Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

FUSION PROCESSOR SIMULATION (FPSIM)

Mark D. Barnell

Computer Science Corporation
PO Box 4308
Rome, NY 13442-4308, U.S.A.

Douglas Wynne

Air Force Research Laboratory
Information Directorate
Rome, NY 13441, U.S.A.

ABSTRACT

The Fusion Processor Simulation (FPSim) is being developed by Rome Laboratory to support the Discrimination Interceptor Technology (DITP) Program of the Ballistic Missile Defense Organization (BMDO). The purpose of the FPSim is to serve as a test bed and evaluation tool for establishing the feasibility of achieving threat engagement timelines for theater missile defense. The FPSim supports the integration, evaluation, and demonstration of different strategies, system concepts, and Acquisition Tracking & Pointing (ATP) subsystems and components. The environment comprises a simulation capability within which users can integrate and test their application software models, algorithms and databases. The FPSim must evolve as algorithm developments mature to support independent evaluation of contractor designs and the integration of a number of fusion processor subsystem technologies. To accomplish this, the simulation contains validated modules, databases, and simulations. It possesses standardized engagement scenarios, architectures and subsystem interfaces, and provides a hardware and software framework, which is flexible to support growth, reconfiguration, and simulation component modification and insertion.

Key user interaction features include:

- Visualization of platform status through displays of the surveillance scene as seen by imaging sensors
- User-selectable data analysis and graphics display during the simulation execution as well as during post-simulation analysis.
- Automated, graphical tools to permit the user to reconfigure the FPSim, i.e., “Plug and Play” various model/software modules.

The FPSim is capable of hosting and executing user’s software algorithms of image processing, signal processing, subsystems, and functions for evaluation purposes.

1 OVERVIEW OF THE FPSIM

The Fusion Processor Simulation (FPSim) is being developed by Air Force Research Laboratory / Information Directorate (AFRL/IF), Rome, NY, to support the Discrimination Interceptor Technology (DITP) Program at BMDO. The DITP effort is developing advanced technologies for theater and national missile defense.

The DITP is developing sensor (IR and LADAR), miniaturized fusion processor, sensor fusion algorithms, missile seeker and Kinetic Kill technology for detection, tracking, discrimination and kill during missile post boost and midcourse phases. The sensor, algorithms and processor would be integrated on a single Kinetic Kill Vehicle (KKV) platform. AFRL/IF at Rome are responsible for the DITP Algorithm suite, DITP fusion processor hardware, integration of the algorithm suite with the hardware (Fusion Processor), and support the integration of the Fusion Processor with the sensors and platform developer.

This paper provides the objectives, capabilities, operational concept and future plans of the FPSim.

2 OBJECTIVES

The purpose of the FPSim is to serve as a test bed and evaluation tool for establishing the feasibility of achieving threat engagement timelines. The FPSim supports the development and testing of Acquisition, Tracking, Pointing-Fire Control (ATP-FC) subsystems and their decision function algorithms. Figure 1 illustrates the ATP-FC for a DEW Platform.

The FPSim provides a high-fidelity ATP-FC simulation of the principal system concept: DITP operating against a threat consisting of multiple ballistic missiles during boost, post-boost, and mid-course phases of flight. (shown in Figure 2.) The primary objectives of the FPSim is to provide necessary model upgrades to perform discrimination, perform demonstration and technology runs, perform data analysis, provide analysis reports, and to provide an end-to-end determination of conceptual systems as well as evaluation of fusion algorithms and processor interfaces.

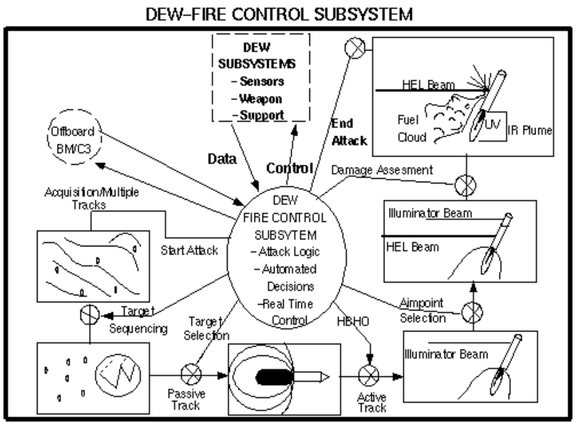

Figure 1: DEW-Fire Control

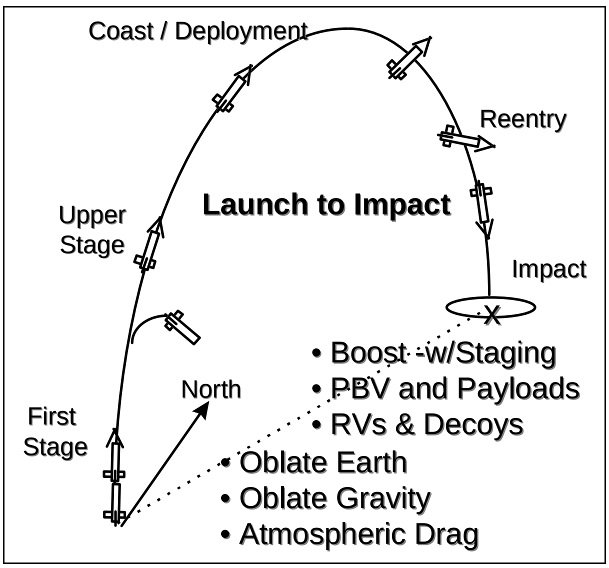

Figure 2: Target Model

3 CAPABILITIES

The FPSim supports the integration, evaluation, and demonstration of different management strategies, object discrimination, track fusion, system concepts, and ATP-FC subsystems and components. It accomplishes this by providing an environment that represents realistic physical phenomenology, ATP system performance and architectures, and Fire-Control management system responses to selected engagement scenarios. The environment comprises a simulation capability within which users can integrate and test their applications software models, algorithms and databases. The FPSim provides the timing controls, data flow, tools and devices (visual and graphics display, etc.) required to generate, collect, and analyze simulation data. This data will help ATP-FC and Object Discrimination technology developers and BMDO personnel to:

- Assess system and subsystem concepts for feasibility and effectiveness by analyzing system performance under realistic conditions.

- Evaluate the feasibility of alternative approaches to various aspects of Fire Control Management design including requirements for onboard databases such as kill probabilities.

- Perform timeline budget studies of Fire Control and computing systems, and establish flowdown of response time constraints.

The FPSim must evolve as the various platforms and ATP-FC algorithm developments mature to support the independent evaluation of contractor designs and the integration of a number of ATP-FC subsystem technologies. To accomplish this, the FPSim contains validated models, databases, and simulations. It includes standardized engagement scenarios, architectures and subsystem interfaces, and provides a hardware and software framework which is flexible enough to support growth, reconfiguration, and simulation component modification and insertion. The FPSim is capable of hosting and executing user's software algorithms of various platforms, subsystems, and functions for evaluation purposes.

The FPSim provides users with tools/models to support seven main features:

- A Standard Environment Model for representing scenarios, threats and backgrounds using community approved models and the tools to integrate these models into the simulation.

- A platform Simulation Configuration Tool for building complete platform simulations, and providing the ability for users to select from a library or import their own ATP-FC algorithm, sensor, illuminator, and structural models.

- A set of baseline models for simulating each of the primary platform types and missions that can be reconfigured by the Simulation Configuration Tool (shown in Figure 3)

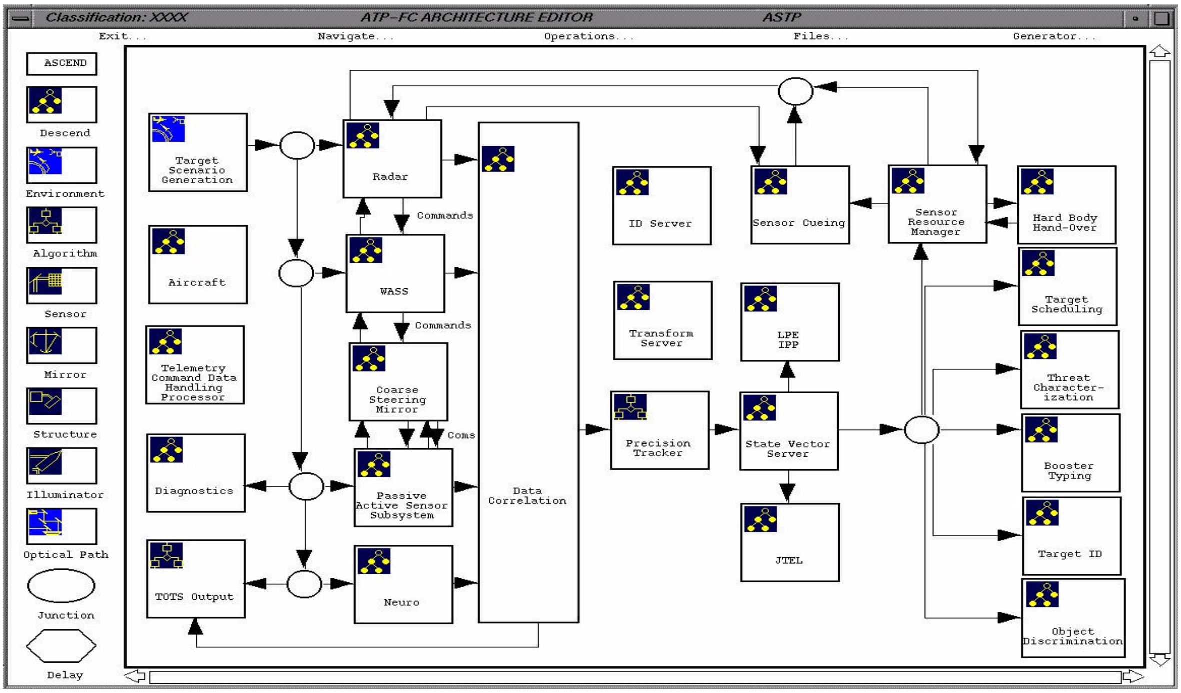

Figure 3: Simulation Configuration Tool

- Analysis and Display Tools that support the evaluation of simulation data and the communication of test results to technical and program management personnel.

- Key user interaction features include visualization of platform status through displays of the battle scene as by the imaging sensors as shown in Figure 4.

- User-selectable data analysis and graphics display during the simulation execution as well as during post-simulation analysis.

- Automated, graphical tools to permit the user to reconfigure the simulation, i.e., "*Plug and Play*" various model /software modules (Figure 3). These modules, either imported to the simulation by the user or available in the FPSim program support library, represent different configurations of various systems, algorithms, components or sensors.

These capabilities support testing of alternative platform designs (hardware or ATP-FC algorithms) through the importation of the appropriate models into a common integrated platform simulation. The simulation is executed, the models' performance analyzed, and the results compared against other models evaluated in the same setting.

The FPSim is a simulation environment for developing surveillance, Directed Energy Weapons (DEW) and Kinetic Kill Vehicle simulations. The FPSim, when completed, will contain three baseline simulations (Spaced Based Laser [SBL], Atmospheric Surveillance Technology Program and DITP). These simulations are called the FPSim baselines. The baseline models include standard launch scenarios, operating in an environment that includes all major observable signatures IR and RF (plume, hardbody, terrain, cloud, earthlimb, aurora, celestial and nuclear bursts). The environment databases and IR sensor phenomenology are supplied by the Synthetic Scene

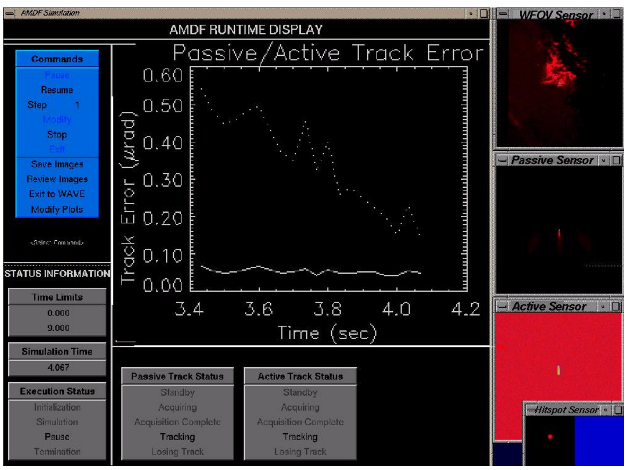

Figure 4: Run-Time Display

Generation Model (SSGM). Platform models include ATP-FC algorithm software emulation, and sensor and dynamics simulation. The baseline provides an initial set of simulation components for the development and testing of new components and architectures.

4 OPERATIONAL CONCEPT

The FPSim may be used to support a variety of analysis objectives. This section summarizes the ways in which a user can utilize the system.

Analyze the Baseline Simulation

A user can execute the simulation directly to analyze the performance of a baseline simulation. The user can vary simulation parameters, modify scenarios, and change phenomenology to study the response of the subsystems aboard the platform.

Modify the Baseline Simulation

The user can modify a baseline architecture to study the performance of alternative ATP-FC algorithms or platform models, or to assess alternative architectures. The user could replace an algorithm with an alternative algorithm in order to assess relative performance. This can be done with components as well, for example sensor models.

Design a Platform Architecture

The user can start from scratch (or use pieces of a baseline architecture) to develop a new platform architecture. For example, the FPSim could to used to model the ATP-FC system for the High Altitude Balloon Experiment (HABE).

Import and Utilize User-Developed Platform Models

Inherent in many of the above operations is the use of user-developed software. The FPSim supports importation of C, C++, FORTRAN, and Ada software that models platform components (algorithms, sensors, mirrors, illuminators, structures, optical path, and environment). There are specific interface requirements for the importation of user software. In order to import software, the user's software must be modified to meet these requirements and inserted into a C++ shell. In addition, a file must be prepared listing the inputs and outputs of the software. The software model can then be used in building a simulation (Plug and Play).

Incorporation of Measure Data

The user can replace sensor models with real world measured data in order to assess system and module performance on measured data.

5 SOFTWARE COMPONENTS IN FPSIM

The major software components of the FPSim release 1.1.2 are the User Interface (Building a Simulation), Kinematics, Support, Executive, Scenario, Application, Engagement, Interface components, and the data analysis system (which utilizes the commercial package, PV-Wave). The Reconfigurable modules include Algorithms, Dynamic Models and Sensor Models. Most of these components are shown in Figure 5. The FPSim has been designed to work with software tools using the scenes that the SSGM stores or builds in run-time library. The software components operate on a Silicon Graphics Workstation (Indy, Indigo2, O2 or Octane Models).

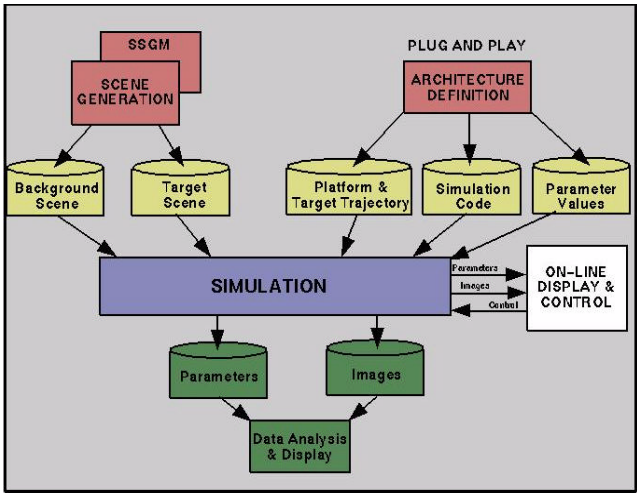

Figure 5: Components of FPSIM

The FPSim is compiled with IRIX OS 5.3,6.x; Base Compiler Development Environment; ANSI C, SGI C++; GNU C++, 2.7.2, and PV-Wave 6.x .

The Simulation Configuration Tool can be used to build a Simulation that supports the user in setting up a simulation, configuring a platform architecture, executing the simulation, and managing the simulation assets. The process is described as "Plug and Play". This tool is written in the C language and utilizes the X Window System.

Figure 6 provides a functional block diagram of a DITP tracking system, showing elements included in the FPSim.

The major components of the Plug and Play configuration tool consists of Algorithm modules, Sensors (IR, UV, Radar and Ladar), Mirror Models, Platform Structures and Optical Path Models.

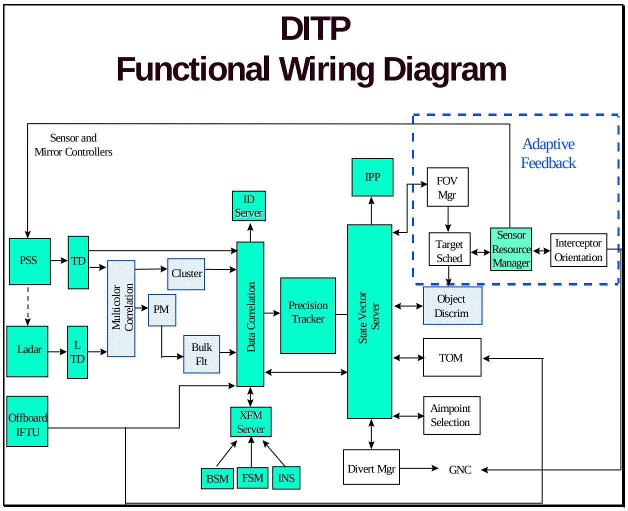

Figure 6: DITP Wiring Diagram

6 INTEGRATION OF FPSIM WITH HIGH PERFORMANCE COMPUTER (HPC)

The two major parts of the DITP mission are being developed at Rome. The first effort is the development, testing and integration of the algorithms into FPSim. The second is the development of the real-time processor Wafer Scaled Signal Processor (WSSP). As part of the development process the algorithms are ported from the FPSim to the High Performance Computer (HPC) in preparation for the final transition stage to the WSSP (See Figure 7)

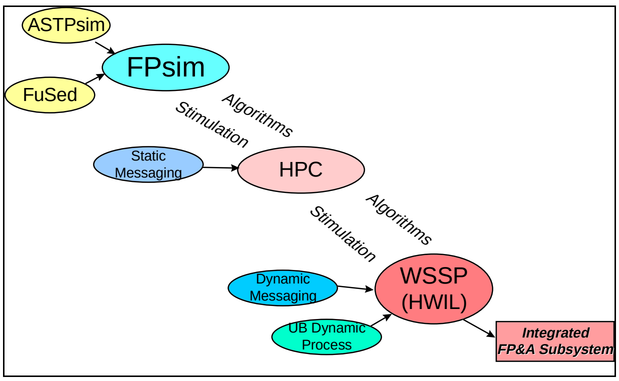

Figure 7: DITP Algorithm Development

The HPC environment (Intel Paragon 320 processor nodes) will allow for further testing of the algorithms while stimulating the image processing algorithms with data in real-time. The HPC Stimulation will take the sensor data developed while running the FPSim and insert the data into the algorithms and closed loop tracking systems in real-time. The algorithms on the HPC will pass data and information through a Message Passing Interface (MPI). The core algorithms tested and developed in the FPSim are used in the HPC environment. This is accomplished through wrappers that interface with the HPC-MPI environment in order to facilitate the same data flow, which exists on the FPSim. The wrappers receive the input message using MPI routines and transform the input message into expected format the DITP algorithm. The wrapper then invokes the algorithm and transforms the algorithm output into an output message. MPI provides synchronous and asynchronous message passing. Asynchronous is advantageous for large messages (e.g. images) when it is desirable to overlap communication and computation. This HPC Stimulation will be the testing area for all the algorithms as the WSSP is developed and the algorithms transition to the DITP Demonstration System.

7 PROPOSED ENHANCEMENTS

In order to make the FPSim more accessible and flexible the Plug and Play interface written in C, X-Windows environment could be re-written in Java. This would allow the Graphical User Interface (GUI) of the Simulation Configuration Tool to run from any computer system running a browser, i.e. Netscape, and connected to the Internet. The FPSim run-time display tools and GUI for running the simulation can be re-written in J-Wave and the user could run the entire FPSim from a browser. J-Wave is a commercial tool that allows PV-Wave routines to be accessed and run from a browser. The host computer (server) at the Rome site would allow remote clients to connect to the server and use the FPSim. The simulation would have the same interface and tools as if running on the host computer. This would allow remote users the capabilities to develop, test and analyze configurations and algorithms in the FPSim.

8 CONCLUSION

The FPSim is a valuable tool for Algorithm testing, Sensor performance as well as overall DITP system performance prediction. By utilizing a step by step approach for the testing and improvement of the algorithm suite from a workstation environment to a parallel computer, to the final dynamic messaging high speed parallel computer, lowers the technology risk for the entire program. Enhancements would allow the user to vary system parameters to better determine the optimal sensor configuration. The FPSim provides an invaluable and cost effective, risk reduction, tool in for continuing technology development.

ACKNOWLEDGEMENTS

This work was supported by the U.S. Dept. of Air Force under Contract DSA-95-C-0059-013

REFERENCES

Koplik, C. 1993. *AMDF Software user's Manual*. Volume 2. AFRL Rome Site: Technical Report.

Mabius, L. 1996. *BANNER CORE VERSION 1.0 RELEASE NOTES*. AFRL Rome Site: Technical Report.

Hobin, B. 1998. *Configuration Document for DITP*. Volume 1. Computer Science: Technical Document.

AUTHOR BIOGRAPHIES

MARK BARNELL is an optical engineer in the Systems Engineering Division of the Computer Sciences Corporation. He received a B.S. in Optical Engineering from the University Of Rochester. He is currently pursuing a M.S. in Computer Science at SUNY Institute of Technology at Utica/Rome. His experience includes large simulations and fire control algorithm development.

DOUGLAS WYNNE is an electronics engineer in the Air Force Research Laboratory/Information Directorate, Rome Operations Site. He received a B.S. in Physics from Utica College of Syracuse University. He is currently pursuing a M.S. in Computer Science at SUNY Institute of Technology at Utica/Rome. His experience includes Airborne Surveillance Radar development and program management in support of DITP/FP&A programs.