DEPLOYABLE ELECTRONIC COMBAT MISSION REHEARSAL, TRAINING AND PERFORMANCE SUPPORT

Patrick G. Heffernan & David W. Galloway
TRW Avionics & Surveillance Group, Warner Robins Avionics Laboratory
Warner Robins, Georgia 31088

ABSTRACT

This paper presents the approach and results of an internally funded project at TRW to develop a portable, self-contained electronic combat (EC) simulation system. This system, known as the Portable Electronic Combat Simulation (PECS) system, provides the ability to conduct EC mission rehearsal, part-task training, and performance support functions in a deployed state using one stand-alone package. For mission rehearsal and part-task training, this tool provides a real-time simulation of the threat environment, a high-fidelity aircraft flight path generator, an electronic warfare (EW) defensive systems processing and environment interaction, a countermeasures effectiveness model, and an audio and video interface to the user via a graphical user interface. For performance support functions, the system provides an encyclopedia of threat information and a tool for conducting initial and refresher training for specific EW defensive systems.

The PECS system real-time and off-line software is hosted on a single VME chassis and employs multiple 68030 and SPARC CPUs. The real-time simulations software was developed in a building block style allowing the user to rapidly reconfigure his EW defensive systems suite from the models available. The off-line software includes a toolset of editors to build mission files and the performance support functions.

This development effort demonstrated that effective real-time EC mission rehearsal and training and off-line performance support could be employed without large weapon system or aircraft part-task trainers. The PECS system software architecture also illustrated tremendous flexibility in supporting a number of different EW configurations, allowing new and qualified air crew members, from several different airframes, to learn and practice on a single turnkey system. The performance support function shows that air crew members can improve their EW knowledge base without the formal constraints of a CBT system.

ABOUT THE AUTHORS

Pat Heffernan is a senior software engineer at TRW. Mr. Heffernan has nine years of experience in the electronic warfare field. He flew operationally as an Electronic Warfare Officer aboard the MC-130E aircraft. He currently designs and develops simulation and training systems. Mr. Heffernan received a B.S. from the USAF Academy and another B.S. from the University of West Florida. (912.929.7406)

David W. Galloway is a senior systems engineer at TRW. Mr. Galloway has designed and developed real-time digital signal processing and data acquisition/control hardware and software for several radar applications and served as lead engineer during the development of the Integrated Electronic Combat Simulation System (IECSS). He is currently the hardware lead for a team implementing a software support facility for an F-15E avionics suite. Mr. Galloway received a M.S.E.E. and B.E.E. from Auburn University.

DEPLOYABLE ELECTRONIC COMBAT MISSION REHEARSAL, TRAINING AND PERFORMANCE SUPPORT

Patrick G. Heffernan, David W. Galloway
TRW Avionics & Surveillance Group, **Warner Robins Avionics Laboratory**
Warner Robins, Georgia

GENERAL DESCRIPTION

The portable electronic combat simulation (PECS) system was an internal development project at TRW to design, build, and demonstrate a deployable system capable of performing mission rehearsal and training functions for air crew members in the specialized area of electronic combat (EC) using commercial-off-the-shelf (COTS) hardware and software components. Due to the nature of large weapon system trainers (WSTs) and part task trainers (PTTs) and the integration of electronic combat tasks within these WSTs/PTTs, deployed air crew EC mission rehearsal and part-task training using these types of tools is almost non-existent. The objective of this project was to develop a proof of concept system that would provide the capability to provide real-time, high-fidelity electronic combat simulation interfaced with a graphical user interface (GUI) for mission rehearsal and training within the framework of a cost-effective, portable package.

The PECS system provides a full real-time EC simulation capability for multiple aircraft simulation models in either a networked or standalone configuration. That is, the PECS system can be networked to other PECS systems to simulate multiple aircraft if a training scenario involves other air crews or it can be operated as a standalone entity if the user requirements are for a single aircraft platform. It also has the capability to interface with dissimilar systems through the distributed interactive simulation (DIS) protocols.

As the system matured through the development stages, a performance support package allowing users to refresh and upgrade their knowledge on EW equipment, threats, and EW concepts was added. An off-line support system is hosted on the PECS to maintain threat and EW equipment databases and build mission scenarios for the real-time EC simulation. Currently, the PECS system is tailored for specific special operations aircraft

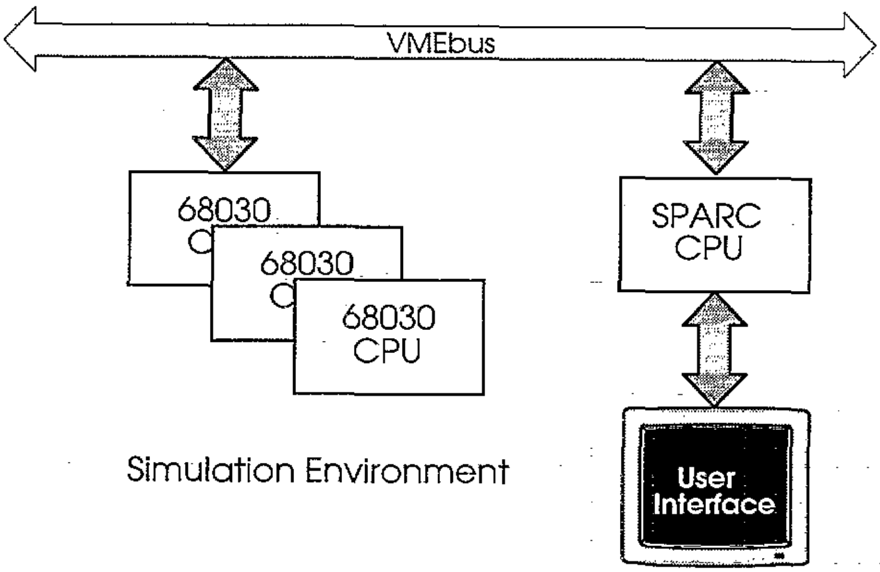

Figure 1. PECS Hardware Block Diagram

(MH-53J, MH-60G, and HC-130P) EW configurations, since many of the real-time simulation models were previously developed for the USAF.

HARDWARE & SOFTWARE ARCHITECTURE

The basic hardware architecture of the PECS system, as shown in *Figure 1*, uses a Force™ Target 32 system as the foundation. This system consists of a 19 inch desktop chassis, a 20 slot VMEbus based motherboard, and power supply with cooling fans. The chassis back panels have been modified to allow cable plug-ins in the rear of the unit. This chassis system was selected due to its lightweight nature, portability, and ability for growth. The VMEbus implementation was selected because it offered a wider and more powerful range of processing devices over a PC based implementation and at a more moderate cost than a minicomputer implementation.

The real-time software simulation executes on three general purpose VMEbus based single board computers. CPU1 is a Force™ CPU-30BE/16 (Motorola 68030 CPU) single board computer and it acts as the VMEbus master. CPU1 also provides the interface to the SCSI disk device containing the required data files for the real-time simulation and the interface to the *Local* Area Network (LAN) via ethernet. CPUs 2 and 3 are Force™ CPU-33B/4 (Motorola 68030 CPUs) single board computers running in a slave mode. These boards provide the processing capability for executing the high-fidelity simulation models required for real-time EC mission rehearsal and part-task training.

The GUI, off-line support system, and performance support system software execute on a Force™ SPARC CPU-2CE (Weitek W8701 SPARC CPU) single board computer utilizing the standard Sun™ UNIX™ operating system environment. The CPU-2CE also interfaces with dual SCSI disk devices to access the off-line executables and data files necessary to operate the system. A 19 inch color video monitor, dual audio amplified speakers, microphone, keyboard, mouse, and any required ethernet lines are connected to the SPARCboard via the back panel cable plug-ins.

The software architecture of the PECS system is based upon a series of modular software components as shown in *Figure 2*. The real-time software simulations executing on CPUs 1, 2, and 3 operate under Integrated Systems pSOS+™ real-time operating system. pSOS+™ allows the simulation's real-time executive software to transparently interface with the system hardware components to accomplish tasks like disk accesses, serial communications, and software interrupts.

The real-time simulation software is coded in a

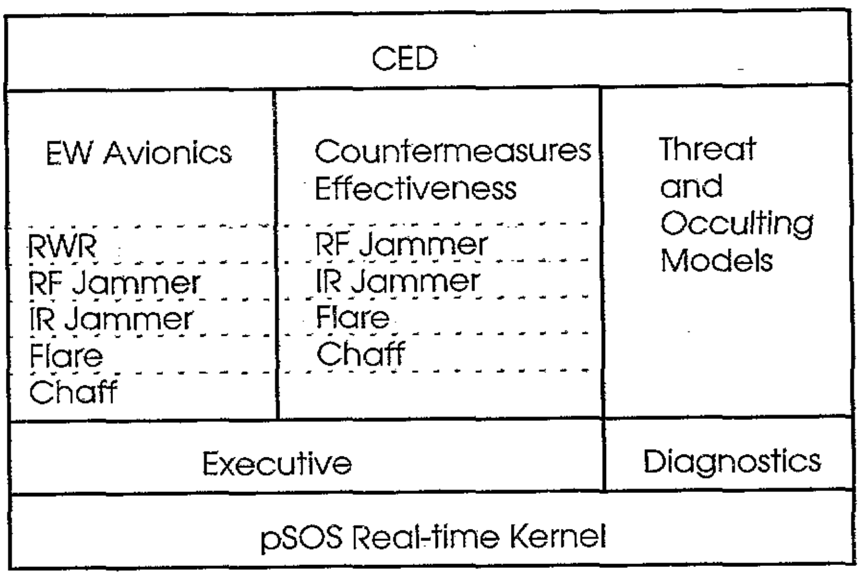

Figure 2. PECS Software Building Blocks.

combination of FORTRAN and C. These high order software languages were chosen due to the number of existing threat and EW avionics software modules. The real-time models are divided into the following major categories:

1) Threat and occulting models
2) EW avionics models,
3) Countermeasures effectiveness models
4) Executive software

The basic data communication path between the threat model simulation and the EW avionics models simulation is a shared memory area mapped to one of the simulation CPUs memory. This shared memory area, called the Common Environment Data (CED) data structure, contains all the information required to define the current electronic combat environment.

The CED describes the battlefield EC environment at any instant in time to the level of fidelity required for a training simulator. This EC environment is composed of data elements that describe threat and emitter entities, weapon entities, aircraft entities, infrared (IR) and electromagnetic (EM) jamming entities, and chaff and flare entities. The CED can be memory mapped so that networked PECS systems can access it across shared memory network using reflective memory techniques.

In a networked simulation, one of the PECS systems is declared as the master of the entire threat simulation environment. It is the master's responsibility to assimilate all scenario and aircraft information from the slave PECS systems to derive the information contained in the CED. That is, the slave PECS systems conduct processing on a number of simulation model functions and then they pass the information to the master PECS system which collects, processes, and redistributes the data necessary for scenario generation.

THREAT & EC SIMULATION HIGHLIGHTS

The PECS system provides a complete, high fidelity threat and EC simulation capability. Since many of the simulation system's models were developed from previous programs (Galloway, et al.; 1993), only a discussion of significant changes and important model highlights of the threat and EC simulation will be presented here. The PECS executes a closed looped EC simulation which integrates the three major components involved in an electronic combat simulation: threat, EW avionics, and countermeasures effectiveness.

The threat simulation includes models which activate and engage targets in a rule-based fashion, perform target assignments, utilize C3 information, drive computer controlled airborne interceptors, fire weapons, conduct 6 DOF weapon flyouts, and perform damage assessments. The threat models use threat and atmospheric data stored in the off-line database to accurately model threat weapon systems and emitters in the gaming environment. During real time execution, the threat simulation supports a maximum of 64 active threats, 128 active emitters, 8 active missiles, 8 active AAA bursts, and 8 active computer controlled airborne interceptors at any given instant in time. In addition, the threat simulation integrates with modules which perform threat encounter recordings, on-line modification of threats and threat status, and threat occulting (terrain masking functions). The threat occulting is performed by a child task on the SPARC board which accesses terrain elevation data stored on the SCSI disks. The occulting function obtains positional information from the threats and targets via the VMEbus, processes it, and passes the occulting results back to the threat simulation on the VMEbus. This process is repeated at a resolution required to support the threat simulation fidelity requirements.

The EW avionics simulation includes models which simulate the operational flight programs (OFPs), emitter identification databases (EIDs), and user displays and interfaces of systems which detect, identify, track, display, and jam or deceive threat systems. In the PECS system, the models drive the graphical user interface buttons, knobs, lights, and switches. The EW avionics OFP and EID simulation samples the threat environment, reads the threat parametric data, processes the threat data, displays the appropriate threat symbology, and creates the associated threat or verbal audio queues. The EW avionics simulation integrates with modules which enable on-line modification of model states (malfunctions) and enables countermeasures effectiveness analysis and simulation.

The countermeasures effectiveness simulation includes models which use the best available effectiveness data to perform a Monte Carlo simulation of jamming and expendable effects on the threat models. The active countermeasures systems models use positional and atmospheric information along with effectivity factor tables associated with jamming techniques and emitter parametrics to assess effectiveness. Expendable models use positional, atmospheric, and expendable timing and quantities to assess effectiveness. Jammer-to-signal calculations are performed and compared with target radar cross section (RCS) and IR signatures. The PECS system can have 8 active chaff clouds per system, 8 active flares per system, 1 active IR countermeasure per system, and 12 active RF countermeasures per system active at any instant in time.

REAL-TIME USER INTERFACE

The real-time graphical user interface was implemented using the Altia™ Design animated visual interface design tool. The software developer can use Altia™ Design to construct, animate, and stimulate static models and provide a seamless interface to the simulation model code. In the PECS system application, Altia™ Design was used to build the EW avionics air crew interfaces, informational data screens, user simulation control screens, and mission situational displays.

The basic structure of the GUI is shown in Figure 3. The user interface provides the actual windowed control environment to the system operator. This portion of the design was implemented using the Altia™ Design development tool. The client application acts as the interface or data conversion code between the user interface and the actual training simulation code. The interface between the client application and the threat and EW simulations is a shared memory area mapped in VME address space to be accessible by all the processors in the system.

The Altia™ Design tool presents the developer with a work space that Altia™ refers to as the universe. This work area is basically a blank area for drawing or producing the views that the end user requires to meet the training strategies. The universe acts as a palette for the client application developer to utilize in presenting the PECS operator with a clipped version of the universe. It is clipped in the sense that only a portion of the universe is presented to the operator in any one window. What portion of the universe that is displayed during run-time is

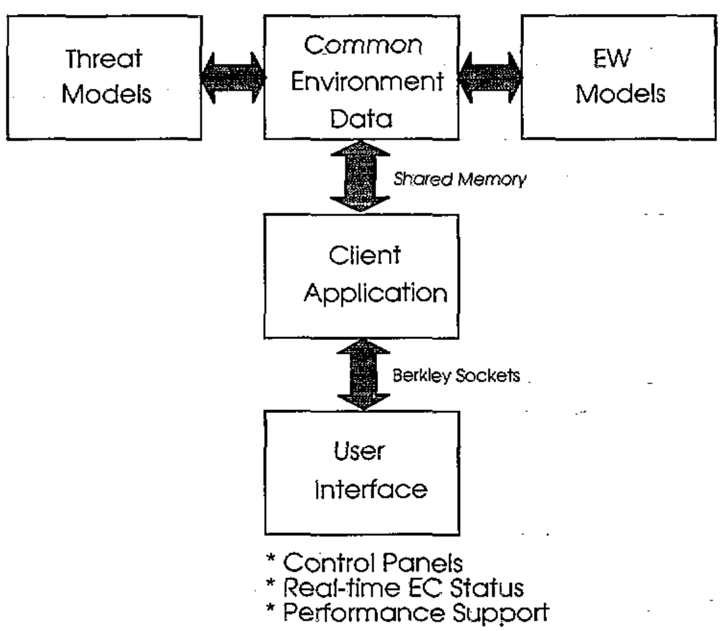

Figure 3. Altia Run Time Configuration.

preprogrammed by the client application developer.

The client application provides more than the view manipulation of the universe. The client application also responds to the run-time interrupts either generated by a stimulus from a displayed view or generated by a time event. An example of a stimulus interrupt would be the operator "pressing" a button on an RWR control panel. The event and the status of the button after the event is transmitted to the client application using operating system sockets. In this example, the client application would translate the button event into the data format defined by the CED documentation.

The combination of the user interface and the client application presents the operator with enough control capability to activate models, control model operation, and display simulation status from a mouse driven interface.

Upon startup, the operator is presented a simulation control panel utilizing one of the views into the Altia universe. The simulation control panel allows the user control of high level simulation functions like the simulation startup and freeze capability. The simulation control panel also presents a menu panel for the user to select the desired control panel views to support his current training session.

The PECS system client application program is coded in C and uses Altia™ Design library routines to manipulate the graphical views. The library routines allow the user to connect the client application to the Altia™ interface, register and process callbacks, create and manipulate graphical views, create and manipulate clone objects, and integrate Motif widgets as an Altia™ interface view.

In the PECS system application, for example, a generic radar warning receiver (RWR) threat display unit was created graphically using the Altia™ Design tool. The RWR screen, buttons, and knobs are drawn and then animated. The stimulus for the animated RWR objects are then defined as being either mouse or keyboard prompted.

Usually, the client application program registers a procedure or routine as a callback for an Altia™ interface event. After this is accomplished, the client application program waits for an event to occur. When the user pushes a button or turns a knob, the client application procedure recognizes the event through the event callback function. Similarly, information from the real-time simulation model is obtained by the client application program and passed to the GUI via the domain socket. The client application procedure is executed after a user specified time interval has elapsed. The time intervals are defined using an Altia™ timer event callback function. The procedure registered as a callback can receive the event data through the callback event timer. This allows data to be displayed to the user in real-time.

Figure 4 is a picture of the PECS system displaying a representative group of user interface screens.

OFF-LINE USER INTERFACE

The off-line user interface consists of a database support system called the off-line support system (OSS) which allows data management for the real-time simulation models. The PECS system uses the ORACLE™ relational database management system and other ORACLE™ products such as SQL*FORMS, SQL*MENU, and ProFortran. Designed using relational database techniques, the database user interface makes data available for update using menu driven interactive screen inputs and standard SQL commands. The database support system is divided into four major categories:

1) Library Maintenance
2) Mission Preparation
3) Configuration Management
4) Miscellaneous Function.

Library Maintenance involves editing threat parametrics, EW avionics data, and threat related information which is used to support a number of EC simulation functions.

The threat parametrics portion of library maintenance includes data used by the real-time threat and weapon models to perform an accurate simulation of the threats as they interact with the EC environment. The threat

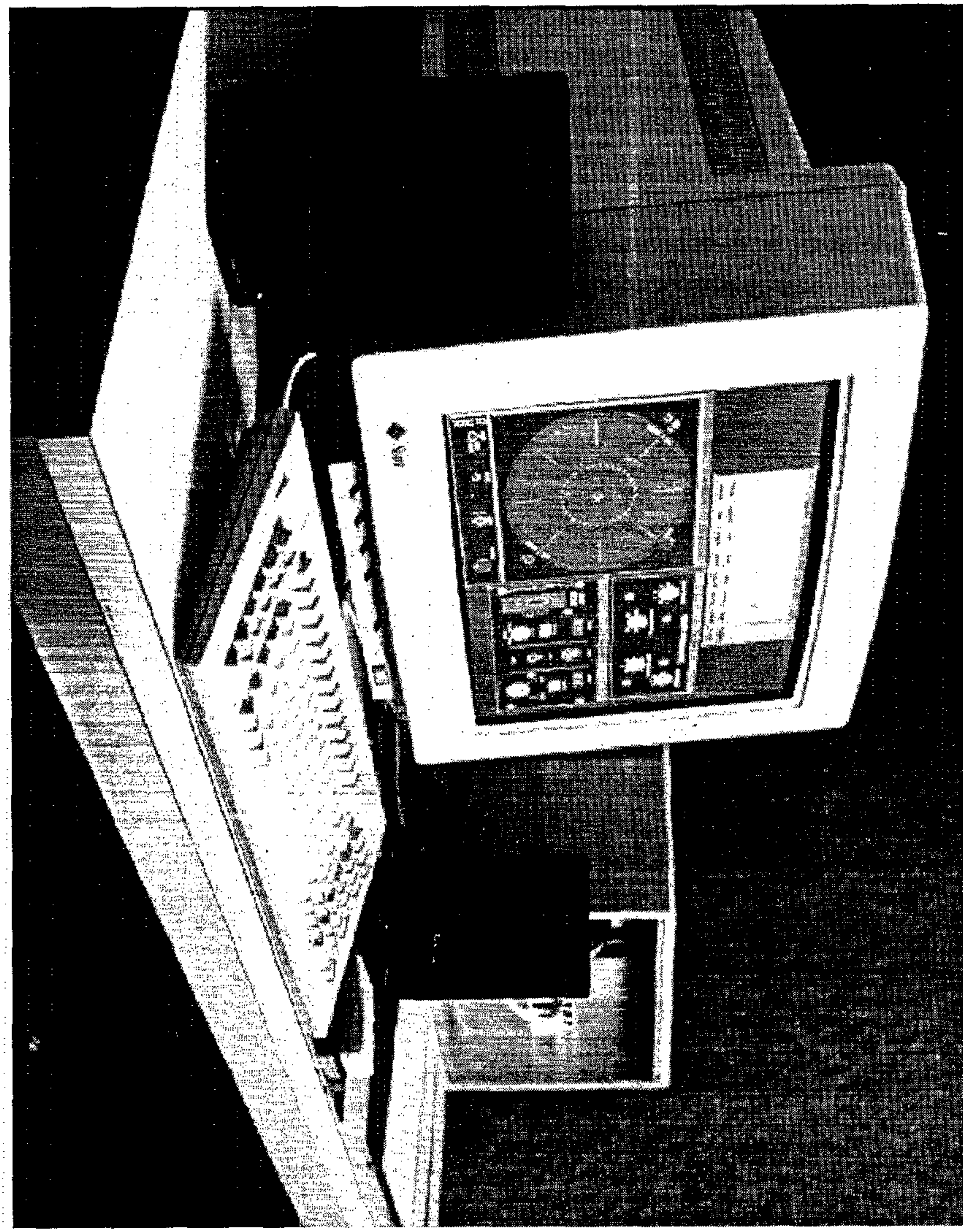

Figure 4. PECS System

parametrics include emitter parametrics and tactics data by mode and emitter, weapon data, and general threat information.

The aircraft EW configuration portion allows specific aircraft system setup data to be defined. This includes system warm-up and cool down times, the positioning of countermeasures dispense hardware onboard the aircraft, and equipment field of view limitations.

The EW system characteristics and EID libraries contain data used by the various EW system models to perform accurate simulations of the EW systems against threats in the environment.

The EOB/C3 library defines engagement modifiers, mode time modifiers, fire time delays, and communication delay times based on a particular conflict level. The conflict level defines the skill level or alert status of enemy forces.

The sites and platforms library maintains a list of threats associated with a site or platform entity. The site is a static entity which allows the user to construct real threat sites using JARM entities. The platform is a dynamic entity which allows the user to associate JARM entities to an object capable of being moved around in the real-time gaming scenario. The platform can be defined as either a computer controlled airborne interceptor (AI) or as a ship/AI platform. The computer controlled AI automatically moves and engages targets in the gaming scenario based upon positional information. The ship/AI platform allows the user to move the platform around the gaming scenario using the mouse or trackball. The sites and platforms built using the OSS can be positioned in a mission scenario causing all associated JARMs to be defined in the gaming environment.

The Mission Preparation function involves the building of mission scenarios, or threat laydowns for the real-time EC simulation. The mission preparation function allows the user to construct a mission load which contains up to 25 mission files consisting of up to 400 threats in each mission file. This mission load is downloaded to the real-time SCSI disk drive and is read during scenario initialization. The mission load also includes all threat and weapon parametrics, EW system characteristics and emitter identification data, and threat positional information required to support the real-time EC simulation.

The mission laydown is the EC gaming environment defined by the user to accomplish a specific part task training or mission rehearsal objective. The mission laydown is assembled by positioning threats using the latitude/longitude coordinate system. Each threat is assigned other pertinent information such as altitude, speed, heading, C3 data, and site/platform information. The mission laydown also defines which threats are networked together or are autonomous in the gaming environment.

The PECS system can be interfaced via an ethernet LAN with other mission planning systems so that mission laydowns can be transferred from the mission planning system to the OSS using standard file transfer protocols.

PERFORMANCE SUPPORT FUNCTIONS

The graphical user interface design for the performance support function (PSF) of the PECS system is an icon based, menu driven approach that allows easy access to the different task areas. For each task area, the PSF allows related information to be accessed. For example, with RWR operation, the user can obtain technical order information, view RWR operation in an automated fashion, or obtain a training lesson on RWR operation. Full audio narration is provided when requested for appropriate task areas and important warnings, cautions, and notes are amplified. Thus, the user can access related information in a myriad of different ways.

The PSF provides the capability of providing information concerning specific task areas related to EW avionics operation, checklist procedures, and recommended tactical operation. Currently, the system is limited to the EW systems currently used in the aircraft system previously mentioned. The function also provides a threat encyclopedia and a computer based training function.

The task area element is designed to provide refresher training and advanced concepts regarding EC to air crew members. The task area graphically details the operation and use of the selected EW avionics.

The threat encyclopedia function provides a graphical representation of the threat system with associated text and audio. The user can choose a selection from radars, missiles, guns, and jammers.

The CBT function provides a formal, step-by-step walk through of system operations, checklist procedures, and malfunction analysis.

SUMMARY

The PECS system provides a substantial EC training and mission rehearsal capability stuffed into a portable package. The system provides the flexibility and capability to train a wide variety of air crew member skill levels. And due to the use of COTS products, it can be more easily integrated with other training products (like portable weapon system trainers and mission planning systems) and its maintainability is greatly improved.

Although a great deal of development has occurred on the PECS system, a number of items can be added to improve training capability. For example, the use of digitized pictures with superimposed, animated knobs and buttons can be used for added realism. Digitized video can be utilized to better show EW system operation in the performance support functions to improve and maintain learning curves. 3D terrain images and cockpit displays can be added to expand training capabilities and realism. The PECS system COTS architecture allows for expandability, fidelity upgradability, and modularity.

The PECS system shows that effective EC part task training and mission rehearsal can be performed within a portable package using COTS products. Air crew member training and mission readiness is enhanced due to the fact that a system like this can be deployed quickly and in environments where larger systems cannot be employed. Maintainability is vastly improved with the use of COTS software and hardware products. Thus, overall system life cycle costs are substantially reduced while specific task training and mission readiness is improved.

REFERENCE

Galloway, D.W. , Heffernan, P.G., Nuss, E.A., Summers, C.M., Proceedings, 15th Interservice/Industry Training And Education Conference, Nov. 1993.