find the minimum distance path from the source to the destination.

The minimal distance path, however, is not necessarily the "best" path. Sometimes it is more desirable to minimize the number of turns, or to maximize the distance from obstacles, for example. The search strategy can be altered accordingly by assigning a cost to each cell prior to adding it to the expansion list; only the minimum-cost cells are then expanded. This is known in the literature as an A* search (Winston, 1984), and was adopted by SSC San Diego for use in this work (Everett et al., 1990) due to the inherent flexibility of the associated cost function.

4.1.3 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 preprogrammed 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 (figure 9). 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 K2A to execute an arcing turn between the path legs. The radius of this turn is determined by a RAM variable that can be modified using the "RADIUS" instruction.

The K2A platform has a serial communications network through which it can communicate with its sensor subsystems. When a K2A executes a powerup, it polls this "Control Link" to determine what sensor systems are available. A K2A platform with no sensors will execute a *RUN* or *RUNON* instruction solely by use of its odometry. The phrase "odometry" is used here instead of "dead reckoning" to emphasize a subtle but critical distinction in the method of position sensing. Dead reckoning is generally thought of as the estimation of position after a significantly large movement of known distance and direction.

Odometry is defined here 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).

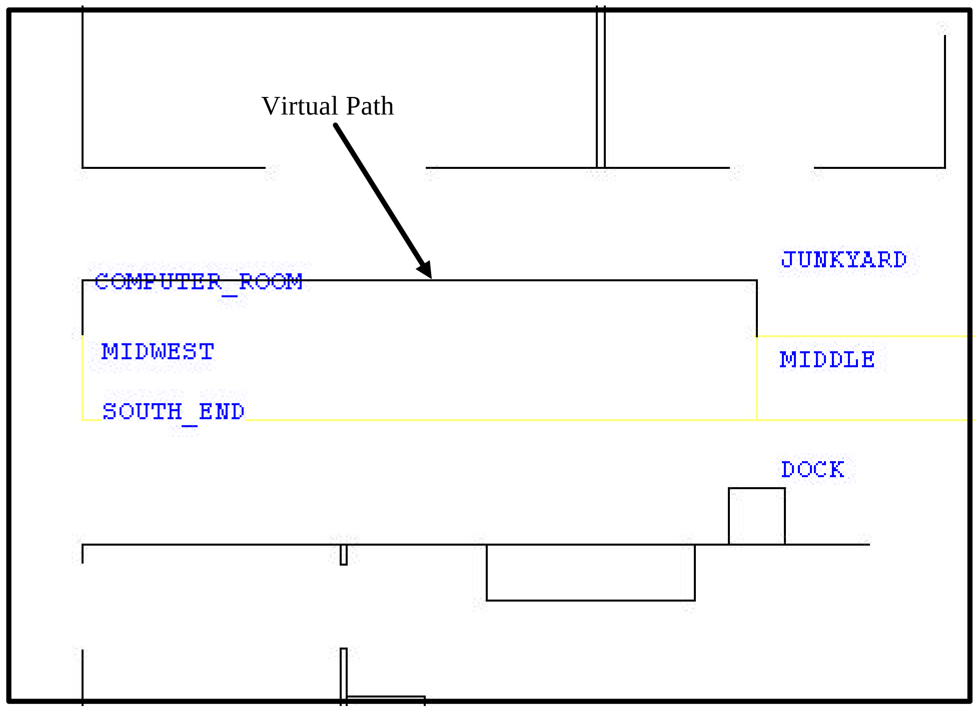

Figure 9. Screen dump of virtual path map (Bldg. F-36) and associated virtual points.

The *Navmaster* is a basic autonomous vehicle consisting of a K2A platform and a sensor turret with a RF data link, a sonar system, and a docking beacon. A K2A platform that finds a sonar system onboard at powerup 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 standoff distances), *WARN* (set beep warning distance), and *C_DEFL* (control veering action) (Holland et al., 1990).

Another sensor available on the *Navmaster* is the Docking Beacon System, which uses a structured light pattern emitted from fixed docking beacons to home into a precise position. Once the vehicle has finished its docking maneuver, it can execute a position and heading correction. Instructions called *MEAN_AZ* and *SETXY* facilitate this correction. Docking is used primarily for auto charging and for controlling fixed equipment such as elevators (Holland et al., 1990).

It is also 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 corrections are 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 is 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 a graphical representation of the world model 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 a Planner for random patrols. The Dispatcher will generate a random virtual path patrol route, and download it via the Link Server to the assigned platform. This onboard K2A 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 Dispatcher.

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, that will generate and download a new random patrol route.

4.2.4 Obstacle Avoidance Functions

The principal duty of the Planner is to plan a path (virtual or unrestricted) and download it to the assigned platform. Under normal circumstances, virtual paths are executed until circumstances arise that require deviation from the predefined route segment. The most common example would involve an obstacle that blocks the virtual path.

Once the Supervisor is made aware that a platform is blocked by an obstacle, it assigns a Planner to resolve the problem. This situation will be reported as an assignable event to the Supervisor. After the Planner has found a path around the obstacle, it downloads the path to the platform. This action starts the platform on its way. The Planner continues to monitor the execution of the avoidance maneuver until the robot reaches the desired destination.

Several problems, however, may occur:

- If there are more robots than Planners, all the Planners could be occupied. In this case, the Supervisor will wait until a Planner becomes available.
- The Planner may fail to find an avoidance path, in which case, it reports a failure to the Supervisor. The Supervisor at this point assigns an Operator Station for the guard to take care of the situation. As above, there may not be an Operator available, and the Supervisor will wait until one becomes free.
- The Planner keeps track of how many planning operations were needed by each platform. At some point the Planner will decide that the platform requires human intervention, and requests the Supervisor to assign an Operator Station.

4.2.5 Intrusion

When the SPI module onboard the platform determines that an intruder is in the area, the Scheduler 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.

4.2.6 Directed Movement

Virtual path routes created by the random path generator will involve Survey stops, as explained in section 4.1 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 Dispatcher will eliminate all the Survey operations altogether, and the path will be executed with no pauses at intermediate virtual points.

4.2.7 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:

Software Development Plan (CSC, 1992e)

Software Test Plan (Laird et al., 1993)

Software Test Description (Gilbreath, 1993)

Design Document for the Planner CSCI (July 1998)

Interface Design Document for MDARS (August 1998)

Design Document for the Common Application Program Components (June 1998)

5. OPERATOR STATION

Detailed hands-on control of a remote platform occurs at the Operator Station. The Operator Station is assigned along with a Planner/Dispatcher 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 accomplish the following tasks:

- 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

If a higher priority need arises, the guard also has the ability to suspend the current operation on the Operator Station. 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 10.

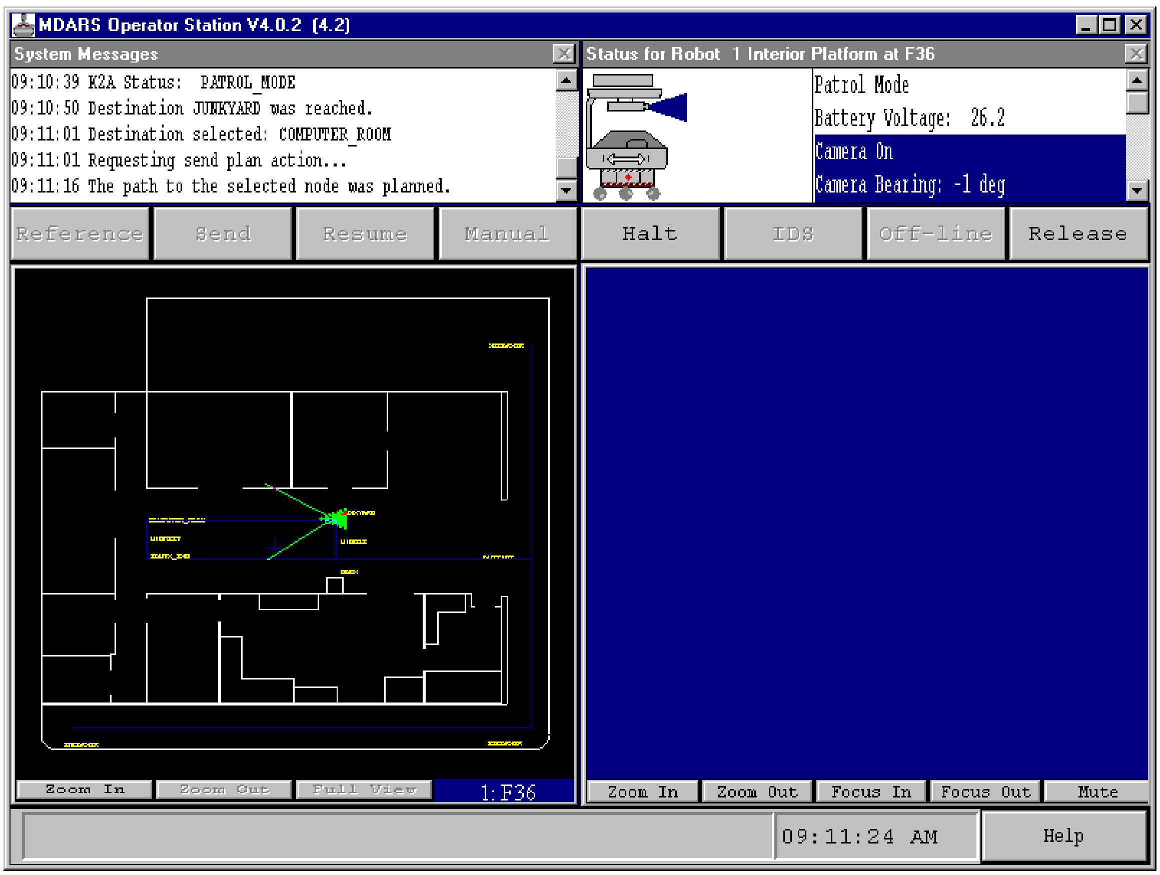

Figure 10. Operator Station screen.

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” appears.

A ***Request Platform*** button allows the guard to request assignment for a specific platform.

5.1.2.2 Standard Display Screen. Upon assignment, the standard Operator Station screen appears. 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 informs 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, trackball). The screen cursor used for on the Operator Station standard screen will have 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 the following: a new situation that may require attention (critical system overlays), task status information (task status overlays), instructions and menu options for completing a command (submenu overlays), and a 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*

Submenu 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, located in the upper left area of the screen, has an identifying caption, *System Messages*, at the top. The window displays all system messages in the format of a scrolling log (figure 10). All messages are time-stamped and stored in chronological order with new messages 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 an overlay window over the message log (figure 11) 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 time period, 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 11. 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 10). 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* conveys 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 includes:

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 an overlay window over the *Status Window* (figure 12). 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.

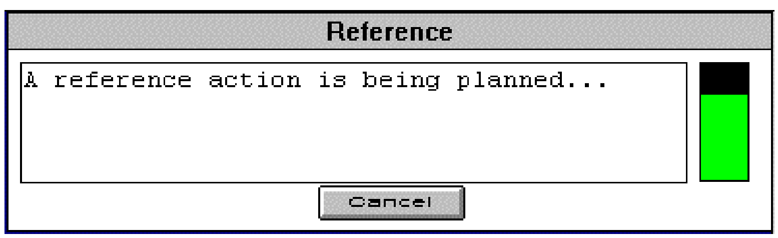

Figure 12**.** Task Status overlay window.

5.1.2.2.3 ***Map Display Window.*** The Map Display Window is located just below the Message Window (figure 10). 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 is 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 <u>Platform Icon.</u> A triangular icon on the map to represents 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.

5.1.2.2.3.3 <u>Threat Vector.</u> During an intruder alarm, the Operator Station graphically portrays a vector showing the bearing to the detected intruder. The color of the displayed vector maintains consistency with the color code scheme described in the *Map Display* section.

5.1.2.2.3.4 <u>Camera Icon.</u> 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 onboard 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 <u>Virtual Points.</u> 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 10). 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 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.* Figure 10 shows a horizontal row of menu buttons below the *Map* and *Video Display Windows*. These buttons allow the guard to input commands to the system; available commands include the following:

Reference:	Reset 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
IDS:	Place a platform in intruder detection mode
Off-line:	Power down the platform and removes it as a resource
Release:	Release control of platform and free Operator Station

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

5.1.2.2.6.1 **<u>Send Overlay Window.</u>** 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 13). A cancel command option is also offered.

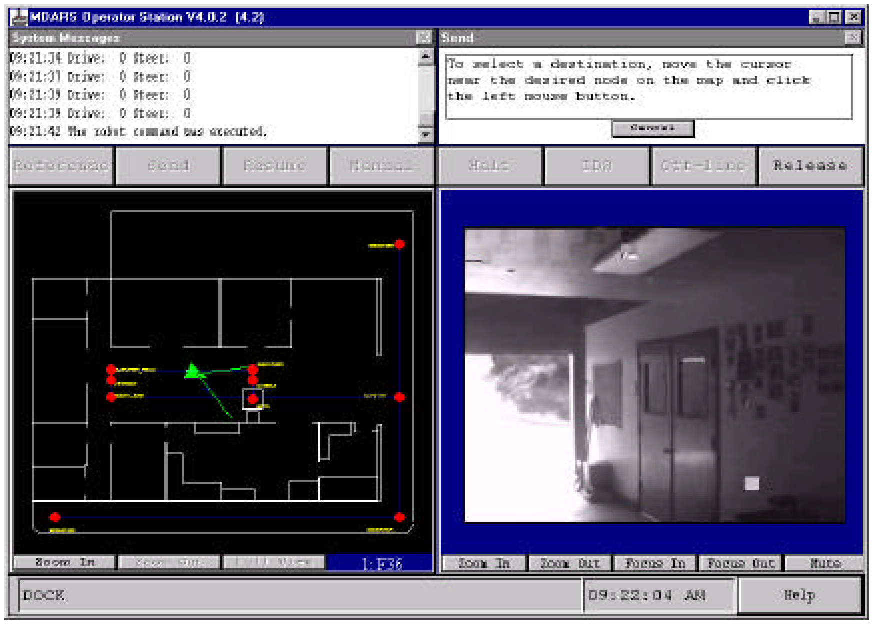

Figure 13**.** Send overlay window.

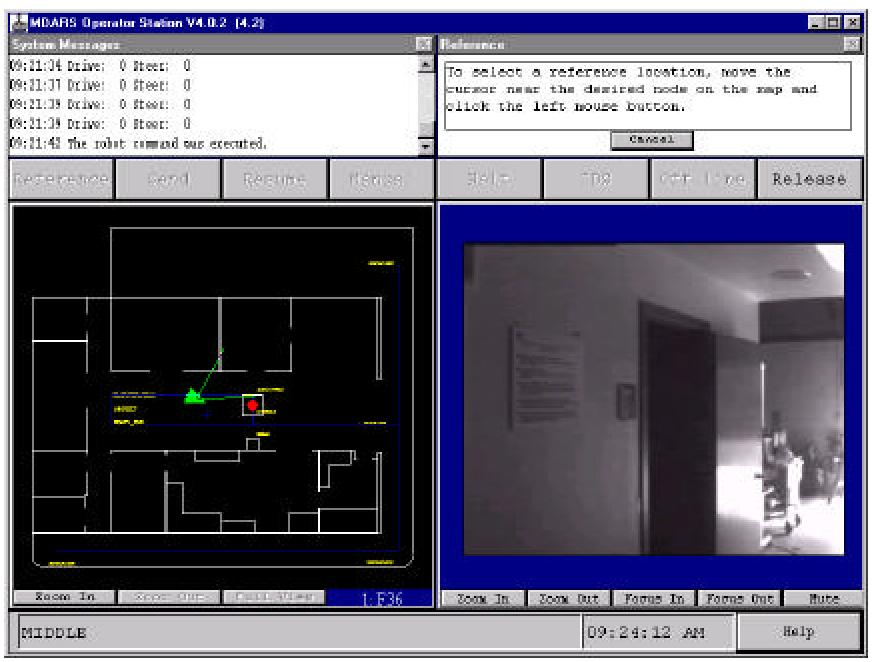

Figure14**.** Reference overlay window.

5.1.2.2.6.2 <u>Reference Overlay Window.</u> 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 14). A cancel command option is also offered.

5.1.2.2.6.3 <u>Off-line Overlay Window.</u> When the ***Off-line*** button is selected, a submenu 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 15). Button menu options ***Power Up***, ***Power Down***, and ***On-line*** are offered.

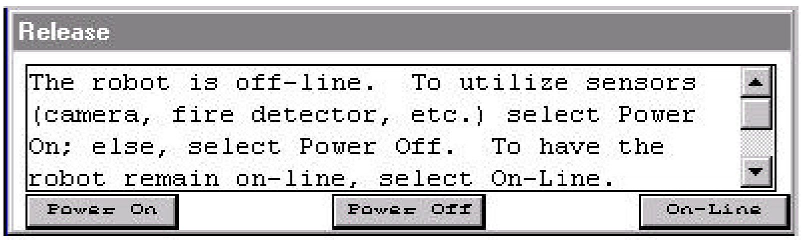

Figure 15**.** Off-line overlay window.

5.1.2.2.6.4 <u>Release Overlay Window.</u> When the ***Release*** button is selected and the guard has left the robot in *Intruder Detection Mode*, a submenu overlay window is displayed. The window text queries the guard about desiring to leave the robot in *Intruder Detection Mode* (figure 16). Button menu options ***IDS On*** and ***IDS Off*** are offered.

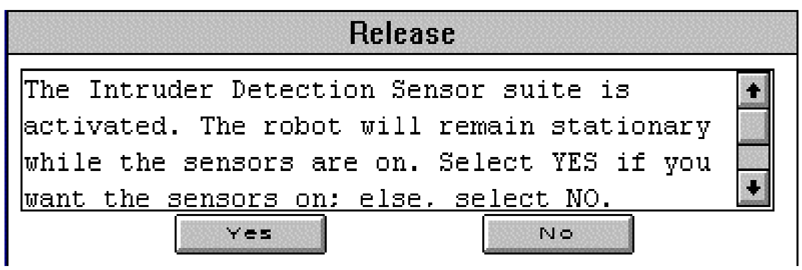

Figure 16**.** Release-IDS overlay window.

5.1.2.2.7 Helpbar Window. The *Helpbar Window* is located at the bottom of the screen below the *Button Menu Window* (figure 10). The *Helpbar Window* displays text messages that provide information about selectable items on the Operator Station display (buttons, virtual nodes, scroll bars). The appropriate text message appears 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 allows 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, trackball) to activate the scroll bar controls to review past messages in the log.

5.1.3.2.1 *Critical System Message Overlay Window.* The System generates critical system messages to inform the guard of an extraordinary situation or event. These messages are posted in an overlay window over the message log (figure 10) 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 time period, 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* conveys detailed information to the guard 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, displaying 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, 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 an 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.

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, 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 autorepeat buttons, located in cells one and two, respectively (figure 10). 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 in motion, the scroll bars will be gray-shaded to indicate deactivation. When the platform is halted, manual scrolling is enabled (the horizontal and vertical scroll bars in the map window will be unshaded). The scroll bars are actuated with a cursor point-and-click input device (i.e., mouse, trackball).

Full View is a momentary button located in cell three (figure 10). 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 10). 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 10). These functions allow the guard to actuate the lens telephoto mechanism on the remote camera, supporting intruder assessment.

Mute is a toggle button (figure 10). Depressing this button deactivates the audio link from the platform to the control station.

Commanding a remote pan-and-tilt mechanism controls the remote camera view. These commands are entered via an external hardware joystick device, see the *User Interface* section. 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. 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 was detected. Automatic intruder tracking occurs when 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* (figure 10). 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 initiates a re-referencing action if a robot is lost. When the **Reference** button is selected, an overlay window appears, 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 display on the map and the names appear in the Reference Overlay Window as the cursor is brought nearer. The guard can re-reference the platform by selecting a re-reference node with the input device. A reference action typically occurs with 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 submenu overlay window appears, 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 display on the map and the names appear in the Interface Assistance Window as the cursor is brought nearer. The desired destination is selected by clicking the input device. The Planner/Dispatcher 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 restores 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., *Manual*) during the time between the *Halt* and *Resume* commands.

5.1.3.6.4 *Manual*. The **Manual** button places the platform in telereflexive mode for direct teleoperation with collision avoidance assistance from platform sensors.

5.1.3.6.5 *Halt.* The **Halt** button immediately stops the motion of the assigned platform.

5.1.3.6.6 IDS. The **IDS** button places 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 continues 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 Supervisor will not send the platform on patrols. The Supervisor *Event Window* lists the *Directed Survey* as a *Non-assignable Event*. If the *Survey* exit option is not selected, the security sensors power down and the platform begins patrolling after release.

5.1.3.6.7 Off-line. The **Off-line** button places a platform in a non-functioning, low-power consumption mode. The platform continues 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 Supervisor will not send the platform will on patrols. The Supervisor *Event Window* lists *Robot Off-line* as a *Non-assignable Event*. If the **Off-line** exit option is not selected, the robot returns to a ready state and the Supervisor sends it on patrols.

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 appears. 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 the guard wants to protect a specific limited area or asset or when the mobility system has suffered a diagnostic failure.

When the ***Release*** button is selected and the guard has left the robot in *Off-line Mode*, an overlay window appears. The window text queries the guard about whether to leave the robot in *Off-line Mode*. This option is appropriate when the guard wants to remove a robot from the system, when, for example, a serious diagnostic failure has been detected.

5.1.3.7 Helpbar Window Controls. The *Helpbar Window* displays text messages that provide information about selectable items on the Operator Station display (buttons, virtual nodes, scroll bars). The appropriate text message appears 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. The following is a list and description of the available commands.

Command: D
Description: Increases, by one level, the Operator Station program debug level.

Command: d
Description: Decreases, by one level, the Operator Station program debug level.

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 also provide a Microsoft mouse-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. Buttons located on the joystick device provide focus in, 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 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."

The Operator Station command line parameters are described below. Note that commands can be entered in either uppercase or lowercase.

Command: -D[level]|-d[level]
Description: Invokes the Operator Station program with diagnostics enabled. Diagnostics are normally disabled. This option enables diagnostics at the designated level, 1 to 3, with 3 being the highest diagnostic level.
Example: `-D2`

Command: -H|-h
Description: Displays Operator Station command line options and program version information, then aborts program.
Example: `-H`

Command: -I{file}|-i{file}
Description: Invokes the Operator Station program with an alternative initialization file designated. The alternative initialization filename is required.
Example: `-Istartup.ini`

Command: -L[file]|-l[file]
Description: Invokes the Operator Station program with packet logging enabled. Communication packets are not normally logged to a file. This option enables packet logging to a file on the hard disk. Filename optional, if not specified, the default name will be used.
Example: `-Ltoday.log`

Command: -N|-n
Description: Invokes the Operator Station program without connecting to the Local Area Network. The Local Area Network is the means for communicating with the other CSCIs that make up the MRHA. The network connection is normally established, this option allows the Operator Station to be run stand-alone.
Example: `-N`

Command: -V|-v
Description: Invokes the Operator Station program with digital video disabled. Digital video from the robot is normally transmitted over the Ethernet link and displayed on the Operator Station. This option allows the Operator Station to be run without the hardware or software necessary to support digital video.
Example: `-V`

5.2 CURRENT STATUS

The Operator Station has been coded, tested, and debugged in Ada for the Windows NT operating system environment in accordance with the Category III functional level described in appendix B. In support of Early User Appraisal Category IV functionality was added on an as-needed priority basis and further debugging was done.

Current documentation status for the Operator Station is as follows:

Software Test Plan, Category I (Laird et al., 1993)

Software Test Description, Category I (Heath-Pastore, 1993)

Software Test Plan, Category II (Laird, 1994)

Software Test Description, Category II (Laird, 1994)

Category II Test Evaluation Log (Laird, 1994)

Abbreviated Test Plan for Technical Feasibility Test (ATC, 1996)

Abbreviated Test Report for Technical Feasibility Test (ATC, 1997)

Interface Design Document (Laird et al., 1998)

Design Document for the Operator Station CSCI (Heath-Pastore, 1998)

User Manual and Training Guide (Grant, 1998)

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/Dispatcher, Operator Station) and the various platforms via a RF modem network. The original MDARS-I communications protocol has been replaced with a generic MRHA protocol, which is UDP-based versus TCP/IP-based, and described in SSC San Diego Technical Document 3039 (Laird et al., 1998). The new MRHA protocol is the same for both interior and exterior platforms, and all references to vendor-specific data have been removed.

The RF modem network consists of one or more Ethernet RF modems stationed within and around the warehouse 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- to 928-MHz 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 robot is also equipped with an Ethernet RF modem and is assigned a unique IP address. Robots 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 communicates with both interior and exterior robots. Robots represent the UDP/IP host while the Link Server represents the client. Hosts listen for connections to clients. Communications to a robot involve establishing a UDP/IP connection between each robot and the Link Server. Once the connection is made, a virtual (serial) data stream is provided for host-robot 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 a separate UDP/IP connection (a second connection to each exterior platform). The DGPS data are 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 the 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 that 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 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 Display, the various Planner/Dispatchers, 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), 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

The Link Server would generate raw data log files (exclusive of the event log maintained by the Supervisor) 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.

Note: This change from the original (TFT) implementation impacts any offline replay capability such as was demonstrated during the Technical Feasibility Testing.

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/TCP connections to the robots and processing network requests from other CSCIs on the MRHA LAN.

It is assumed that the virtual connections to the robots are fragile and may be lost at any time. The Link Server continually checks to see if each robot 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 robot, 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 III- capable with full Ethernet modem and serial modem support.

Current documentation status for the Link Server is as follows:

Software Development Plan (CSC, 1992f)

Software Test Plan (Laird et al., 1993)

Software Test Description (Laird, 1993)

Design Document for the Link Server CSCI (August 1998)

7. PRODUCT ASSESSMENT SYSTEM

The Product Assessment System, depicted in figure 17, 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), R/F inventory tags, and R/F tag interrogator(s)/reader(s). 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
- Manually enter tag item information into the system

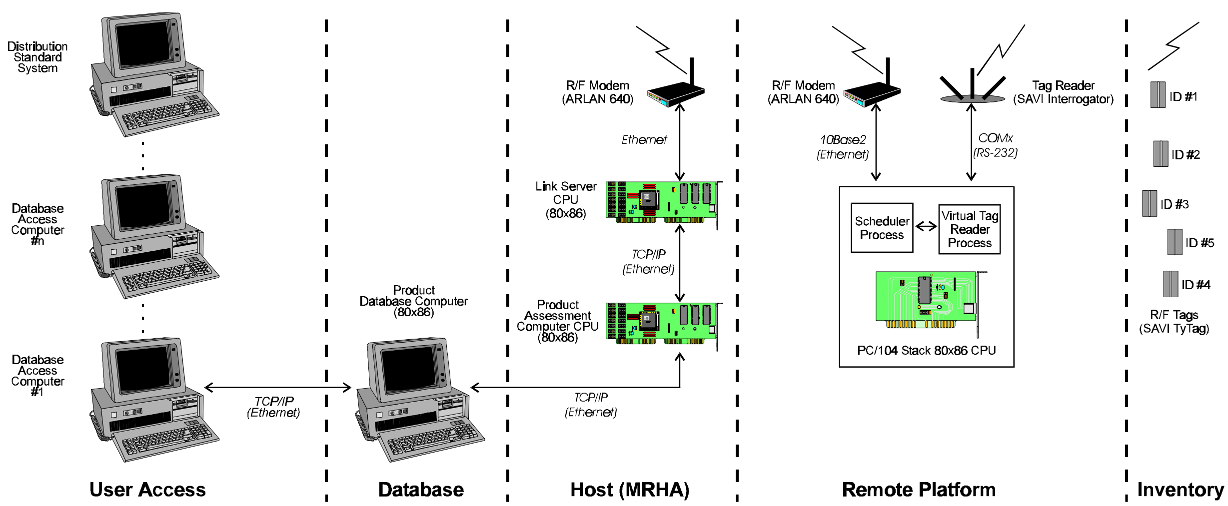

Figure 17. Overall block diagram (PAS).

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

- The **Platform Subsystem:** Located on the remote platform, and consisting of a SAVI Interrogator and a Virtual Tag Reader Computer (VTRC) incorporated in the Scheduler.
- The **Host 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).

- The **User Access Subsystem:** Located at various points within the warehouse environment and consisting of the Database Access Computers (DACs) and, eventually, connection(s) to the Distribution Standard System (DSS).

Under the Product Assessment System (PAS), R/F transponder tags (each with a unique tag ID) are placed on inventory items. 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 manually via data entry on the User Access Subsystem (DACs). Information on tag IDs and their locations is made available to users via ad hoc queries or reports.

The various subsystems of the PAS use different means of communications with one another as depicted in figure 18. The Platform Subsystem collects tag information remotely using a SAVI Interrogator (Tag Reader), and passes data to the Host Subsystem via an ARLAN R/F Modem. The Host, Database, and User Access Subsystems communicate with each other via an Ethernet LAN.

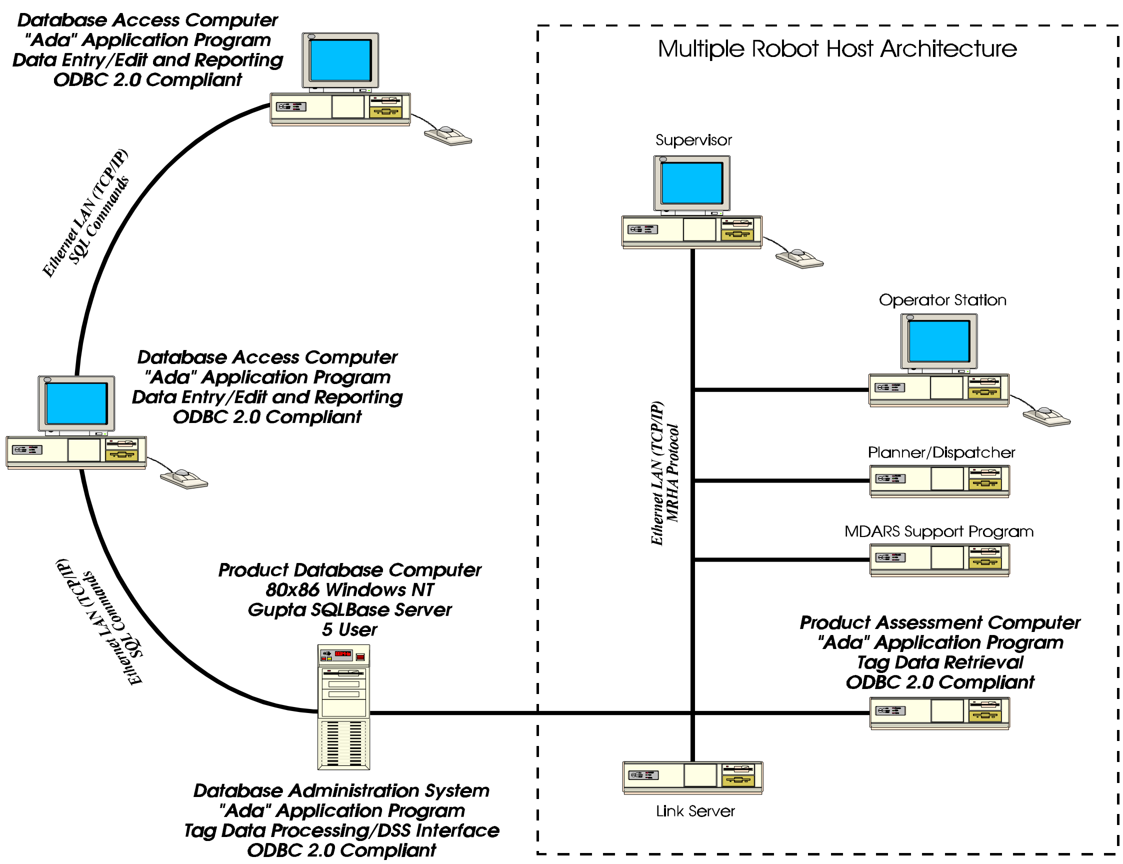

Figure 18. Connectivity diagram (PAS).

7.1 PLATFORM SUBSYSTEM

7.1.1 Platform Subsystem Functions

The functions of the 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 and executing tag reads or interrogations 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 that is collected is from tags that are attached to various items within the warehouse. The tags and tag interrogator currently used are produced by Savi Technology. This system was recommended by the DoD Microcircuit Technology in Logistics Applications (MITLA) Program Management Office (PMO) after a review of the stated MDARS radio frequency identification requirements relative to existing capabilities within the industry.

A tag-read operation is executed when the Virtual Tag Reader Computer (VTRC) on a robot platform instructs the tag interrogator to transmit a wake-up signal and to perform a tag collection. After a tag collection is completed, the VTRC then packetizes the tag data into its on board 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 VTRC on the robot platform to the PAC. The PAC will continue reading tag information in this manner until the *Read Complete* flag on the VTRC indicates that there is no more data.

7.1.2 Platform Subsystem Development

Current Platform subsystem development has reached the completion of Category III capabilities, as outlined in appendix B.

7.2 HOST SUBSYSTEM—PRODUCT ASSESSMENT COMPUTER (PAC)

The PAC resides on an 80486 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 Platform Subsystem

- Inserts tag information into the MDARS database as temporary survey tables

7.2.1.1 Receiving Tag Information. The PAC receives packets of tag information data from the Platform Subsystem, validates the data and logs and displays data errors. The tag information received from the Savi Interrogator consists of the tag ID number and a received signal strength.

7.2.1.2 Insert Into The Database. The PAC connects to the database server and the product database, establishing a handle to the server in preparation for database insertions.

The principal duty of the PAC is to insert tag information data into the survey tables. When the PAC receives tag location information from the tag reader, it performs the following steps to get data from the tag reader into survey tables:

- Creates a temporary table, with a name of the form
- 'TBLTMPYYMMDDHHMMSS' (where YYMMDDHHMMSS is the current year, month, day, hour, minute, and second)
- Receives a data packet from the tag reader, via the link server
- Parses the data packet into SQL-compatible variables
- Compiles the SQL *insert* command
- Binds the SQL-compatible variables to the compiled *insert* command
- Executes the SQL command once for each set of tag data received in the data packet
- Commits the newly inserted data to the database, completing the *insert* transaction
- Renames the temporary table to the form 'TBLSRVYYMMDDHHMMSS' (this name is the form expected by the Database Administration System Update function).

7.2.2 PAC Current Status

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

Current documentation for the PAC is as follows:

MDARS Early User Appraisal Product Assessment Tests (TD 3040, August 1998)

Design Document for the Product Assessment System (PAC) CSCI (MDARS-DD-PAC of 24 February 1995, R3C0 dated 25 September 1997).

MDARS Technical Feasibility Test Results (ATC, 1996)

MDARS Technical Feasibility Test Plan (ATC, 1996)

MDARS Product Assessment System, R. P. Smurlo et al., Association of Unmanned Vehicle Systems, 10 July 1995.

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 an 80486 PC running the Windows NT operating system. The database server software is the multi-user SQLBase from Gupta Technologies, Inc. The PDC communicates with the PAC and one or more DACs using the Ethernet LAN (NetBIOS) 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 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 Gupta Structured Query Language (SQL) base is better suited for the PC network environment.

- Informix On-line is better suited for the medium corporate-flavored environment.

- Oracle 7 is 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 then can criteria and tradeoffs between cost, expandability, and performance be established with a given level of confidence and assurance. The Gupta 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. Finally, during development the Microsoft ODBC32Test program was selected over SQLTalk/Windows as a better interface for database maintenance until a user interface for maintenance can be developed. It should be noted that, if necessary, a change can 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 Gupta SQLBase server software, which accepts and executes SQL commands from the client applications.

- Host the MDARS database (named dbMDARS), which is the database of tag IDs and their locations.

- Host the Database Administration System (DAS) CSCI.

For Category III, database administration and maintenance functions will be performed on the PDC using the ODBC32 test program from Microsoft Corporation. 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 two permanent tables and a variable number of temporary tables. The permanent tables are the user table and the update table. The user table (named tblUser_Data) contains information that may be entered and/or updated by human users. Data are 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 Interrogator
- Strength of signal when tag ID was "read"
- Assessed location of tag ID
- Flags to indicate whether tag ID is Found, Missing or Moved

There are also temporary tables (with names of the form 'TBLSRVYYMMDDHHMMSS'), that the PAC creates. Each table contains data collected from the VTRC on a remote 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 have 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) that performs these functions:
- Incorporate data stored by the PAC into permanent tables.
- Compute assessed locations for each tag ID.
- Mark tag IDs Found, Missing, or Moved as appropriate.
- Make the rows of the user and update tables consistent so that each contains rows for all tag IDs.
- Manually clear any or all tables.

7.3.2.3.1 ***Update Process.*** Figure 19 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.

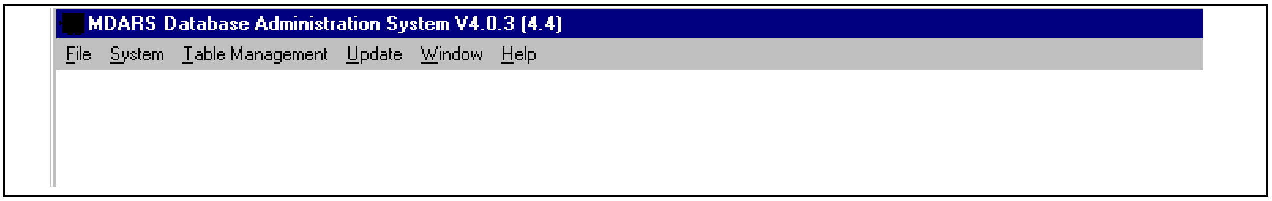

Figure 19. Main Screen (DAS).

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 generally vary by site. When the DAS is running, it schedules and runs the update process at the specified times of day or intervals. If desired, however, the update process can also be started manually. Figure20 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.

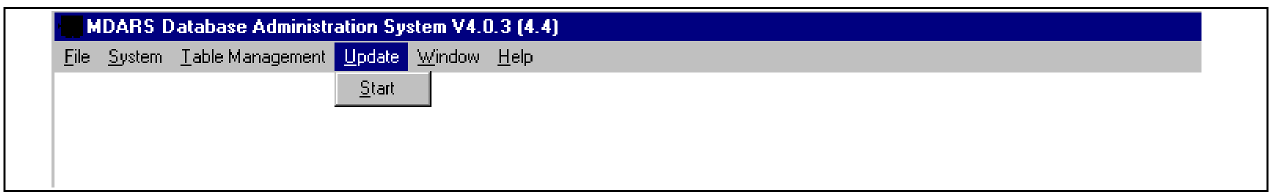

Figure 20. Update pull-down menu (DAS).

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 21shows the System pull-down menu.

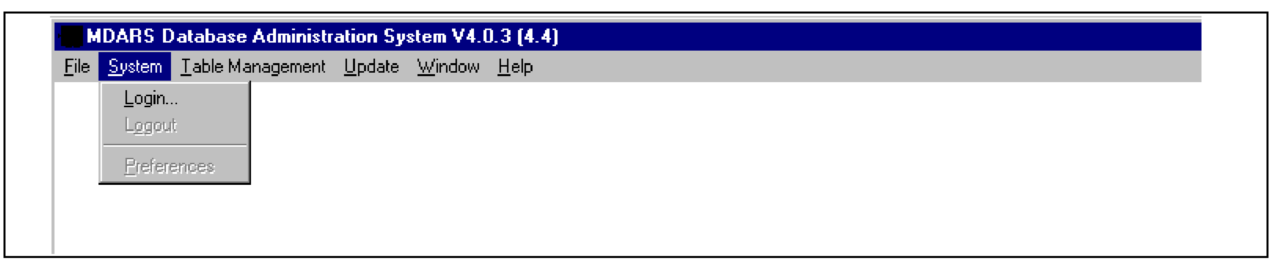

Figure 21. System pull-down menu (DAS).

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 22 shows the Clear Tables selection window.

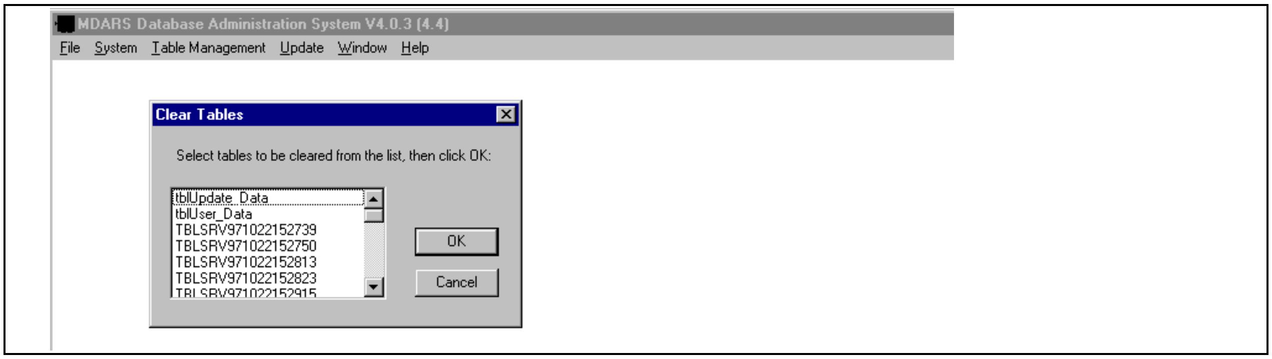

Figure 22. Table Management/Clear Tables function (DAS).

7.3.2.3.3 ***DAS Current Status.*** DAS development has reached the completion of Category III capabilities, as outlined in appendix B.

Current documentation for the DAS is as follows:

MDARS Early User Appraisal Product Assessment Test (TD 3040, August 1998)

Design Document for the Data base Administration System (DAS) CSCI (MDARS-DD-DAS of 14 March 1995, R4C0 dated 23 September 1997)

MDARS Technical Feasibility Test Results (ATC, 1996)

MDARS Technical Feasibility Test Plan (ATC, 1996)

MDARS Product Assessment System, R. P. Smurlo et al., Association of Unmanned Vehicle Systems, 10 July 1995.

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). The future interface of the Product Assessment System (PAS) with the Distribution Standard System (DSS) is being investigated and is planned for Category IV.

7.4.1 DAC Functions (Category III)

Multiple DACs may be attached to the Product Assessment System (PAS). These computers are PC/ATs and are used by inventory management personnel to perform data entry, item reporting, and location functions.

The functions of a DAC are:

Data entry: users can add, modify, and delete data from the database.

Reporting: users can generate item reports and locate items on site maps.

Monitoring Update Status: users can monitor the status of the Database Administration System (DAS) update process.

7.4.1.1 Data Entry, Reporting, and Product Location Functions. Figure 23 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 24 shows the pull-down menu for the "system" functions.

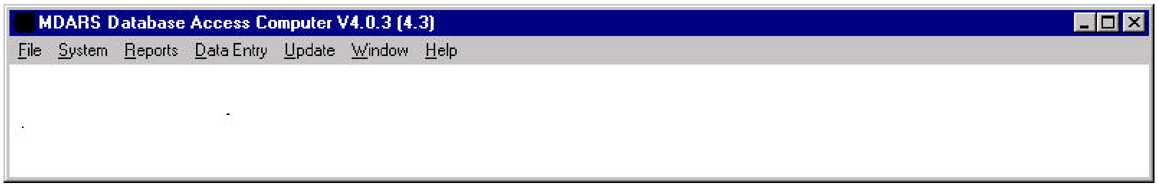

Figure 23. Main Screen (DAC).

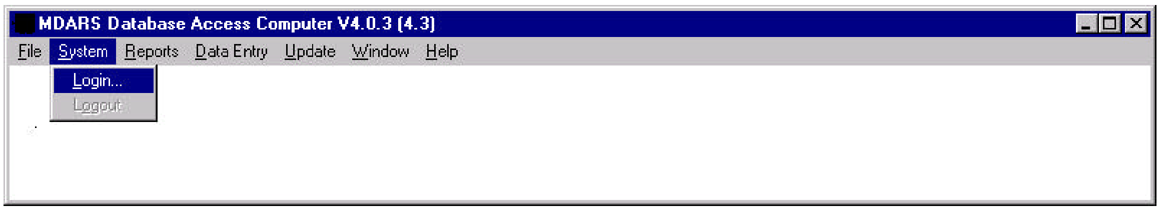

Figure 24. Sample pull-down menu (DAC).

The user has access to context-sensitive help using the various Help menu functions. Figure 25 shows the pull-down menu for Help.

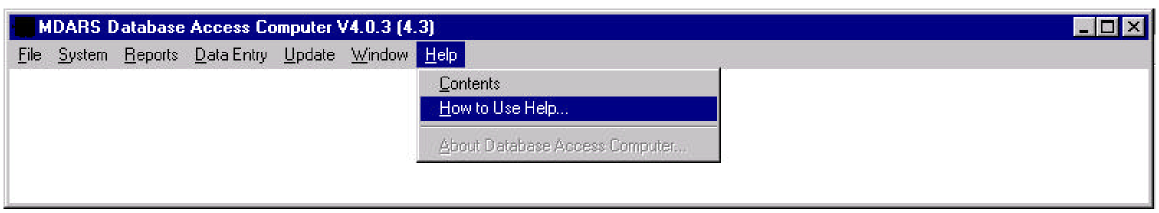

Figure 25. Help pull-down menu (DAC).

7.4.1.1.1 *Data Entry.* Manual entry and editing of data in the database is done using functions under the Data Entry menu option. Data Entry menu suboptions are:

Add Item: adds new tag IDs to the database.

Edit Item: manually updates data associated with a tag ID.

Delete Item: removes tag IDs from the database.

Figure 26 shows the screen layout for the Add Item function, which is similar to the layout for all of these functions.

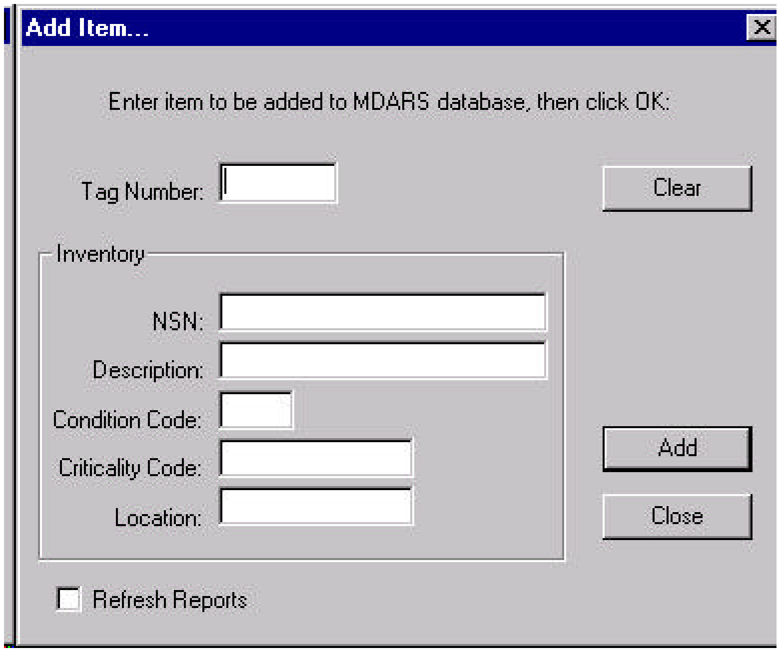

Figure 26. Database/Add Item function (DAC).

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 that are considered unusual and that 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 27 shows 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).

Also available under the Reports menu option is the Locate suboption. Locate allows a user to request the location of a particular database item by tag ID. The item is displayed on a site map and the available location information is listed.

7.4.1.1.2.1 **Database Reports Locate.** Site mapping of an entire report is available (via Locate button on the report) for all available reports.

7.4.1.1.3 ***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.

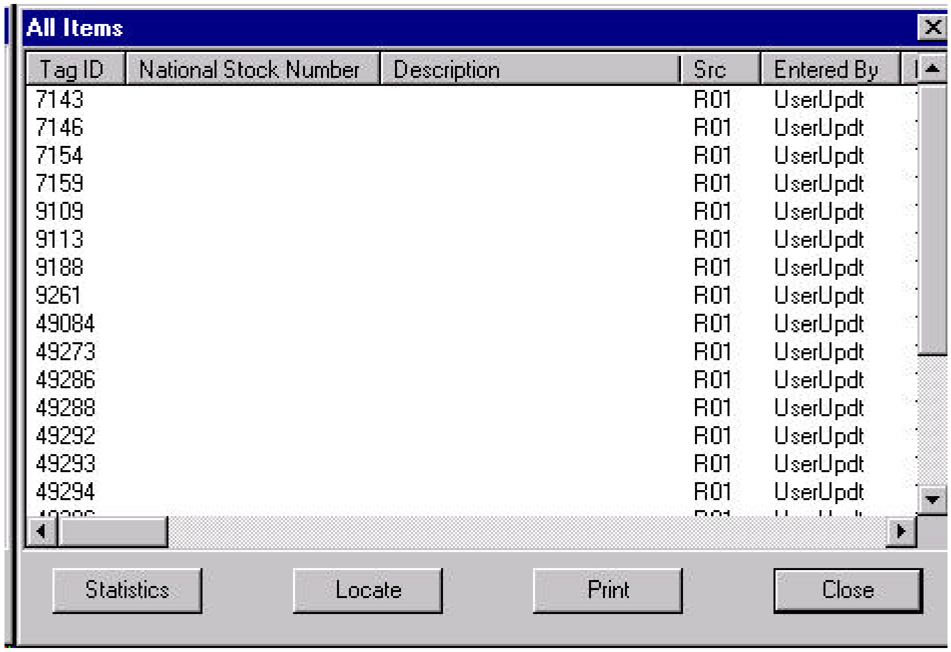

Figure 27. Sample All Items report (DAC).

7.4.2 DAC Functions (Category 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).

7.4.3 DAC Current Status

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

Current documentation for the DAC is as follows:

MDARS Early User Appraisal Product Assessment Tests (TD 3040, August 1998)

Design Document for the Database Access Computer (DAC) CSCI (MDARS-DD-DAC of 14 March 1995, R4C0 dated 23 September 1997).

MDARS Technical Feasibility Test Results (ATC, 1996)

MDARS Technical Feasibility Test Plan (ATC, 1996)

MDARS Product Assessment System, R. P. Smurlo et al., Association of Unmanned Vehicle Systems, 10 July 1995.

8. 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, the LAN consists of the network interface hardware and the supporting software layers as shown in figure 28. The physical layers are implemented using Ethernet hardware configured in a bus topology. Ethernet provides a 10-Mbps network and is widely supported.

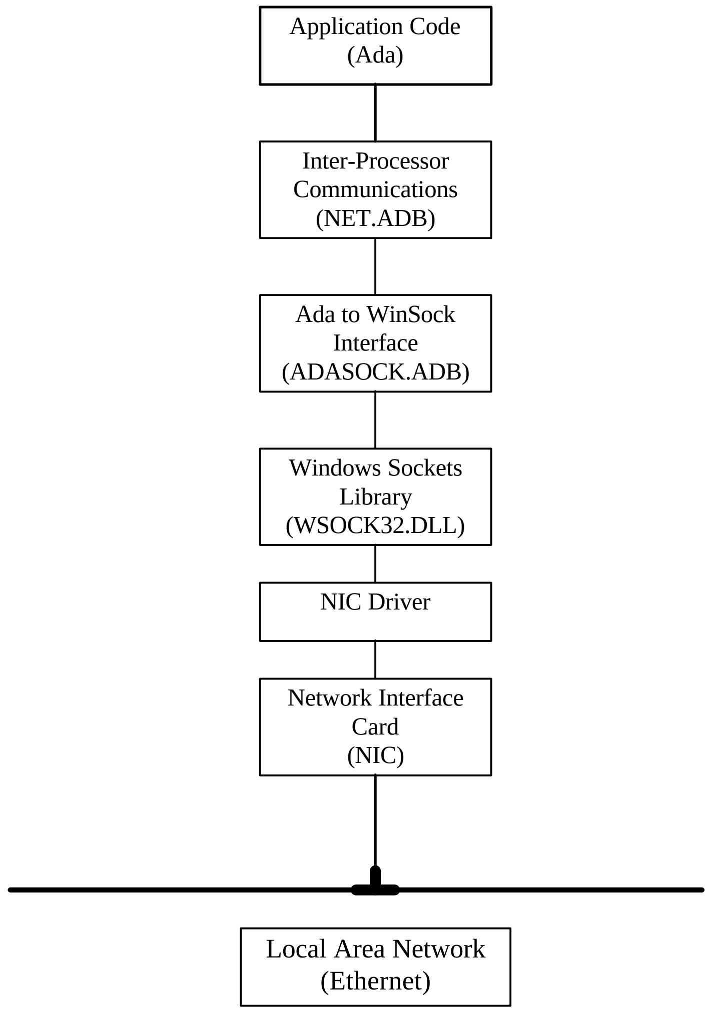

Figure 28. Local Area Network Communications Interface block diagram.

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.

9. REFERENCES

Albert, A. E. 1982. "The Effect of Graphic Input Devices on Performance in a Cursor Positioning Task." *Proceedings of the Human Factors Society 26th Annual Meeting*, pp. 54–58, Santa Monica, CA.

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

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

Computer Sciences Corporation. 1992a. "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.

Computer Sciences Corporation. 1992b. "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.

Computer Sciences Corporation. 1992c. "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.

Computer Sciences Corporation. 1992d. "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.

Computer Sciences Corporation. 1992e. "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.

Computer Sciences Corporation. 1992f. "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.

Computer Sciences Corporation. 1993. "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.

Everett, H. R. 1988. "Security and Sentry Robots." 1998. In *International Encyclopedia of Robotics Applications and Automation.* R. C. Dorf, Ed. John Wiley and Sons, New York, NY.

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

Everett, H. R. 1995. *Sensors for Mobile Robots: Theory and Application.* A. K. Peters, Ltd., Wellesley, MA.

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

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

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

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

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

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

Holland, J., H. R. Everett, and G. A. Gilbreath. 1990. "Hybrid Navigational Control Scheme for Autonomous Platforms," *SPIE Mobile Robots V*, pp. 291–298.

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

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

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

* SSC San Diego
† SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare Systems Center, San Diego. For further information, contact the author(s).

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

Laird, R. T. 1998. "Design Document for the Link Server CSCI of the Mobile Detection Assessment Response System (MDARS) Multiple Resource Host Architecture (MRHA)." SPAWAR Systems Center, San Diego, CA.

Laird, R. T., K. J. Grant, T. A. Heath-Pastore, G. A. Gilbreath, T. T. Tran, R. S. Inderieden, D. Jaffee. 1998. "Design Document for the Common Application Program Components of the Mobile Detection Assessment Response System (MDARS) Multiple Robot Host Architecture (MRHA)." MDARS-DD-COMMON (22 Jun). SPAWAR Systems Center, San Diego, CA.

Laird, R. T., K. J. Grant, T. A. Heath-Pastore, G. A. Gilbreath, T. T. Tran, and R. S. Inderieden. 1998. "Interface Design Document for the Mobile Detection Assessment Response System (MDARS) Multiple Robot Host Architecture (MRHA)." Internal Working Document (21 Aug). SPAWAR Systems Center, San Diego, CA.

Gilbreath, G. A. 1998. "Design Document for the Planner CSCI of the Mobile Detection Assessment Response System (MDARS) Multiple Resource Host Architecture (MRHA)." MDARS-DD-PLANNER-R6C0(13 Jul). SPAWAR Systems Center, San Diego, CA.

Grant, K. J. and S. Elaine. 1996. "Software Test Report for the Supervisor CSCI of the Mobile Detection Assessment and Response System (MDARS)." Contract No. DAAK70-92-D-0005 CDRL Sequence No. A007 (Jun).

Grant, K. J. et al. 1998. "Training Document and User Manual for the Mobile Detection Assessment Response System." Early User Appraisal (Jan).

Grant, K. J. et al. 1998. "Users Manual for the Mobile Detection Assessment Response System." Early User Appraisal (Jan).

Heath-Pastore, T. A. 1998. "Design Document for the Operator Station CSCI for the Mobile Detection, Assessment, and Response System Multiple Resource Host Architecture," MDARS-DD-OPERATOR (3 Aug), SPAWAR System Center, San Diego, CA.

Heath-Pastore, T. A. 1995. "Design Document for the Operator Station CSCI for the Mobile Detection, Assessment, and Response System Multiple Resource Host Architecture" (DOS Version). SPAWAR System Center, San Diego, CA.

Laird, R. T. 1994. "Software Test Plan for the Mobile Detection, Assessment, and Response System Category II." MDARS-STP-CAT-II (May). SPAWAR System Center, San Diego, CA.

Laird, R. T. 1994. "Software Test Description for the Multiple Robot Host Architecture and Remote Platform Architecture of the Mobile Detection, Assessment, and Response System Category II." MDARS-STD-CAT-II (30 May). SPAWAR System Center, San Diego, CA.

Laird, R. T. 1994. "Category II Test Evaluation Log for the Mobile Detection, Assessment, and Response System Multiple Robot Host Architecture." MDARS-TEL-CAT-II (21 Jun). SPAWAR System Center, San Diego, CA.

Sherali, I. I. 1996 (Jun). "Abbreviated Test Plan for the Technical Feasibility Test of the Mobile Detection, Assessment, and Response System Inteior," U. S. Army Test and Evaluation Command, Aberdeen Proving Ground, MD.

Sherali, I. I. 1997 (Apr) "Abbreviated Test Report for the Technical Feasibility Test of the Mobile Detection, Assessment, and Response System Interior," U. S. Army Test and Evaluation Command, Aberdeen Proving Ground, MD.

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

Lozano-Perez, T. and M.A. Wesley. 1979."An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570.

Moravec, H. P. 1987. "Certainty Grids for Mobile Robots." *Proceedings of the Workshop on Space Telerobotics*, January, Jet Propulsion Laboratory, Pasadena, CA.

Rubin, F. 1974. "The Lee Path Connection Algorithm," *IEEE Transactions on Computers*, vol. C-23, no. 9 (Sep), pp. 907–914.

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

Smurlo, R. P. and H. R. Everett. 1992. "Intelligent Security Assessment for a Mobile Robot," *Proceedings of Sensors Expo* (Sep), Chicago, IL.

Winston, P. H. 1984. *Artificial Intelligence.* Addison-Wesley, Reading, MA.

10. BIBLIOGRAPHY

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

Borenstein, J., H. R. Everett, L. Feng, and D. Wehe. 1997. “Mobile Robot Positioning—Sensors and Techniques.” Invited paper for the *Journal of Robotic Systems*, Special Issue on Mobile Robots, vol. 14, no. 4, pp.231–249.

Chun, W. and H. R. Everett. 1995. “Guest Editor’s Note,” *Unmanned Systems*, Magazine of the Association of Unmanned Vehicle Systems, vol. 13, no. 4 (Fall), Washington, DC.

Everett, H. R. 1982. "A Computer Controlled Sentry Robot," *Robotics Age* (Mar/Apr), pp. 22–27.

Everett, H. R. 1982. "A Microprocessor Controlled Autonomous Sentry Robot," Master's Thesis (Oct), Naval Postgraduate School, Monterey, CA.

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

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

Everett, H. R. 1985. "Robotics in the Navy: Industrial Development Efforts," *Robotics Age*, (Nov).

Everett, H. R. 1985. "Robotics in the Navy, Part II: Non-industrial Development Efforts," *Robotics Age* (Dec).

Everett, H. R. 1985. "A Second Generation Autonomous Sentry Robot," *Robotics Age* (1 Apr).

Everett, H. R. 1985. "A Multi-Element Ultrasonic Ranging Array," *Robotics Age* (Jul).

Everett, H. R. 1985. "Robotics Technology: Areas of Needed Research and Development," White Paper presented to ONR/ONT, Serial Number 90G/119 (6 Sep), Naval Sea Systems Command, Washington, DC.

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

Everett, H. R. 1987. "Non-Contact Ranging Systems for Mobile Robots," *Sensors*, vol. 4, no. 4 (Apr), pp. 9–19.

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

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

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

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

Everett, H. R.and G. A. Gilbreath. 1988. "A Supervised Autonomous Security Robot," *Robotics And Autonomous Systems*, North-Holland.

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

Everett, H. R. 1989. "Survey of Collision Avoidance and Ranging Sensors for Mobile Robots," *Robotics And Autonomous Systems*, North- Holland.

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

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

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

Everett, H. R., G. A. Gilbreath, and J. M. Holland. 1990. "Hybrid Navigational Control Scheme," *SPIE Mobile Robots V* (Nov), pp. 291–298.

Everett, H. R., G. A. Gilbreath, G. A., and R. T. Laird. 1992. "Multiple Host Robotic Architecture," NRaD* Technical Note† 1710 (Feb). Naval Command, Control and Ocean Surveillance Center RDT&E Division, San Diego, CA.

Everett, H. R., G. A. Gilbreath, R. T. Laird, and T. A. Heath. 1992. "Multiple Robot Host Architecture." NRaD* Technical Note† 1710 (Nov), Revision 1. Naval Command, Control and Ocean Surveillance Center RDT&E Division, San Diego, CA.

Everett, H. R., G. A. Gilbreath, R. T. Laird, T. A. Heath-Pastore, and P. J. Bresnowitz. 1993. "Multiple Host Robotic Architecture." NRaD* Technical Note† 1710, Revision 2 (Dec), Naval Command, Control and Ocean Surveillance Center RDT&E Division, San Diego, CA.

* SSC San Diego
† SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare Systems Center, San Diego. For further information, contact the author(s).

Everett, H. R., G. A. Gilbreath, R. T. Laird, T. A. Heath, R. P. Smurlo. 1995. "Multiple Robot Host Architecture," NRaD* Technical Note† 1710, Revision 3 (Jul, draft), Naval Command Control and Ocean Surveillance Center RDT&E Division, San Diego, CA.

Everett, H. R., G. A. Gilbreath, T. A. Heath, and R. T. Laird. 1993. "Coordinated Control of Multiple Security Robots," *SPIE Mobile Robots VIII* (Sep), pp. 292–305, Cambridge, MA.

Everett, H. R., G. A. Gilbreath, T. A. Heath-Pastore, and R. T. Laird. 1994. "Controlling Multiple Security Robots in a Warehouse Environment*." AIAA-NASA Conference on Intelligent Robots,* March, Houston, TX.

Everett, H. R., D. W. Gage, G. A. Gilbreath, R. T. Laird, and R. P. Smurlo. 1994. "Real-world Issues in Warehouse Navigation," *SPIE Mobile Robots IX*, vol. 2352 (Nov), pp. 249–259, Boston, MA.

Everett, H. R. 1995. *Sensors for Mobile Robots: Theory and Application.* 1995. A. K. Peters, Ltd., Wellesley, MA.

Everett, H. R. 1995. "Theory and Application of Optical Gyroscopes: Part I," *Sensors* (Jul), pp. 54–59.

Everett, H. R., T. A. Heath-Pastore, and R. T. Laird. 1995. "Mobile Detection Assessment Response System,"*Unmanned Systems,* Magazine of the Association of Unmanned Vehicle Systems, vol. 13, no. 4 (Summer),Washington, DC.

Everett, H. R. and W. Chun. 1995. "Guest Editor's Note," *Unmanned Systems*, Magazine of the Association of Unmanned Vehicle Systems, vol. 13, no. 3 (Summer), Washington, DC.

Everett, H. R. 1995. "Theory and Application of Optical Gyroscopes: Part II," *Sensors* (Aug), pp. 26–32.

Everett, H. R.and D. W. Gage. 1995. "Navigational Referencing Techniques for Real-World Environments," *SPIE Mobile Robots X*, vol. 2591 (Oct), Philadelphia, PA.

Everett, H. R. 1995. "Understanding Ultrasonic Ranging Sensors," *The Robotics Practitioner* (Fall).

Everett, H. R. 1996. "Autonomous Navigation on a Shoestring Budget," *The Robotics Practitioner* (Winter), pp. 15–23.

Everett, H. R. 1996. "Representational World Modeling," *The Robotics Practitioner* (Spring).

---

* SSC San Diego
† SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare System Center, San Diego. For further information, contact the author(s).

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

Everett, H. R. and D. W. Gage. 1996. "A Third Generation Security Robot," *SPIE Mobile Robot and Automated Vehicle Control Systems*, vol. 2903 (Nov), pp. 20–21, Boston, MA.

Everett, H. R. 1998. “Breaking Down the Barriers,” *Unmanned Vehicles*, vol. 3, no. 1 (Feb), pp. 18–20, Shephard Press.

Feng, L., J. Borenstein, and H. R. Everett. 1994. "Where am I: Sensors and Methods for Mobile Robot Positioning," Technical Report UM-MEAM-94-21 (Dec). University of Michigan, Ann Arbor, MI.

Gage, D. W., H. R. Everett, R. T. Laird, and T. A. Heath-Pastore. 1995. "Navigating Multiple Robots in Semi-Structured Environments," *ANS 6th Topical Meeting on Robotics and Remote Systems*, February, Monterey, CA.

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

Heath-Pastore, T. A. and H. R. Everett. 1994. "Coordinated Control of Interior and Exterior Autonomous Platforms," *ISRAM 94, Fifth International Symposium on Robotics and Manufacturing*, August, Maui, HI.

Holland, J. M., A. Martin, R. P. Smurlo, and H. R. Everett. 1995. "MDARS Interior Platform," Association of Unmanned Vehicle Systems, *22nd Annual Technical Symposium and Exhibition (AUVS '95)*, July, Washington, DC.

Hughes, T. W., H. R. Everett et al. 1990. "Issues in Mobile Robotics: Unmanned Ground Vehicle Program Teleoperated Vehicle," *SPIE Mobile Robots V* (Nov), pp. 587–597.

Laird, R. T., G. A. Gilbreath, T. A. Heath-Pastore, H. R. Everett, R. S. Inderieden, and K. Grant. 1995. "MDARS Multiple Robot Host Architecture," Association of Unmanned Vehicle Systems, *22nd Annual Technical Symposium and Exhibition (AUVS '95)*, July, Washington, DC.

Laird, R. T., H. R. Everett, and G. A. Gilbreath. 1993. "A Host Architecture for Multiple Robot Control," *ANS Fifth Topical Meeting on Robotics and Remote Handling* (Apr), pp. 531–538, Knoxville, TN.

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

* SSC San Diego

† SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare Systems Center. For further information, contact the author(s).

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

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

Smurlo, R.P. and H. R. Everett. 1994. "Intelligent Sensor Fusion for a Mobile Security Robot," *Sensors* (Mar), pp. 18–28.

Smurlo, R. T and H. R. Everett. 1994. "Robotic Inspection of Hazardous Waste Storage Sites: A Navigation Technology Assessment." NRaD[*] Technical Note[†] 1739 (Jun), Naval Command Control and Ocean Surveillance Center RDT&E Division, San Diego, CA.

Smurlo, R. P., R. T. Laird, H. R. Everett, R. S. Inderieden, S. Elaine, and D. M. Jaffee. 1995. "MDARS Product Assessment System," Association of Unmanned Vehicle Systems, *22nd Annual Technical Symposium and Exhibition (AUVS '95)*, July, Washington, DC.

[*] SSC San Diego
[†] SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare Systems Center, San Diego. For further information, contact the author(s).

APPENDIX A

TN-1710[†] DOCUMENT CONFIGURATION CONTROL

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

[†] SSC San Diego Technical Notes (TNs) are working documents and do not represent an official policy statement of the Space and Naval Warfare Sytems Center. For further information, contact the author(s).

APPENDIX B

SYSTEM FUNCTIONAL REQUIREMENTS

Required functions for all system Hardware Configuration Items (HWCIs) are normalized with respect to the Supervisor into the following four 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	04/01/97
PIP	Pre-planned Improvement	N/A	TBD

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

(NS)	Not Started
(IP)	In Progress
(OH)	On Hold
(CP)	Completed

Updated: 08/17/98 at 9:47 AM.

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	NS	ON HOLD
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 reposts 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 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
II	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Display	1	Use new Scheduler status bit to display when the platform is actually performing a tag read operation.	SW-1d	NS	N/A	CP	9/26/96
III	"	2	Read and display tag reader status.	SW-1d	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-12w*	NS	N/A	CP	1/27/97
III	"	2	Monitor Planner for completion using task status.	SW	N/A	N/A	OH	
III	"	3	Assign new Planner to Operator upon request.	SW	N/A	N/S	CP	1/27/97
III	User Interface	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
IV	Initialization	1	Automatically reference robots at charger on startup if at dock.	SW	NS	N/A	NS	
	"	2	Site-specific emergent work	SW	NS	N/A	NS	
IV	Display	1	Display one-line help message in help bar when mouse moves over menu buttons.	SW-3D	NS	N/A	NS	
IV	"	2	Provide capability to control/display maps of up to 8 robots.	SW-4w	NS	N/A	NS	
IV	"	3	Site-specific emergent work	SW	NS	N/A	NS	
IV	Command	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Event Processing	1	Display multiple events per platform	SW	NS	N/A	NS	
IV	"	2	Allow assignment of Operator w/o Planner, assign Planner later.	SW-1W	NS	N/A	NS	
IV	"	3	Generate messages for Operator when higher priority event is waiting.	SW	NS	N/A	NS	
IV	"	4	Site-specific emergent work	SW	NS	N/A	NS	
IV	"	5	Log pass-through events/conditions from other CSCI's.	SW	NS	N/A	CP	1/27/97
IV	Housekeeping	1	Provide support for multiple Link Servers.	SW	NS	N/A	NS	
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	NS	
IV	"	5	Provide on-line (windows) help capability.	SW-3.5w*	NS	N/A	CP	7/15/97
IV	"	6	Site-specific emergent work	SW	NS	N/A	NS	
IV	User Interface	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	"	2	Provide full printing capability.	SW-2w	NS	N/A	NS	
IV	"	3	Provide on-line (Windows) help capability	SW	N/A	N/A	NS	
PIP	Initialization	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Display	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Command	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Event Processing	1	ICIDS integration	SW	NS	N/A	NS	

PIP	Housekeeping	1	ICIDS integration	SW	NS	N/A	NS	
PIP	User Interface	1	ICIDS integration	SW	NS	N/A	NS	

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

•

Category	Major Function	#	PLANNER/DISPATCHER* 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	Display	None	N/A	N/A	N/A	N/A	N/A	N/A
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	User Interface	None	N/A	N/A	N/A	N/A	N/A	N/A
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	Display	None	N/A	N/A	N/A	N/A	N/A	N/A
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
II	Directed Movement	None	N/A	N/A	N/A	N/A	N/A	N/A
II	User Interface	None	N/A	N/A	N/A	N/A	N/A	N/A
II	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Initialization	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Display	1	Convert S/W to Windows NT Operating System to alleviate memory restrictions.	SW-8w*	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-3w	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-3d	NS	N/A	CP	6/17/96
III	"	2	Support task status.	SW	N/A	N/A	CP	1/27/97
III	User Interface	None	N/A	N/A	N/A	N/A	N/A	N/A

III	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
IV	Initialization	1	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	Display	1	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	Random Patrols	1	Allow disabling of paths.	SW	NS	N/A	CP	6/3/96
	“	2	Site-specific emergent work	SW	NS	N/A	N/A	N/A
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	“	3	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	“	4	Implement automatic wall following for collision avoidance.	SW-3d	NS	N/A	N/A	N/A
IV	“	5	Implement automatic lateral post detection for collision avoidance.	SW-3d	NS	N/A	N/A	N/A
IV	“	6	Improve collision avoidance algorithm to include modeling of added sensors to detect pallets.	SW	N/A	N/A	NS	
IV	Directed Movement	1	Mix virtual and unrestricted paths.	SW	NS	N/A	N/A	N/A
IV	“	2	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	“	3	Support unrestricted plans.	SW	N/A	N/A	IP	
IV	Housekeeping	1	Degrade gracefully/recover when another CSCI or LAN cable is disconnected	SW	NS	N/A	N/A	N/A
IV	“	2	Determine if any path segment has not been recently traversed	SW	NS	N/A	N/A	N/A
IV	“	3	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	User Interface	1	Provide on-line (windows) help capability	SW	NS	N/A	N/A	N/A
		2	Site-specific emergent work	SW	NS	N/A	N/A	N/A
IV	Diagnostics	1	Site-specific emergent work	SW	NS	N/A	N/A	N/A
PIP	Initialization	1	ICIDS integration	SW	NS	N/A	N/A	N/A
PIP	Display	1	ICIDS integration	SW	NS	N/A	N/A	N/A
PIP	Random Patrols	1	ICIDS integration	SW	NS	N/A	N/A	N/A
PIP	Housekeeping	1	ICIDS integration	SW	NS	N/A	N/A	N/A
PIP	User Interface	1	ICIDS integration	SW	NS	N/A	N/A	N/A
PIP	Diagnostics	1	ICIDS integration	SW	NS	N/A	N/A	N/A

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

Category	Major Functions	#	OPERATOR* 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 scheduler 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 Scheduler 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
II	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Initialization	1	Check for access to robot configuration information (maps and databases) during initialization process.	SW-3d	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-1w	NS	N/A	CP	8/25/96
III	"	2	Implement pop-up message windows for process status information.	SW-1.5w	NS	N/A	CP	8/25/96
III	"	3	Display platform video on Operator display.	SW-1w	NS	N/A	IP	
III	Command	1	Provide Release options menu when the robot is released in a resumable state.	SW-1w	NS	N/A	N/A	N/A
III	"	2	Provide Release options menu when the robot is released in intruder detection mode.	SW-1w	NS	N/A	CP	8/25/96
III	"	3	Modify robot commands for new Cybermotion memory maps and computer numbers.	SW-1w	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-2d	NS	N/A	CP	1/27/97
III	"	2	Convert S/W to Windows NT Operating System to alleviate memory restrictions.	SW-10w*	NS	N/A	CP	9/26/96
III	"	3	Improve software maintainability by converting software to Ada programming language.	SW-10w*	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-4d	NS	N/A	CP	8/15/96
III	"	6	Incorporate video link status communication with Supervisor	SW-4d	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-4d	NS	N/A	CP	1/27/97
III	"	2	Provide device control of camera pan, tilt, center functions	SW-1w	NS	N/A	CP	1/27/97
III	"	3	Provide device control of telereflexive functions	SW-2w	NS	N/A	CP	1/27/97
III	"	4	Provide Operator Station immunity to superfluous input from the normal operating mode input devices.	SW-2d	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-3d	NS	N/A	CP	8/25/96
IV	Initialization	1	Site-specific emergent work	SW	NS	N/A	N/A	N/A
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	Site-specific emergent work	SW	NS	N/A	NS	
IV	“	5	Converge on final verbiage for text and status information	SW	NS	N/A	NS	
IV	“	6	Implement “help line” window to display information on user controls when the cursor is within control boundaries.	SW-1w	NS	N/A	IP	
IV	“	7	Implement site-speciifc X-Y location look-up table.	SN	NS	N/A	NS	
IV	“	8	Change Operator Station designation to Directed Control Station (DCS).	SW	NS	N/A	NS	
IV	“	9	Add Operational Time Remaining Status Element.	SW	NS	N/A	NS	
IV	Command	1	Incorporate new-object-encountered detection feature (awaiting User Group feedback).	SW	NS	N/A	N/A	N/A
IV	“	2	Request an unrestricted path planning operation, upon user command.	SW	NS	N/A	NS	
IV	“	3	Site-specific emergent work	SW	NS	N/A	NS	
IV	“	4	Incorporate Request Platform function.	SW	NS	N/S	NS	
IV	Housekeeping	1	Pass information to Supervisor for logging.	SW	NS	N/A	NS	
IV	“	2	Request and upload paths from Planner	SW	NS	N/A	N/A	N/A
IV	“	3	Forward packets intended for another MRHA module (planner complete packets to supervisor after operator has released) - (Pending CCB review)	SW	NS	N/A	NS	
IV	“	4	Degrade gracefully/recover when another CSCI or LAN cable is disconnected	SW	NS	N/A	CP	1/27/97
IV	“	5	Site-specific emergent work	SW	NS	N/A	NS	
IV	User Interface	1	Provide on-line (windows) help.	SW	NS	N/A	NS	
IV	“	2	Site-specific emergent work	SW	NS	N/A	NS	
PIP	Initialization	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Display	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Command	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Housekeeping	1	ICIDS integration	SW	NS	N/A	NS	
PIP	User Interface	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Diagnostics	1	ICIDS integration	SW	NS	N/A	NS	
PIP	“	2	Perform detailed platform diagnostic analysis when necessary	SW	NS	N/A	NS	
PIP	“	3	Site-specific emergent work	SW	NS	N/A	NS	

** POC for the Operator is Tracy Heath Pastore - (619) 553-3674*

Category	Major Functions	#	PRODUCT ASSESSMENT COMPUTER* Functionality / Requirement	HW or SW	DOS Status	DOS CP Date	NT Status	NT CP Date
I	N/A	None	N/A	N/A	N/A	N/A	N/A	N/A
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 Information 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-1d	NS	N/A	CP	6/1/97
III	Display	1	Convert S/W to Windows NT Operating System to alleviate memory restrictions	SW-8w*	NS	N/A	CP	6/1/97
III	Housekeeping	1	Modify robot commands for new Cybermotion memory maps and computer numbers	SW-2w	NS	N/A	CP	6/1/97
III	Tag Information Processing	1	Modify PAC interface to database to accommodate new Database format	SW-1d	NS	N/A	CP	6/1/97
III	"	3	Write Ada pragmas to C/API and write functions in Ada	SW-2w	NS	N/A	CP	6/1/97
IV	Initialization	1	Site-specific emergent work	SW	NS	N/A	N/A	
IV	Display	1	Site-specific emergent work	SW	NS	N/A	N/A	
IV	Housekeeping	1	Degrade gracefully/recover when another CSCI or LAN cable is disconnected	SW	NS	N/A	NS	
IV	Tag Information Processing	1	Modify PAC to communicate reliably with up to 8 robots simultaneously	SW-2d	NS	N/A	NS	
IV	"	2	Site-specific emergent work	SW	NS	N/A	N/A	
IV	"	1	Site-specific emergent work	SW	NS	N/A	N/A	

** POC for the Product Assessment Computer is Robin Laird - (619) 553-3667*

Category	Major Functions	#	DATABASE ACCESS COMPUTER* Functionality / Requirement	HW or SW	DOS Status	DOS CP Date	NT Status	NT CP Date
I	N/A	None	N/A	N/A	N/A	N/A	N/A	N/A
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 Information 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	Initialization	None	N/A	N/A	N/A	N/A	N/A	N/A
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-2d	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 Information 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
IV	Initialization	1	Site-specific emergent work	SW	N/A	N/A	NS	
IV	Display	1	Site-specific emergent work	SW	N/A	N/A	NS	
IV	User Interface	1	Provide hand held data entry device support to be used for tagging inventory and entering data into the database (Launch Station)	SW	N/A	N/A	NS	
IV	“	2	Improve upon user interface based on solicited user feedback	SW-2w	N/A	N/A	NS	
IV	“	3	Provide final tag programming capability	SW/HW	N/A	N/A	NS	
IV	“	4	Site-specific emergent work	SW	N/A	N/A	NS	
IV	Tag Information Processing	1	Provide reports in formats usable by other automated depot systems (SDS/DSS)	SW	N/A	N/A	NS	
IV	“	2	Site-specific emergent work	SW	N/A	N/A	NS	
PIP	Initialization	None	N/A	N/A	N/A	N/A	NS	
PIP	Display	None	N/A	N/A	N/A	N/A	NS	
PIP	User Interface	None	N/A	N/A	N/A	N/A	NS	

** POC for the Database Access Computer is Rebecca Inderieden - (619) 553-4760/Doriann Jaffee - (619) 553-6915*

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 Information 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 Information 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
IV	User Interface	1	Provide administration/maintenance functions: - Adding/removing users - Changing passwords - Database backups - Database "crash" recovery	SW	N/A	N/A	NS	
IV	Tag Information Processing	1	Use triangulation or orther techniques to better localize tags	SW	N/A	N/A	NS	
IV	"	2	Provide interface to other automated depot systems to pass data to/from those systems	SW	N/A	N/A	NS	
PIP	Tag Information Processing	None	ICIDS Integration	N/A	N/A	N/A	NS	

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

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: a) incoming/outgoing robot message and network packet information b) time/date and program version information c) robot communication status d) package initialization debug messages 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 robot 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 robots 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 robot in the system	SW	CP	7/2/93	CP	8/26/96
I	"	2	Periodically request status from each robot 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 robot 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 the K2A robots	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 robot 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 robot and releasing the video channel assigned to a robot, manage robot 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 robot reports the same platform ID	SW	CP	12/7/93	CP	1/2797
II	"	3	Pass robot ID information to other CSCIs on LAN	SW	CP	3/4/94	CP	8/26/96
II	Display	None	N/A	N/A	N/A	N/A	N/A	N/A
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 robots identified in configuration file (FIX)	SW	CP	2/23/94	CP	8/26/96
II	Emergency Halt	None	N/A	N/A	N/A	N/A	N/A	N/A
II	Data Log/Eavesdrop	1	Log robot message traffic to individual files based upon robot ID	SW	CP	4/28/94	CP	9/26/96
II	"	2	Provide filtering capabilities for robot 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 (FIX)	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
II	Diagnostics	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Initialization	1	Provide support for multiple serial I/O cards (2 cards minimum, 8 robots)	SW-3d	N/A	N/A	CP	8/26/96
III	Display	1	Convert S/W to Windows NT Operating System to alleviate memory restrictions	SW-10w*	N/A	N/A	CP	8/26/96
III	Message Routing	1	Verify Robot status integrity when received from robot	SW-3d	N/A	N/A	CP	8/26/96
III	"	2	Improve communications throughput by upgrading to new Ethernet-addressable modems	SW-4w	N/A	N/A	CP	8/26/96
III	"	3	Integrate robot communications link with control station remoting electronics	SW-1w	N/A	N/A	CP	6/1/97
III	"	4	Implement Cybermotion "#" abbreviated messages	SW-1w	NS	N/A	CP	8/26/96
III	Status Polling	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Emergency Halt	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Data Log/Eavesdrop	None	N/A	N/A	N/A	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	User Interface	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Diagnostics	1	Quantify modem communications integrity (i.e., reliability information).	SW-3d	IP	CP	CP	6/1/97
IV	Initialization	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Display	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Message Routing	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Status Polling	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	"	2	Send robot status to each CSCI at regular intervals.					
IV	Emergency Halt	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Data Log/Eavesdrop	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	"	2	Site-specific emergent work	SW	NS	N/A	NS	
	User Interface	1	Provide on-line (windows) help	SW-2w*	NS	N/A	NS	
IV	"	2	Site-specific emergent work	SW	NS	N/A	N/A	
IV	Diagnostics	1	Site-specific emergent work	SW	NS	N/A	N/A	
PIP	Initialization	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Display	1	ICIDS integration	SW	NS	N/A	NS	

PIP	Status Polling	1	Provide variable polling rate according to predefined operational (robot) modes	SW	NS	N/A	NS	
PIP	“	2	ICIDS integration	SW	NS	N/A	NS	
PIP	Data Log/Eavesdrop	1	Provide sophisticated network packet and robot message filtering (i.e., field filters, log bad data)	SW	NS	N/A	NS	
PIP	“	2	ICIDS integration	SW	NS	N/A	NS	
PIP	Status Polling	1	Provide variable polling rate according to predefined operational (robot) modes	SW	NS	N/A	NS	
PIP	Housekeeping	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Diagnostics	1	ICIDS integration	SW	NS	N/A	NS	

** POC for the Link Server is Robin Laird - (619) 553-3667*

Category	**Major Function**	**#**	**Scheduler* Functionality / Requirement**	**HW or SW**	**DOS Status**	**DOS CP Date**	**NT Status**	**NT CP Date**
III	Command	1	Patrol Mode	SW	CP	1/1/94	CP	8/26/96
III	“	2	Tele-reflexive mode	SW	CP	1/1/94	CP	8/26/96
III	“	3	Resume Mode	SW	CP	1/1/94	CP	8/26/96
III	“	4	Survey Mode	SW	CP	1/1/94	CP	8/26/96
III	“	5	Inventory Mode	SW	CP	1/1/94	CP	8/26/96
III	“	6	Docking Mode	SW	CP	1/1/94	CP	8/26/96
III	“	7	Emergency Halt Mode	SW	CP	1/1/94	CP	8/26/96
III	“	8	Off-line Mode	SW	CP	1/1/94	CP	8/26/96
III	“	9	Diagnostic Mode	SW	CP	1/1/94	CP	8/26/96
III	“	10	Selective Halt Mode	SW	CP	1/1/94	CP	8/26/96
III	“	11	Broadcast mode enabled at startup to broadcast platform position and heading to all computers onboard the robot.	SW	CP	1/1/94	CP	8/26/96
III	“	12	Status block variables updated at a 2 Hz rate or faster	SW	CP	1/1/94	CP	8/26/96
IV	“	13	Manual Mode	SW	CP	1/1/94	CP	8/26/96

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

Category	Major Function	#	NBSC* Functionality / Requirement	HW or SW	DOS Status	DOS CP Date	NT Status	NT CP Date
I	Command	1	Mode 0 - Idle mode (sonars not fired) Mode 1 - 9 sonars fired, stop robot if any range less than the specified stop range Mode 2 - 9 sonars fired, do not cause the robot to stop	SW	IP	1/1/94	N/A	N/A
I	“	2	Current robot position and heading along with current sonar readings stored in a contiguous array of bytes	SW	IP	1/1/94	N/A	N/A
I	“	3	A history buffer that stores all the sonar ranges and proximity readings for the last 10 feet of path traversed	SW	IP	1/1/94	N/A	N/A

** POC for the NBSC is John Holland - (703)562-7626*

Category	Major Functions	#	TAG READER COMPUTER* Functionality / Requirement	HW or SW	DOS Status	NT CP Date	NT Status	NT CP Date
I	N/A	None	N/A	N/A	N/A	N/A	N/A	N/A
II	Initialization	1	Provide blackboard data storage interface as described in Platform-Host Message Protocol document	SW	CP	7/15/94	CP	8/26/96
II	"	2	Provide debug information through NT event services	SW	CP	7/15/94	CP	8/26/96
II	"	3	Interface Tag Reader Computer to Virtual System	SW	CP	7/15/94	CP	8/26/96
II	Housekeeping	1	Properly respond to Scheduler data requests (in Cybermotion message format)	SW	CP	7/15/94	CP	8/26/96
II	Tag Information Processing	1	Provide interface to Savi Interrogator	SW	CP	7/15/94	CP	8/26/96
II	"	2	Query Interrogator (when directed) and update blackboard data structure with status, tag count, platform's X,Y position, list of tags found, and power levels received (in accordance with Host-Platform Message Protocol document)	SW	CP	7/15/94	CP	8/26/96
II	"	3	Provide communications (handshaking, etc.) with Product Assessment Computer as described in Host-Platform Message Protocol	SW	CP	7/15/94	CP	6/1/97
III	Initialization	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Housekeeping	None	N/A	N/A	N/A	N/A	N/A	N/A

** POC for the Tag Reader Computer is Robin Laird - (619) 553-3667*

Category	Major Functions	#	ROBOT SIMULATOR* 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 (robot id, serial port and baud rate) 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	CP	8/26/96
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 K2A manual	SW	CP	7/2/93	CP	8/26/96
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	CP	8/26/96
I	"	3	Provide robot status window display	SW	CP	7/2/93	CP	8/26/96
I	Housekeeping	1	Provide simulated communication using Cybermotion 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 K2A instructions which currently are implemented	SW	CP	7/2/93	CP	8/26/96
I	"	3	Simulate K2A and Scheduler computers while in patrol mode	SW	CP	7/2/93	CP	8/26/96
I	"	4	Simulate most of the instructions described in the K2A control language section	SW	CP	7/2/93	CP	8/26/96
I	User Interface	1	Provide user-controlled blackboard memory and decoding program path scroll	SW	CP	7/2/93	CP	8/26/96
I	"	2	Provide automatic scrolling in the program path decoding mode while the simulator robot is in patrol mode	SW	CP	7/2/93	CP	8/26/96
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	CP	8/26/96
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/Dispatcher enhancements with new Scheduler	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 K2A and scheduler computer 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 K2A program path execution with varying speed	SW	CP	10/94	IP	
II	"	9	Convert to Windows NT	SW	NS	N/A	CP	8/26/96
II	"	10	Support both serial and Ethernet communications protocol	SW	N/A	N/A	NS	
II	"	11	Simulate K2A in AUTOMATIC, HALT, RESUME, MANUAL, TELE-REFLEXIVE mode	SW	NS	N/A	CP	8/26/96
II	"	12	Simulate Scheduler in MANUAL, TELE-REFLEXIVE, PATROL, EMERGENCY_HALT, SELECTIVE_HALT mode	SW	N/A	N/A	CP	8/26/96
II	"	13	Support the Cybermotion BAA template protocol	SW	N/A	N/A	CP	8/26/96
II	User Interface	1	Bullet-proof keyboard input	SW	CP	3/1/94	IP	

II	“	2	Provide additional variable modification capabilities, i.e., program path speed, battery rate of change, and tag reader configuration	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 robot	SW	CP	1/20/95	CP	8/26/96
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 Scheduler RECALL mode.	SW	NS	N/A	NS	10/01/96
III	“	2	Simulate K2A RECALL mode.	SW	NS	N/A	NS	10/01/96
III	“	3	Simulate Scheduler SURVEY mode.	SW	NS	N/A	NS	10/30/96
III	“	4	Simulate SPI SURVEY mode.	SW	NS	N/A	NS	12/30/96
III	“	5	Simulate Scheduler INVENTORY mode.	SW	NS	N/A	NS	1/10/97
III	“	6	Simulate TRC INVENTORY mode.	SW	NS	N/A	NS	1/20/97
III	“	7	Simulate diagnostic failures	SW	NS	N/A	NS	
III	“	8	Degrade gracefully/recover when Link Server CSCI or LAN cable is disconnected	SW	NS	N/A	CP	1/8/97
III	“	9	Simulate the K2A instructions for the Cybermotion BAA enhancements (i.e., Tag G…, Do When …)	SW	N/A	N/A	NS	1/5/97
III	“	10	Simulate SPI Pan/Tilt	SW	N/A	N/A	CP	5/30/97
III	User Interface	1	Provide on-line (windows) help	SW	NS	N/A	NS	
III	“	2	Provide menu selection for setting Scheduler mode and status.	SW	N/A	N/A	CP	8/15/97
III	“	3	Provide menu selection of various diagnostic failures	SW	NS	N/A	NS	
III	“	4	Implement pull-down menus for user interface	SW	NS	N/A	NS	
IV	Command	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	Housekeeping	1	Site-specific emergent work	SW	NS	N/A	NS	
IV	User Interface	1	Site-specific emergent work	SW	NS	N/A	NS	
PIP	Command	1	ICIDS integration	SW	NS	N/A	NS	
PIP	Housekeeping	1	ICIDS integration	SW	NS	N/A	NS	
PIP	User Interface	1	ICIDS integration	SW	NS	N/A	NS	

** POC for the Robot Simulator is Theresa Tran - (619) 553-3671*

Category	**Major Functions**	**#**	**MDARS SUPPORT PROGRAM* Functionality / Requirement**	**HW or SW**	**DOS Status**	**NT CP Date**	**NT Status**	**NT CP Date**
I	N/A	None	N/A	N/A	N/A	N/A	N/A	N/A
II	N/A	None	N/A	N/A	N/A	N/A	N/A	N/A
III	Initialization	None	N/A	N/A	N/A	N/A	N/A	N/A
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 robot 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 robot 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
III	Housekeeping	None	N/A	N/A	N/A	N/A	N/A	N/A
IV	Initialization	1	Site-specific emergent work	SW	N/A	N/A	NS	
IV	Display	1	Site-specific emergent work	SW	N/A	N/A	NS	
IV	User Interface	1	Provide on-line (windows) help capability	SW	N/A	N/A	CP	6/1/97
IV	Configuration	1	Site-specific emergent work	SW	N/A	N/A	NS	
IV	Housekeeping	1	Site-specific emergent work	SW	N/A	N/A	NS	
PIP	Initialization	None	N/A	N/A	N/A	N/A	NS	
PIP	Display	None	N/A	N/A	N/A	N/A	NS	
PIP	User Interface	None	N/A	N/A	N/A	N/A	NS	
PIP	Configuration	None	N/A	N/A	N/A	N/A	NS	
PIP	Housekeeping	None	N/A	N/A	N/A	N/A	NS	

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1998

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
MULTIPLE RESOURCE HOST ARCHITECTURE FOR THE MOBILE DETECTION ASSESSMENT RESPONSE SYSTEM

5. FUNDING NUMBERS
PE: 0603228D
AN: DN309216
WU: CH01

6. AUTHOR(S)
H. R. Everett, R. T. Laird, G. A. Gilbreath, T. A. Heath-Pastore, R. S. Inderieden, K. Grant, D. M. Jaffee

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Space and Naval Warfare Systems Center, San Diego
San Diego, CA 92152–5001

8. PERFORMING ORGANIZATION REPORT NUMBER
TD 3026

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army R&D Center
Physical Security Equipment Division
Fort Belvoir, VA 22060–5420

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The initial effort during Phase I of the Mobile Detection Assessment and Response System (MDARS) Program was aimed at demonstrating the feasibility of a single robotic security platform operating under the high-level control of a remote host with the direct supervision of a human operator. This document describes the concept of a distributed host architecture geared towards a single guard controlling up to eight robots and explores some of the key issues that were considered in the actual development. The objective is to field a supervised robotic security system that basically runs itself until an exceptional condition is encountered that requires human intervention.

A globally shared world model is maintained to provide a real-time collision avoidance capability complementing the Cybermotion virtual path navigation scheme employed on their K2A robotic platform. 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. Each robot is equipped with microwave and passive infrared motion detection sensors providing full 360-degree coverage, and an intelligent security assessment algorithm is employed to maximize the probability of detection while simultaneously filtering out nuisance alarms.

14. SUBJECT TERMS
Mission Area: Ocean Engineering
robotics — artificial intelligence
sensors — security

15. NUMBER OF PAGES
120

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

Approved for public release; distribution is unlimited.