Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

A SIMULATION FOR COMBAT SYSTEMS DEVELOPMENT AND ACCEPTANCE TESTING

Thad J. Janowiak

U. S. Navy
Integrated Combat Systems Test Facility
San Diego, California 92152-6900

ABSTRACT

This paper will address a single U. S. Navy simulation program used by both program developers and acceptance test agents at the Combat System level, the highest hierarchial level on a ship class basis. The Combat Direction System (CDS) subsystem performs Command and Control of the Combat System aboard ship. To perform this function, sensor information from radars, sonars and communication data links is required. A coordinated picture, based on these sensor inputs, is presented for proper decision making on what ordnance should be placed on each threat. CDS also interfaces with Weapon Systems to direct engagements based on threat priority logic. Sensor information enters CDS continuously and depending on the environment, amount of vehicular traffic and weather clutter, high steady state and instantaneous data rates can be realized. For this paper, the combination of CDS, Sensor, Communication and Weapon Systems equate to a Combat System.

CDSs employed by the U. S. Navy utilize military digital computers that have the capability to process an enormous amount of Input/Output data along with high performance, multiple Central Processing Units. Display equipments typically use both graphic and data presentations and Man Machine Interface is extremely important to ensure the proper information is available and understandable, and that decisions are executable without undo complexities. This hardware, coupled with the Command and Control requirement, demands that a real time software architecture be employed to meet the demanding processing tasks.

Development of CDS programs necessitates actual or reasonable facsimile (simulation) of the various sensor, data links and weapon system interfaces. This is required for proper stimulus during the latter stages of development and subsequent testing. This paper describes the U. S. Navy's Integrated Combat Systems Test Facility involvement in the development of a real time simulation program used by program developers and testers.

1. INTRODUCTION

Prior to Combat System computer program development, each subsystem requiring an external interface with another subsystem computer, develops an Interface Design Specification (IDS) which describes the communication protocol and messages sent between them. During program design, the IDS is the governing specification for Input/Output communications. The individual development sites do not possess the necessary equipment to provide all the actual/real digital interfaces as specified by the IDSs and simulation must be used. Stimulus at the computer Input/Output frame of reference, is used to vector the program through the various processing tasks it was designed to perform.

This stimulus, referred to as "Interface Simulations (INTSIMs)", are simulations of subsystems at the IDS level. INTSIMs take on the character of the activity or agency designing them. They are functionally designed to respond with a given set of messages under the event constraints that are specified within the IDS. The fidelity, or degree to which they simulate the host subsystem, depends on the initial requirement. From a timing aspect alone, computer to computer interfaces can perform quite differently than an IDS implies. For instance, if an IDS requires that the receiving system transmit a message within 85 milliseconds from a specific event, a simulation designer can choose from a multitude of response times as long as they stay within the initial requirement. The actual system measured with a logic analyzer may respond between 3 and 8 milliseconds, depending on processing load. Since the INTSIM focuses on the IDS, it provides the developer with limited level of confidence that the system performs within original specifications.

The U. S. Navy has established a land based test site where subsystem computer programs are integrated into the Combat System. U. S. Navy's Integrated Combat Systems Test Facility (ICSTF) performs a comprehensive set of tests on the newly developed subsystem programs and uses a coordinated environment to stimulate at the Combat System level. ICSTF replaces the INTSIMs used at the developers site with actual subsystem programs and uses "Satellite Simulations (SATSIMs)" to drive the individual subsystems. The SATSIMs host functional simulations of missing components, such as radar signal processing equipment and missile launcher interfacing elements, all needed by the subsystem, for stimulus, as it progresses through the various stages of development and acceptance testing. SATSIMs are normally developed by the subsystem program developer and are a derivative of their Wrap Around Simulation Program (WASP). The name was derived from the very nature that all external interfaces were simulated or "wrapped around" the operational programs. The WASPs perform the function of interface stimulus to operational programs where Input/Output exciters are needed to kick off certain program processing. They in fact host INTSIMs at the subsystem level. ICSTF provides technical support to the developer during conversion to a SATSIM.

Faced with similar simulation requirements from various field activities, the U. S. Navy embarked on an approach to develop a single simulation program that could meet a multitude of requirements. The Standard Simulation System, later changed to the Combat System Simulation (CSS) is the tool to be used by both Combat System program development and test activities. Figure 1 illustrates the concept of how INTSIMs and SATSIMs perform under control of the CSS. SATSIMs can be switched in at ICSTF during test events, while INTSIMs are used by CDS development agencies.

The CSS is the heart of the simulation system by providing a coordinated environment of vehicular targets and ownship motion. CSS also provides control linkages between itself and the SATSIMs to allow for specific operator or scenario control of the different SATSIMs. Additionally, events such as missile launch are feed back to the CSS for vehicular track generation. The remaining sections of this paper will concentrate on the CSS.

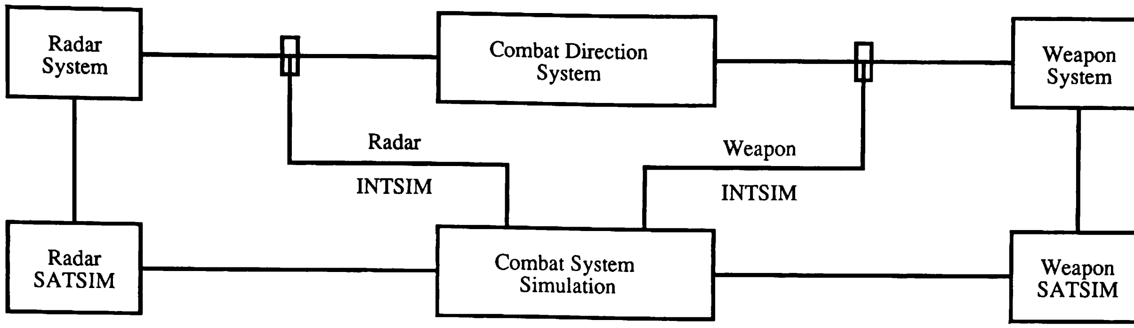

Figure 1. Combat System Simulation Functional Block Diagram

2. SYSTEM REQUIREMENTS

The CSS architecture to support both developers and the ICSTF testing environment is based on the concept of:

* Coordinated Environment
* Operator Control
* Scenario Control
* Core Library

2.1 Coordinated Environment

The CSS is the master simulation element in the simulation system. The CSS maintains data base integrity throughout all INTSIMs and SATSIMs by computing all track and ownship motion and projecting it into the "future" one second ahead in time. CSS data is transferred to the individual simulation components along with clock synchronization time, so unpacking the "future" track and ownship data can be accomplished accurately. This is necessary since each SATSIM and/or INTSIM may have various timing restrictions placed on them by their operational programs. This technique ensures ample processing time is available, for each simulation component, to present the same coordinated picture.

2.2 Operator Control

Complete operator control of the CSS is necessary to support the previous stated requirements of program development and test operations. U. S. Navy standard display consoles employing graphic and alpha numeric capabilities are currently part of the CSS equipment suite. These consoles allow operators to interact with the system for environment control and to operate the INTSIMs. Vehicular tracks are entered at the console and in turn, are displayed from specified reference points, with appropriate symbology and attributes. When INTSIM control is required, the operator can select appropriate console arrays and take control of the interface information. With this real time operator control, the CSS can effectively support test operations. Operator interaction with the system is necessary during system initialization and is very useful in support of CSS and operational program development and fault localization activities.

2.3 Scenario Control

To ensure the environment to the Combat System can be placed under strict control by the users, a repeatable test environment capability in the form of a scenario control feature is available. Two distinct methods of creating a scenario are implemented. One records on the CSS magnetic tape unit all scripted operator actions performed over a selected period of time that can later be played back. This method requires the CSS mainframe computer along with display and peripheral equipment to record the scenarios. This amount of equipment is often difficult to schedule and obtain, and a second method was developed that provides the testers with the ability to create scripted operator actions on a different host system, remote from the test laboratory.

Due to the real time architecture of the CSS, operator actions cause track and ownship movements by entering new vehicular velocities, desired altitude and course. The motion generator computes the required movement, each one second update cycle, until the desired vehicle positions are achieved. Therefore, any operator action performed during scenario playback would cause additional motion changes from what was intended. This design does not implement a feedback loop between the CSS motion generator and scenario control, and all operator actions performed on a specific vehicular track during playback will result in the loss of scenario control for that track. During actual Combat System testing, the majority of all testing is performed using the CSS scenario feature, without operator intervention.

2.4 Core Library

To achieve the goal of a single simulation system supporting a multitude of users, a CSS library was established by the U. S. Navy in San Diego under ICSTF's cognizance. This library holds all software and associated documentation for the CSS and Configuration Management is strictly adhered to ensure library validity. As digital interfaces between subsystem elements change to improve performance or correct anomalies, new INTSIMs are added to replace or compliment existing library held products.

3. SYSTEM DESCRIPTION

The CSS is comprised of different program modules that are linked together to represent a specific U. S. Navy ship class. The modules are categorized into four main groups:

* Operating System
* Support Modules
* Interface Simulations
* SATSIM Control Links

3.1 Operating System

The operating system consists of modules that provide basic program control and scheduling responsibilities for the real time architecture requirement. Functions such as program loading, executive scheduling, common user routines, inter-module message communications and use of peripheral devices are performed by the operating system. The U. S. Navy standard operating system, 43 Runtime Support Software (43RSS), is utilized by the CSS which consist of:

3.1.1 Standard AN/UYK-43 Executive (SDEX/43)

SDEX/43 provides software access to the hardware features of the AN/UYK-43 computer. It allocates computer resources among programs within the system, performs initial software decoding of all interrupts within the computer, initiates and controls Input/Output operations, identifies software and hardware errors and provides access to the Fault Tolerant and System Reconfiguration Module (FTRSM).

3.1.2 Dynamic Modular Replacement (DMR)

DMR determines internal hardware resources, loads/initializes the 43RSS system and performs error recovery processing in conjunction with FTRSM. DMR also provides user service functions for loading/deleting segments, modules, or configurations, and for dedicating memory for use by system programs.

3.1.3 Debug (DB)

DB provides efficient debug tools to maintain both application and executive programs in a real time environment. Breakpoint definition, SNAP parameters, and event triggered data collection are some of the DB features.

3.1.4 Resident Debug (RDB)

RDB assists in the real-time on-line debugging of program functions by providing inspection, change, and dump of designated memory locations, registers and tables.

3.1.5 Common Peripheral (CP)

CP controls all Input/Output activity to the standard peripheral devices consisting of magnetic tape units, disks and printers. Processing includes checking user software requests and operator commands along with received peripheral interrupts.

3.1.6 Common Systems (CS)

CS provides system data management, mathematical and conversion routines, system service routines, recovery routines, diagnostic operator notification and operator command interrogations.

3.1.7 Fault Tolerant and System Reconfiguration Module (FTRSM)

FTRSM is a ROM resident program located in each CPU. FTRSM contains resource status, module fault history queue tables and RAM scratch area for diagnostics. FTRSM Interface Module (FIM) is the RAM resident program portion of the FTRSM program located in main memory. FIM performs system hardware fault detection, fault processing, fault isolation to the functional module. FIM also assists the Executive in reconfiguration and reinitialization.

3.1.8 Fault Acceptance Module (FAM)

FAM coordinates the built in AN/UYK-43 fault tolerant capabilities and supports on-line error detection, reconfiguration and repair of system hardware.

3.1.9 File Handler (FH)

FH provides users with an efficient means of utilizing mass storage without being physically dependent upon the physical structure of the target mass storage device.

3.2 Support Modules

The modules associated in this group interface with the 43RSS operating system and provide additional functions needed by subsequent application modules. The support modules consist of:

3.2.1 Display Control (CD)

CD provides the capability for the CSS operator to interface with the system. Up to five OJ-194 U. S. Navy standard consoles, configured with Console Internally Generated and Refreshed Symbols and Digital Display Indicators can be controlled by CD. Operator assistance is provided by limited help menus keyed to console push button entries.

3.2.2 Environment Control (CE)

CE provides the capability to simulate ship motion components of roll, pitch, heave, and ship heading changes along with environment parameters of wind, ocean currents and sensor noise. CE can be controlled by the operator or via scripted scenarios.

3.2.3 Printer Control (CN)

CN interfaces with and controls the print requests and queueing for the medium speed printer TT 624. Application modules requiring hard copy print outs will inform CN for all requests.

3.2.4 Scenario Control (CA)

CA provides the means to generate a scripted scenario on magnetic tape via the CP module. Editing, playback, and fast forwarding are resident features of CA. A scenario consists of recording events (operator actions) over a period of time and affixing event/test number for identification purposes.

3.2.5 Track Data Control (CT)

CT provides the means to enter, delete, and modify vehicular track positions and rates via the OJ-194 display console and scenario control.

3.2.6 Track Management (CK)

Once CT or scenario control has entered the initial vehicular track parameters, CK provides dynamic updating of the track positions, transforming latitude and longitude and height parameters to global X, Y, and Z coordinates. CK also transforms global to local cartesian coordinates and computes planar range, bearing and elevation values referenced from ownship.

3.2.7 USQ-69 Interface (CQ)

CQ provides an additional operator interface with the CSS via an USQ-69 data terminal. This terminal provides alpha numeric data and is used to display system status information such as channel assignments or error messages. CQ can also be used to enter and control ownship and vehicular track parameters.

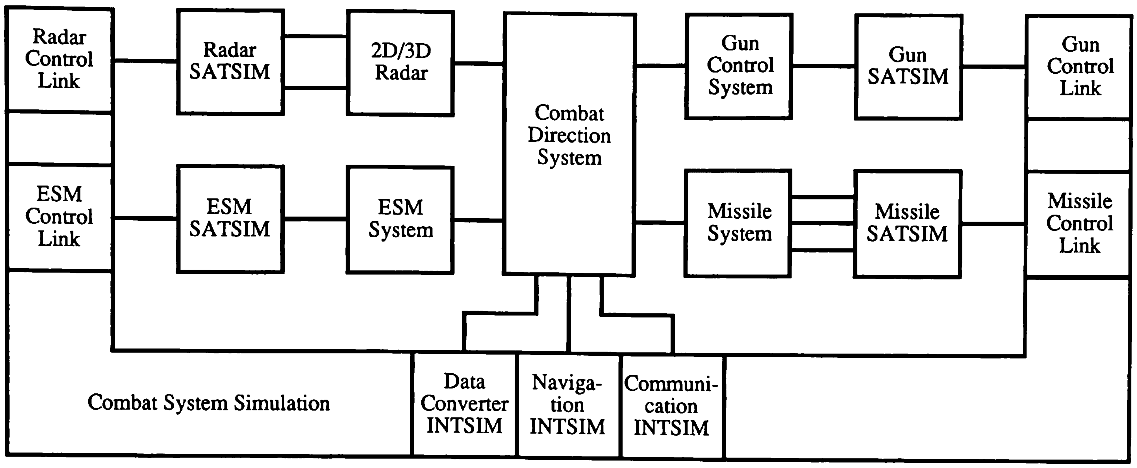

Figure 2. Combat System Integration Test Bed

3.2.8 Video Simulation (CV)

CV interfaces with up to four SM 441 Video Signal Simulator's or Radar Video Simulators to provide sensor video to the Combat System under test. Radar and Identification Friend or Foe (IFF) video with selections for bias, beam width and noise are capable of being controlled by CV.

3.3 Interface Simulations

Previous discussions centered around the Operating System (OS) and Support Modules (SM). These functions are the nucleus of the CSS, required by all users and form the basis from which application modules can be developed. The CSS has limited utility without applying external interfaces to the Combat System. Interface Simulations (INTSIMs), the provider of external interface communications, were described in the introduction section and require little or no elaboration here. However, INTSIMs are designed to perform in accordance with the IDS, provide operator help menu displays and scenario scripting capabilities, and perform under a systems approach addressing the proper Input/Output and executive entrance scheme established by the OS and SM.

3.4 SATSIM Control Links

Satellite Simulation Control Links, as are INTSIMs, are application software that require both OS and SM functions. The SATSIM concept was previously discussed under the introduction section and this presentation will concentrate on how control of the SATSIM is performed.

Since both target and ownship motion generators resides in each SATSIM, the CSS must synchronize all independent SATSIM generators for proper target and ship movement presentation to the Combat System. It must be emphasized that correlation of sensor data is critical for the Combat System to determine potential threats and subsequent weapon selection. The CSS must be extremely accurate on target presentations for all SATSIMs. Each SATSIM uses their own motion algorithms which can differ between systems. With independent motion generators, sensor correlation would be non-existent. To compensate for this, the CSS was designed to provide target and ownship position and motion rate data projected one second into the future, each and every second.

The SATSIM Control Links provide this information as defined in an IDS developed between CSS and SATSIM personnel. Since the information is one second ahead of present time, the SATSIM is provided with enough reserve processing time to extrapolate the data for further presentation to operational program as defined in equipment specifications or other IDSs. Using this method of synchronization ensures the degree of accuracy required by system testers.

Besides synchronization techniques, the Control Link modules are tailored to the specific SATSIM for the uniqueness each operational program has. Bias, noise and precision errors, selectable by the operator, are additional features employed by the Control Link to present less than perfect conditions to the Combat System under test.

4. COMBAT SYSTEM TEST BED

Figure 2 represents a typical Combat System test bed that has been designed with both INTSIMs and SATSIMs. The outer horseshoe is the CSS, with SATSIMs as the next tier, followed by the operational programs. This would represent a complete test bed except for unique data extraction and reduction for test analysis. The INTSIM/SATSIM combination is necessary due to cost and technical considerations. As stated previously, both INTSIMs and SATSIMs receive a timely coordinated environment picture and process the unique protocol and data presentations required by the individual operational programs.

A real advantage of the CSS concept is how INTSIMs are used by multiple U. S. Navy activities. Since Combat System Integration Testing (CSIT) is the final qualification test prior to shipboard delivery of the operational programs, CDS software developers delivering programs to undergo CSIT, have previously developed and tested their programs using the CSS hosted INTSIMs. Due to this sequence of events, the INTSIMs acquire an additional level of maturity and stability since they are validated during the operational programs development and acceptance testing phase. Once the CSIT process is reached, the INTSIMs have a minimum of one year prior testing with resultant corrections installed.

5. SUMMARY

The CSS concept is not new, and the U. S. Navy has been using this "Wrap Around" architecture to develop and test programs for many years. The standard library of simulations under the CSS umbrella, with core life cycle maintenance responsibilities and strict adherence to Configuration Management has been in place for the past three years and is working.