BFTT AMN APPLICATION OF STOW TECHNOLOGIES FOR CATEGORY 1 NAVAL TRAINING

CDR Peggy Feldmann
NAVSEA PMS 430
Crystal City, Virginia

Gary Kollmorgen
BMH Associates, Inc.
Norfolk, Virginia

Dave Cavitt
BMH Associates, Inc.
Norfolk, Virginia

The Battle Force Tactical Training (BFTT) Improvement Program is an in-port shipboard combat system team training capability to provide, a realistic unit level team training in all Navy warfare areas. This training is partially accomplished with stimulation to shipboard sensors via onboard trainers provided by tactical equipment program managers and simulation of non-shipboard forces such as friendly, neutral, and enemy aircraft and submarines. One of the first BFTT Improvement Program components is the BFTT Air Management Node (AMN). The BFTT AMN is designed to provide Air Traffic Control (ATC) and limited Air Intercept Control (AIC) proficiency training. This training is accomplished by integrating DARPA Synthetic Theatre of War (STOW) technologies with the existing BFTT architecture. This combination brings intelligent aircraft and a High Level Architecture (HLA) simulation base to the BFTT arena. This paper will discuss the STOW technologies and the software architecture designed to provide a robust ATC trainer to the fleet.

AUTHOR'S BIOGRAPHY

CDR PEGGY FELDMANN is the Technical Director in the Performance Monitoring, Training, and Assessment Office (NAVSEA PMS430). She has a Masters of Science degree in Acoustical Engineering where she worked in the development of fiber optic hydrophones. She was a project manager at the Naval Oceanography Research and Development Activity (NORDA) where she developed processes, procedures, and training for the Integrated Undersea Surveillance System (IUSS). She was the Assistant Officer-in-Charge of the Oceanographic Systems Support Detachment (OSSD) responsible for liaison between fleet IUSS units, the Systems Command program office, and defense contractors. In 1990, LCDR Feldmann was the Project Manager of the IUSS environmental acoustic performance prediction systems and as the Project Manager for the Fixed Distributed System – Deployable. She was the Program Manager of Synthetic Forces, Synthetic Theater of War (STOW) Advanced Concept Technology Demonstration for the Defense Advanced Research Projects Agency (DARPA). After gaining valuable advanced modeling and simulation experience at DARPA, CDR Feldmann reported to the Naval Sea Systems Command (NAVSEA) in 1997

GARY S. KOLLMORGEN is the Director of Operations with BMH Associates, Inc. He holds an MSEE and a B.S. in Operational Analysis. He is a retired US Navy Helicopter Pilot with over 10 years program management experience and over 5 years collectively in the development and application of campaign level simulation. He has an extensive background in U.S. Navy Operations which includes tours as a carrier Weapons Officer, amphibious assault ship Air Officer, and several helicopter detachment Officer-in-Charge. Mr. Kollmorgen was the project manager responsible for establishing an authoritative representation of the Navy for Modeling and Simulation (M&S) applications through the Knowledge acquisition/Engineering (KA/E) process for the Joint Simulation System - Maritime. He is also the BMH program manager for the BFTT AMN project that includes KA, software development, system integration, and Verification and Validation activities.

DAVID B. CAVITT is a research engineer with BMH Associates, Inc., Norfolk, Virginia, and is a Ph.D. candidate at Old Dominion University, Norfolk, Virginia. He received a BS degree in Computer Science at Old Dominion University. He has 10 years of experience in the use and development of simulations for military and engineering applications. His research interests include modeling and simulation, performance analysis, and distributed systems. Mr. Cavitt is a member of ACM and IEEE CS.

BFTT AMN APPLICATION OF STOW TECHNOLOGIES FOR CATEGORY 1 NAVAL TRAINING

CDR Peggy Feldmann
NAVSEA PMS 430
Crystal City, Virginia

Gary Kollmorgen
BMH Associates, Inc.
Norfolk, Virginia

Dave Cavitt
BMH Associates, Inc.
Norfolk, Virginia

BACKGROUND

Figure 1. Air Management Node "Cage" ¾ View

The Air Management Node (AMN) to the Battle Force Tactical Training (BFTT) on-board training system is a compact multi-processor simulation based training system capable of supporting Air Traffic Control (ATC) and Air Intercept Control (AIC) proficiency training. (See Figures 1 and 2) The BFTT system will integrate simulation generation capability to warfare sensor stimulation systems to enable personnel to train with their actual warfighting systems. The AMN was originally planned as a Human-Computer Interface (HCI) improvement for the Combat Simulation Test System (CSTS) AIC/ATC mode in response to feedback from the Fleet for a more "user friendly" trainer. It expanded to become not only an improved HCI, but also a simulation system in its own right. The engineering team of PMS 430, NSWC PHD Dam Neck, Litton, Lockheed Martin Tactical Defense Systems (Lockheed Martin), BMH, and Soar Technologies consulted technical experts from NSWCTSD, Pensacola and NAWCAD, St. Inigoes to identify AMN requirements that would ensure the end product met the users' training goals.[1] The AMN will allow ATC personnel to train to at-sea ATC procedures involved in the departure and recovery of aircraft and limited AIC aircraft control onboard carriers and large deck amphibious ships of the U.S. Navy.[2]

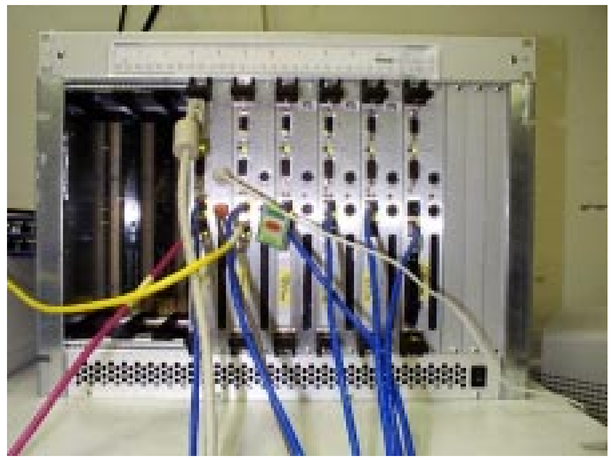

Figure 2. AMN "Cage" - Front View

The training accomplished is not intended to look at launch and recover techniques and/or deck loading in order to stimulate plans or documentation development for combat scenarios. It is designed to keep air controllers proficient at terminology and control procedures in the shipboard environment. The AMN is designed to train standard shipboard ATC controllers such as, Marshal, Departure, Approach, and Final controllers and AIC controllers such as Tactical Air Control Center (TACC) and Strike controllers.

Concept of Operation

While the AMN is currently funded for delivery to the USS Bonne Homme Richard (LHD 6), it provides capability for use on all large deck amphibious ships and carriers and at shore sites. The AMN creates simulated aircraft entities that result in the stimulation of onboard radar systems, which ultimately generate realistic controlled aircraft symbols at the trainee's console in the Combat Information Center (CIC). Verbal commands from the trainee to direct controlled aircraft movements are relayed to a BFTT Operator Position Console (BOPC) operator in the BFTT space via Improved BFTT Digital Voice (IBDV). The BOPC operator in the AMN environment is a "pseudo-pilot" who is knowledgeable in the proper ATC/AIC terminology. When the pseudo-pilot hears a trainee's command, he responds verbally with the proper response, as would a real pilot, and maneuvers the aircraft in response to the controller's (trainee) command via the HCI. HCI inputs (e.g., course changes) are interpreted by the AMN software, which makes the necessary corrections to the flight dynamics of the simulated controlled aircraft using sophisticated Joint Semi automated Forces (JSAF) modeling algorithms. JSAF is a High Level Architecture (HLA) Simulation System that was developed by the Defense Advanced Research Projects Agency (DARPA) for the Synthetic Theater of War STOW ACTD. The result of the kinematics modifications directed by the HCI are then translated through an HLA gateway to DIS information that stimulates the onboard radar systems, and the controlled aircraft symbol at the trainee's console moves in accordance with the aircraft maneuvers made at the AMN HCI. Only those radar systems provided with stimulators will be able to "see" the simulated aircraft. In addition to verbal commands, pseudo-pilots receive and similarly respond to automated Link-4A commands generated from the Data Link Auxiliary (DLA).

KA/E Defined Requirements

- Provide ATC Training for CVx and LHA/LHD ship types.
- Provide aircraft types that include Navy and USMC fixed and rotary wing as well as US Army rotary wing.
- Provide controlled Departures and Approaches (CV and LHA/LHD)
- Provide inflight behaviors that include Auto-commencing Approaches on time and in position, Voice Reports, and both "Challenge & Response" as well as standard reports required in an shipboard ATC environment.
- Provide for glide path accuracy ($\pm 1^{o}$ heading, $\pm 1^{o}$ glide path, ±10 Knots airspeed, 100 fpm rate of climb/descent) in order to correctly stimulate onboard radar systems.
- Provide for Speed/Heading/Altitude changes based on published approach procedures or as directed by Carrier Air Traffic Control Center or Helicopter Direction Center (CATCC/HDC) controllers
- Provide for emergency behaviors that include failure to commence approach on time and/or out of position, inoperable IFF, divert, fast approach, TACAN failure, Delta, and Navigation system failure (no-gyro).
- Provide instrumentation that includes IFF Modes and Codes where all IFF modes shall be adjustable and simulated aircraft shall report all IFF modes and codes through standard shipboard C4I system and speed, heading, and altitude.

TECHNICAL CHALLENGES

Several technical challenges (listed below) were levied at the development team. Through Knowledge Acquisition and Engineering practices, user input, and leveraging existing technologies these challenges were met head on. Most of the technology used to provide the simulation portion of the AMN was leveraged from the Synthetic Theater of War (STOW) program. This included the HLA-DIS gateway and the Joint Semi Automated Forces (which includes the intelligent agents used to "fly" the synthetic aircraft in the AMN). **THE CHALLENGE** was to integrate these technology pieces into an existing DIS training system.

GUI

Requirements

The AMN was required to reduce "Pseudo Pilot" workload from current CSTS training capability. On large deck aviation ships (LHA/LHD, CVx) three to four BFTT Operator's Consoles (BOPC) are available. Based on this physical constraint and the "typical" aircraft controlled by a given ships air traffic control center, a threshold of 34 and a goal of 41 aircraft controllable by no more than three pseudo pilots was required.

The AMN HCI was also required to provide the method of entering the AMN specific data by the pseudo-pilot for a BFTT exercise. This included frequency data, reference point data, creation and control of aircraft, maneuvering the aircraft in response to trainee commands, and customization of the pseudo-pilot display.[1]

Fleet feedback on current BFTT human interfaces was also considered a driving motivation for reduced workload.[3]

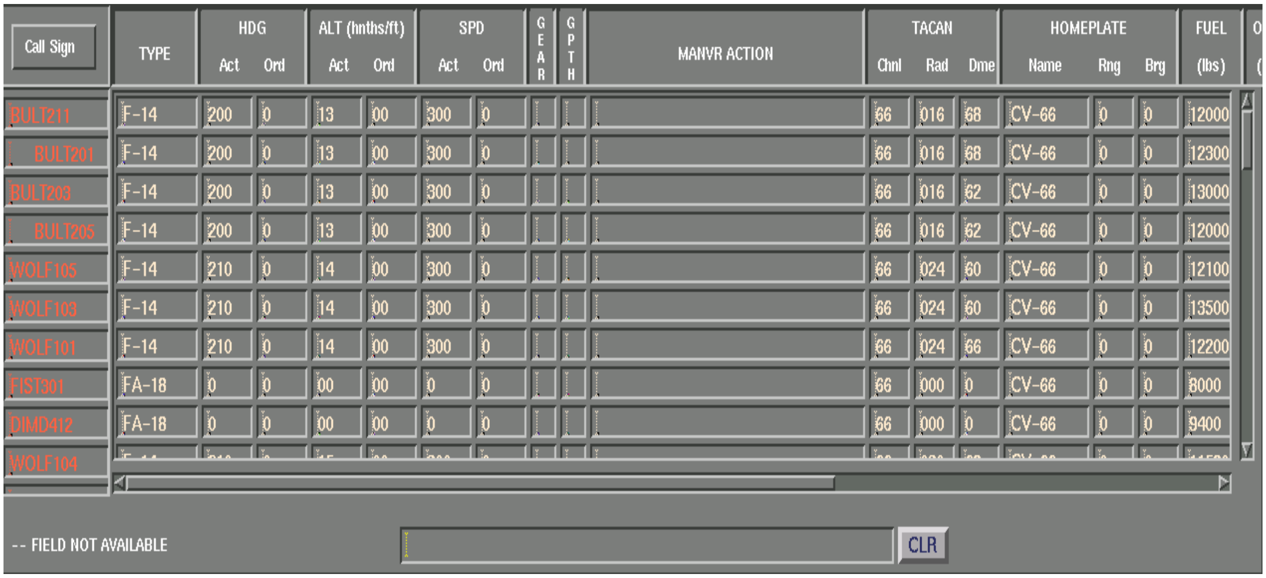

Figure 3. Representative Example of Operating Human Computer Interface (HCI).

Solution

The AMN Human Computer Interface was conceptualized and designed to reduce the workload of the pseudo pilot (see Figure 3). Single button clicks were designed in to the system as much as possible to allow ease of use and quick responses.

The AMN HCI processing is a software library that is embedded into the Joint Semi Automated Forces (JSAF) process. This allows ease of data sharing between the HCI and JSAF. It also simplifies updates of the pseudo pilot display and polling for messages from the AMN server and other AMN HCIs. The AMN server is a control agent for the receipt of Link-4A data from the BFTT Link-4A stimulator. The AMN server receives Link-4A messages and forwards them to the appropriate AMN HCI for processing. In addition, the AMN server controls the display of the AMN HCI as well as the frequency, point, and aircraft sections of the HCI.[1]

Dynamic STOW/SOAR Technologies

Requirements

Intelligent agents were another design option to reduce the workload of the pseudo-pilot. JSAF includes intelligent aircraft agents based on the Soar AI language, referred to as TacAir-Soar agents, or just *agents* here. "Dynamic" agent creation allows a user to create agents on an existing SAF process without the need to re-start the process each time (a limitation within JSAF prior to AMN development). The current JSAF implementation involves specifying the agent scenario and necessary agent files before running the SAF process. This means that a SAF process can only run one event in the scenario and when the agents are finished, the SAF process must be restarted to run another event.

To meet the AMN requirement of starting and stopping agents throughout a training scenario, a design was developed to allow a remote process to specify where the agent files are located, create an agent in a running SAF process with those files, and allow agent deletion.

Solution

Soar agent creation involves various steps. The "before and after" implementations are elaborated below.[4]

In the implementation prior to AMN:

1. The user specifies the mission parameters using the Exercise Editor.
2. This data is saved out into an exercise scenario file and a set of agent files, sometimes called mission files, for each agent in each event in the exercise.
3. The SAF process is run and a command line option specifies which exercise and event to load.
4. The SAF process runs until all the agent in that event are done, then the SAF process is re-run with a new event.

In the AMN implementation

1. The user can specify the mission parameters using the AMN HCI. There will be a set of default scenario and agent template files that will be merged with the data from the HCI to generate the final agent and scenario files.
2. The agent and scenario files will be stored out into a directory and a file with the path names and commands on how to create each of the agents, specific points and any other scenario data will be created.
3. Agent creation, deletion or reset commands will be sent to the Agent scheduler as needed.
4. The Agent scheduler will create, delete or reset the agents on the various SAF processes as needed.

Processes Description

There are several components in the dynamic agent creation design (see figure 6).

Agent Scheduler / Agent Command Parser

The agent scheduler consists of two components. One that exists within the HCI/SAF called the Agent Scheduler, and one that exists within the SAF/Soar process called the Agent Command Parser. Each component performs different functions.

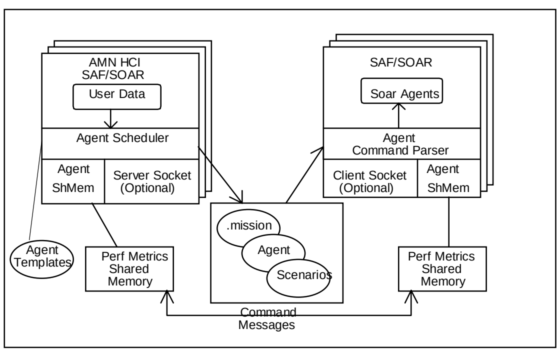

Figure 4. Dynamic Agent Creation High Level Design

Agent Scheduler

The Agent Scheduler is tasked with taking data from the HCI, generating agent files based on templates, scheduling agent creation actions for the SAF processes and sending agent creation, delete or reset commands to the SAF process. The Agent Scheduler connects to the shared memory partition to transfer agent create commands. Through the interface the agent scheduler will know about all of the SAF and HCI processes that exist in the AMN node. It will know the load of the various SAF processes and know when the SAF processes are active. The scheduler uses a round robin scheduler approach in deciding in which SAF process to create the agents for load sharing.

Agent Command Parser

The Agent Command Parser runs on the SAF/Soar process and connects to shared memory and checks for incoming create commands. Upon receiving a create agent (aircraft) or create point (such as CAP point or Marshal Point) command the agent command parser will call the SAF/Soar command parser to create agents on the SAF processes. It will then issue a response command to acknowledge that the agent has been created.

Aerodynamics

Requirements

The fidelity of the aerodynamics model that would drive the aircraft entities for the AMN had to meet the radar stimulation requirements as shown earlier.

The existing JSAF implementation used two fixed-wing aerodynamic models. The model used by TacAir-Soar agents did not provide a functional interface sufficient for close-control of the physical models by the agents. The second, and newer flight dynamics model provided a more robust model control interface and greater model resolution. However, it had only been tested and used in a limited context (i.e., JSAF taskframe-based Fixed Wing Aircraft (FWA)) and had never been integrated for use with TacAir-Soar. Model analysis and preliminary validation concluded that the newer flight dynamics model, when used by the TacAir-Soar agents, could provide sufficient model fidelity for simulating the fixed-wing aircraft required for ATC training.

Solution

Unlike the old flight dynamics model, the new model has much finer resolution in turn, acceleration/deceleration, climb and descent rates and incorporates a control interface to provide aircraft-specific commands. The high resolution of rate control is essential in the ATC environment, especially for precision approaches. The validation of this new FWA flight dynamics remained to be addressed to satisfy the AMN requirements.

Initial Test

Initial testing of the new flight dynamics revealed a number of problems. Aircraft did not always execute given commands and performance was not stable. Oscillations of speed and altitude took place and occasionally as well as unpredictably the model became completely unstable

Technical Approach

Rather than using standard debugging techniques, a specially designed driver program was written to test the new FWA flight dynamics model. This approach allowed us to isolate the simulation of the flight dynamics itself from any external influence. Inputs to the driver program were all the commands that the AMN could issue to the simulated aircraft. Outputs were the flight variables such as position, velocity, heading, pitch, etc.

Control and visualization were achieved by using a single GUI (see Figure 5) for the both input commands and output display. The commands were entered to the flight dynamics simulation by typing the relevant parameter in a command window and pushing a button. The outputs, numerical vice graphical, were constantly updated on the screen and copied to a specified log file. It was possible to run the model in a continuous or step mode. Using this single GUI it was possible to test all of the flight regimes necessary to support the AMN.

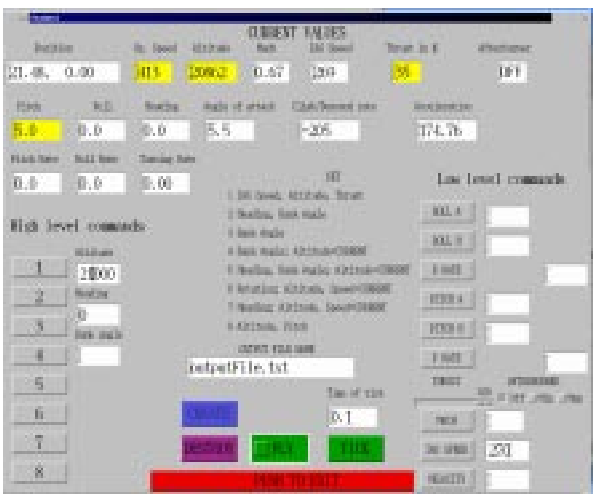

Figure 5. Flight Dynamics Model Control GUI

Error Discovery and Optimization

Errors of all kinds were revealed using the driver/GUI such as wrong unit conversions, arithmetic errors, wrong function calls, etc. Some errors were significantly more serious and required changes in the code. Low-level commands that influenced only one aspect of the model caused few and only minor problems. The high level commands that attempted to achieve multiple goals such as constant velocity/new altitude caused significant stability problems and required rewriting and testing of significant portions of the model's software code. See Figures 6 and 7 that demonstrate the before and after effects of the model trying to maintain multiple goals, in this case constant velocity while climbing to a new altitude.

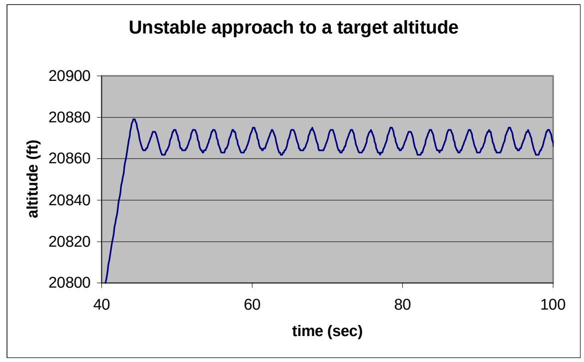

Figure 6. Incorrect approaching to the altitude target 21,000 ft.

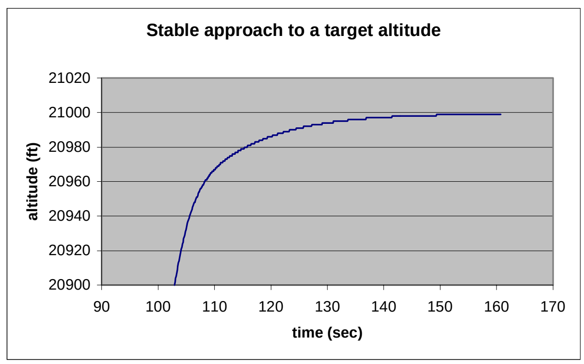

Figure 7. Correct approach to the altitude target 21,000 ft.

The code was also optimized and a reduction in the number of operations, especially the floating point calculations, was possible and resulted in a more stable faster performing flight dynamics model.

Results

After all the modifications/optimizations, the new flight dynamics correctly responds and executes all the commands. It produces stable, oscillations free flight performance. After the code was corrected and optimized, the driver/GUI was used to refine the parametric values required for defining the flight of particular planes (e.g. F/A-18).

The new FWA flight dynamics model has been successfully integrated with the TacAir-Soar implementation. Initial testing has provided favorable results regarding fine-grained control of the FWA physical model by the agents. On-going tests will continue to focus on the run-time use of the model interface, the objective being to make sure the Soar agents are able to exhibit correct simulated flight behavior.

Behavior Representations

Requirements

The requirements for behavior representations are listed as part of the KA/E Defined Requirements above. Essentially the aircraft had to be able to display all the characteristics that would be expected of Rotary Wing Aircraft (RWA) and Fixed Wing Aircraft (FWA) in a U.S. Navy ATC environment.

Solution

Soar agents have never operated in the ATC environment so extensive Knowledge Acquisition was performed in order to build upon the behaviors currently residing in JSAF Soar agents. This KA resulted in four major functional description documents. These documents were Functional Description for Approach & Departure Communications, Functional Description for Perform Instrument Departure From A Ship, Functional Description For Perform Instrument Approach To A Ship, and Functional Description For Link-4A. The descriptions were used to develop the complex behaviors as well as a basis for validation of the resulting behaviors. The documents were reviewed by the ATC schoolhouse for accuracy and completeness prior to being used for development. They have not been validated as of this writing.

HLA/DIS

Requirements

HLA Compliance

The AMN shall provide the initial step in the BFTT migration toward HLA compliance. A Distributed Interactive System (DIS) to HLA gateway shall be used to translate new HLA compliant communications to and from legacy BFTT DIS communications.[5]

Solution

Although the BFTT program is looking to move to the DMSO standard RTI 1.3, JSAF has not currently moved to this standard thus RTI-s, the RTI developed at Lincoln Labs for the STOW ACTD was used for gateway development.[5] Another requirement of the HLA gateway is to use the JSAF Federation Object Model (FOM).

Figure 8 and 9 show the hardware architecture for the BFTT AMN and figure 9 also depicts the gateway implementation for the AMN. The BFTT DIS network currently uses the IEEE 1278.1 DIS specification. This HLA-to-DIS gateway provides an interface to the BFTT DIS LAN via an Ethernet connection and FDDI Bridge (the BFTT STOW LAN is FDDI-based). Other BFTT AMN requirements related to dynamic relocation of the exercise center resulted in modifications to the gateway to control the location of the exercise *playbox* within the STOW synthetic environment. There are several significant points regarding the design and implementation of the HLA gateway. First is the utilization of established FOMS. The effort required to design, implement, test, and optimize (important when using DDM) a FOM can be significant. The reuse of the JSAF FOM was a cost-effective path for integrating JSAF with the DIS BFTT system. Another significant characteristic of the gateway design is the reuse of a core set of STOW source code and libraries thereby benefiting from the ongoing JSAF life cycle. A significant point to using this gateway approach is it is proven technology in connecting DIS compliant aircraft simulators as part of a Distributed Mission Training experiment at the Air Force Research Laboratory in Mesa, AZ.[6]

The gateway is powered by a single processor in the AMN cage (Figures 1 and 2 show the actual prototype AMN cage.)

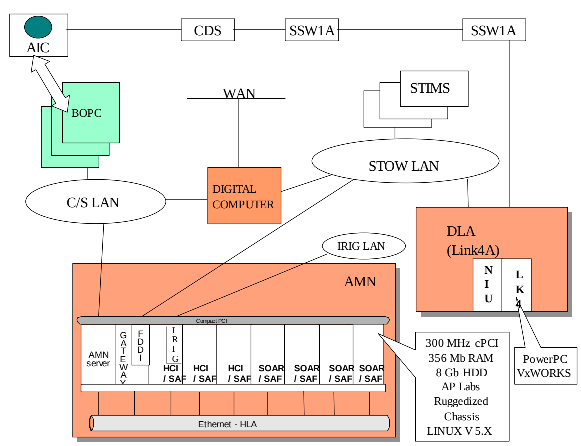

Figure 8. High Level Architecture Design of BFTT AMN

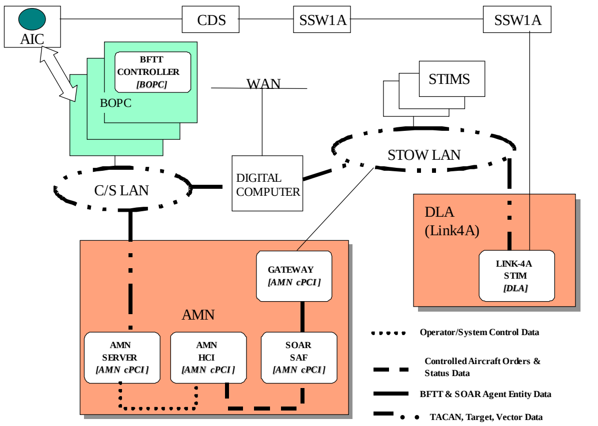

Figure 9. Detailed View of AMN Networking Topology

PROGRESS

The AMN is nearing completion and is currently undergoing rigorous reliability/stability testing for planned delivery to the Bon Homme Richard (LHD 6) in late summer/early fall 1999. The current AMN implementation is near completion and integration testing is focused on increasing the reliability and usability of the system. Additionally, the tests are evaluating the operational training capabilities of the architected system. Early reports from the fleet indicate that they are anxiously awaiting its delivery.

The HCI has been difficult to deliver in a fashion that will please all users (just ask Microsoft on the difficulties of GUI development). However, it has met all challenges and will most likely receive significant feedback from the fleet on even more work reducing and "user friendliness" tweaks.

Requirements for dynamic agent creation, deletion, and control represent a significant enhancement to the existing STOW JSAF implementation. The design and development of the integrated agent scheduler and parser, and the reuse of the PerfMETRICS architecture provides a readily extendible agent control mechanism.[7,8] The run-time feedback provided by performance monitoring supports automated load scheduling which can lead to better run-time performance. The successful implementation of this design now appears to be one of the most stable components of the current AMN implementation. The aerodynamics models have proven to be robust and capable for ATC training. Future enhancements to meet requirements for simulating close in air-to-air engagements can benefit from the design and validation tools created for the AMN project. Changes to the flight dynamics models could include algorithmic modifications and will most likely require parametric changes to account for the higher agility flight regimes of close-in combat. The methodology used during this work is a useful case study. Behavior enhancements have proven to be less than straightforward. It is difficult for the software developer to "see" all of the intricacies in handling an aircraft in the ATC environment. The functional description documents were instrumental in reducing the development time. However, even the most capable system engineer does not always anticipate the differences between the real world flight activities and those that must be considered in the simulated world. Many technical exchanges between the behavior developers and the subject matter experts nailed down the missing pieces. Additionally, simulated entities (e.g., Soar agents) react according to published doctrine. Published doctrine often lags changing fleet procedures. Every attempt was made to include fleet input to minimize the risk of delivering "out of date" Soar agents.

The HLA/DIS gateway has proven to be a reliable component of the AMN. While some translation adjustments had to be made, the technology that was first demonstrated with the U.S. Air Force DMT Experiment has remained virtually unchanged for this implementation. Additional work should reveal current discrepancies related to DIS PDU bundling used to optimize traffic flow (i.e., bandwidth reduction) on the BFTT LANs / WANs

SUMMARY

This is the first major attempt to transition STOW technologies to an existing fielded program. The system has produced only limited fleet feedback to date, the successful integration of STOW technologies have shown the feasibility of using a high-resolution, high-fidelity synthetic environment to meet the Navy's ATC training requirements. The BFTT AMN can also serve as a use case for other Navy programs considering the use of STOW technology to meet intership and intraship training requirements. Key issues still exist related to the current AMN implementation, its stability, and reliability for use in a shipboard environment. Pending the successful completion of integration tests, the AMN will be ready for shipboard installation. Validation of the AMN capabilities will help prove the soundness of the procedures used to implement STOW technologies and to involve the fleet in future system development.

BIBLIOGRAPHY

[1] Lockheed Martin Tactical Defense Systems, Eagan, Battle Force Tactical Training (BFTT) Air Control Training Capability Concept Paper for the Integration of the Air Management Node into BFTT, Draft v1.0, Nov 1998 for Naval Surface Warfare Center – PHD, Dam Neck Detachment

[2] BMH Associates, Inc., Battle Force Tactical Training (BFTT) system Air Management Node (AMN) Requirements Analysis, v1.01 16 Nov 1998 for Naval Sea Systems Command (NAVSEA) PMS-430.

[3] USS JOHN F KENNEDY MESSAGE, SUBJ: COMBAT SYSTEM STATUS, DATED: R 241501Z JUL 98

[4] Kenny Patrick, Soar Technology, Design Document for Dynamic Agent Creation, v1.0, Nov 1998.

[5] Calvin, J.O., C. J. Chiang, S.M. McGarry, S.J.Rak, and D.J. Van Hook. 1997. Design, Implementation, and Performance of the STOW RTI Prototype (RTI-s). In Proc. 1997 Spring Simulation Interoperability Workshop. 97S-SIW-019.

[6] Cavitt, David, B., M. Alan Gibson, Edward P. Harvey, 1998. HLA/STOW Interface Development To Support Distributed Mission Training (DMT), 1998 Fall Simulation Interoperability Workshop. 98F-SIW-204.

[7] Cavitt, D.B., C.M. Overstreet, and K.J. Maly. 1997. A Performance Monitoring Application for Distributed Interactive Simulation (DIS). In Proc. 1997 Winter Simulation Conference, December 1997, 421-428. Association for Computing Machinery, New York, NY.

[8] Cavitt, D.B., J. Bell, M. Checchio, C.M. Overstreet, and K.J. Maly. 1998. Performance Monitoring for the Design, Configuration, and Control of DIS/HLA Exercises. In Proc. 1998 Spring Simulation Interoperability Workshop. 98S-SIW-066..