Embedded LVC Training:
A Distributed Training Architecture for Live Platforms

Jaclyn Hoke, Jason Wenger, Brian Wolford
Rockwell Collins, Inc.
Cedar Rapids, IA
jahoke@rockwellcollins.com, jcwenger@rockwellcollins.com, brwolfor@rockwellcollins.com

ABSTRACT

Given the reductions in Department of Defense budgets it is imperative that every dollar spent on training warfighters be used in a cost efficient manner. One approach for cost effective training is distributed training exercises that include live, virtual, and constructive participants, but injecting the training functionality into live aircraft platforms is challenging. Many of the current architectures and approaches for presenting the information to a pilot require modifications to the Operational Flight Program (OFP) software. This is an expensive approach that can be challenging and time consuming to certify for flight safety. Ongoing research and development in embedding distributed training functionality within flight hardware has led to a new architecture that is presented in this paper. This research system demonstrates a partitioned architecture for embedded training that interfaces with the OFP through a single, standards based hook, allowing training functionality to be injected into flight displays in a manner with a credible path to certification.

In addition to illustrating the architecture, this paper explains how the approach provides the capability for the end user to train with systems and sensors that are not physically present on the platform, such as the multiple radar simulators currently integrated. These onboard simulated sensors and systems consume pilot inputs as well as participant state data and interactions sent over a datalink, enabling embedded distributed training on live platforms in exercises that can contain combinations of live, virtual and constructive (LVC) participants. The results of test bench experiments are provided, and the planned flight test experiments that will be conducted during LVC exercises are described. Finally, the paper discusses research that will leverage the system, steps to further mature the proposed architecture, and the foreseeable challenges with fielding this approach to enabling embedded training.

ABOUT THE AUTHORS

Mrs. Jaclyn Hoke is a Senior Systems Engineer with the Rockwell Collins Advanced Technology Center System Virtualization Group. Jaclyn has been with Rockwell Collins for five years. Her primary background is in developing software for image processing applications, but the last two years have been focused on maturing technologies and developing architectures for Live Virtual Constructive systems. Jaclyn has a Bachelors' Degree in Applied Mathematics, Master's Degree in Electrical and Computer Engineering and is currently pursuing her Doctorate in Electrical and Computer Engineering.

Mr. Jason Wenger is a Senior Systems Engineer with the ATC Systems Virtualization Group of Rockwell Collins, Inc. Jason has been with Rockwell Collins for nine years and has worked in Modeling and Simulation throughout the span of his career. He has been involved in modification, instrumentation, and flight test efforts on platforms ranging from Beechcraft Bonanza, Bombardier M145 and Global Express, to Boeing 727, 757, and C-135 Stratolifter. Jason has a Masters' Degree in Electrical and Computer Engineering with a focus on software engineering from the University of Iowa.

Mr. Brian Wolford is a Senior Software Engineer with the Rockwell Collins Advanced Technology Center Systems Virtualization Group. Brian has been with Rockwell Collins for seven years. His primary background is in embedded software development for various applications. This includes Live Virtual Constructive applications, and DO-178B qualified avionics software. His experience with Live Virtual Constructive includes applications for distributed pilot and soldier training, insertion of distributed training protocols into embedded environments, and simulation frameworks. He has avionics applications experience in a wide range of areas including: primary flight instruments, situational awareness displays, flight management systems, and synthetic vision. Brian has a Masters' Degree in Systems Engineering and a Bachelors' Degree in Computer Engineering.

Embedded LVC Training: A Distributed Training Architecture for Live Platforms

Jaclyn Hoke, Jason Wenger, Brian Wolford
Rockwell Collins, Inc.
Cedar Rapids, IA
jahoke@rockwellcollins.com, jcwenger@rockwellcollins.com, brwolfor@rockwellcollins.com

INTRODUCTION

In 1997 the U.S. Air Force identified key shortfalls in the ability to safely and affordably train aircrew, including safety considerations, mission complexity, airspace and range restrictions, and real world commitment and costs (U.S. Air Force, 1997). Since then, various organizations have invested in technologies to mitigate these training gaps. In 2003, the U.S. Air Force Distributed Mission Operations (DMO) Concept of Operations (CONOPS) identified the objective to "train warfighters as they expect to fight; maintain combat readiness at home or deployed; conduct mission rehearsal in an environment as operationally realistic as necessary; and provide support to operations" through a combination of real-world operational systems, simulators, and constructive simulations (U.S. Air Force, 2003). There are now several centers and networks currently online, including the Distributed Mission Operation Network, Air Reserve Component Network, Distributed Mission Operations Center and the Distributed Training Operations Center. Unfortunately, the reduction in Department of Defense budgets leading to reduced flying hours and the struggle of the existing ranges to support new combat capabilities mean that many aircrew are still unable to achieve training minimums.

In fact, a 2011 RAND report clearly states that shrinking resources and expanding mission requirements are jeopardizing the ability to meet proficiency standards to accomplish wartime missions. The report also cautions that reducing the number of flight hours and increasing the number of simulated missions only shifts the expense to the simulator environment because the value (fidelity) of training must be maintained (Ausink, Taylor, Beigelow & Brancato, 2011). Data compiled by RAND indicates that the high costs of training are largely driven by the need to field red forces and is further compounded by the need to ensure that these red forces are effective training adversaries. For the F-22 alone, it is estimated that it would cost an additional $63 million for T-38 aggressors, $132 million for F-16 aggressors or $593 million for F-22 aggressors each year above what is currently spent just to meet the current training requirements (Ausink, Taylor, Beigelow & Brancato, 2011).

Experts see increased use of simulators; the DMO; and live, virtual, constructive (LVC) training as a means for reducing this gap; in addition the RAND report concludes that "in the long run, development of the LVC ability to inject simulated and constructive threats into live aircraft may be the only fiscally responsible approach to improving training." Whereas significant research in the last 4-6 years has focused on the integration challenges of LVC, such as datalinks and cross domain solutions for training with allied forces, the ability to achieve the "injection" of virtual and constructive entities into live platforms is still a relatively new challenge for the research community. The Combat Air Force LVC Pilot Project is currently in the process of making significant modifications to the Operations Flight Programs (OFPs) of several F-15 and F/A-18 aircraft (Sidor, 2012). While significant OFP upgrades will most likely prove to be an effective solution, it is likely that it will also prove to be an expensive solution when retrofitting currently fielded aircraft and re-certifying them for flight.

Ongoing research and development in embedding distributed training functionality within flight hardware has led to an alternative approach for injecting virtual and constructive entities into live avionics displays. The research system demonstrates a partitioned architecture for embedded training that interfaces with the OFP through a single, Aeronautical Radio, Incorporated (ARINC) standards based hook. It presents the capability to train with systems and sensors not physically present on the platform and the ability to fully participate in Live Virtual Constructive (LVC) training exercises. The details of this architecture are presented in this paper, along with a description of the research that leverages the system and the next steps to further mature the architecture.

ARCHITECTURE

Research Architecture Philosophy

Several major goals influenced the research system design:

1. Graceful degradation during failure
2. Minimal modification to OFP
3. Federated architecture

Safety of flight concerns present a need to separate the level of criticality of the research training functions as completely as possible from functions needed for basic aviation and navigation. As these research functions are being developed without the full rigor of a formal certification process, care must be taken to ensure that failures of research functions cannot interfere with flight critical symbology. To this end, a partitioned architecture was developed. A clear separation of training function from flight safety function was made in software, in hardware, and in display formats.

The training system architecture provides three levels of reversion for research aircraft safety of flight. While in the training mode, large areas of the displays contain content from the research training function with very low level of criticality. Safety of flight is ensured through reserved portions of the displays that are kept for primary flight instruments, and which are presented through a processing chain derived from a certified baseline. In the event of a software fault within the prototype training function, these reserved areas continue to operate normally while the remainder of the display is blanked. In the event that a fault occurs, a dedicated line key selects a reversionary mode featuring a full screen standby format that is again derived from a certified baseline. Finally, a standalone set of standby instruments is physically and electrically separated from the research displays within the cockpit.

Two 6”x8” Multi-Function Displays (MFDs), mounted in portrait orientation, were used in the final architecture. The selected displays are standard production units for a military rotary wing flight deck. Each MFD runs a research-tailored OFP which was derived with minimal modification from an existing certified flight deck. The OFP application consists of a processing block which communicates with aircraft data sources and performs source selection and filtering for a minimal baseline set of flight parameters regarded critical, such as aircraft attitude, speeds, position, etc. A primary flight display format was taken from a certified baseline, again with minimal modification, and designated as a standby format. When this format is active, system partitioning ensures that no training function symbology is permitted to be rendered anywhere on the MFD.

The primary modification to the certified baseline occurred within the window management system. When the training system applications are active, a line select key on the standby format allows the pilot to select the training format. When the training format is active, a portion of each MFD display area is reserved for a compressed version of the standby format, while the remainder of the display is available for presentation of training formats.

As the training format symbology is generated by software with a lower level of criticality, partitioning within the rendering subsystem again ensures that no training function symbology is permitted to be rendered within the dedicated area of the screen reserved for the compressed standby format. In addition, to prevent a hazardous or misleading presentation to the pilot, several restrictions on the training function applications are enforced through development practice. No functions that present attitude data to the pilot are permitted to be implemented within the training format processing chain. Speed, altitude and position data are permitted in the context of a training function, such as a target closure speed, or ownship position relative to bullseye on simulated radar, but presentation of this data is not allowed to be visually similar to a primary instrument.

Hardware Architecture

System Function Allocation

Three major computing and display devices provide the computing resources necessary for the research flight deck. Two MFDs provide a display surface and a high assurance data path for critical flight data, while a Dzus-mount mission processor and data transfer unit (MP/DTU) provides training functions. The processing required to perform training functions on the research flight deck is hosted on a 7448 processor card installed within the MP/DTU. Two additional mission processors are available, one within each MFD, for future expansion, and are currently untasked. This architecture is illustrated in Figure 1. System Architecture

The entire system is installed in two configurations. First, the system can be operated in a simulator, where control inputs are provided to a flight model and outside viewing is provided by an image generator and screen. Second, the system is installed in an Aero L-29 Delfín jet trainer. While not matching the performance of a modern fighter, its speed and dynamics still represent a stepping stone in that direction. Additionally, this aircraft provides an excellent cost per hour to operate. For this reason, it is being used in our research as a proxy for an introductory jet trainer. For the purposes of this section, the live aircraft architecture will be illustrated. Differences in the simulator configuration will be discussed later in the section titled “Testing”.

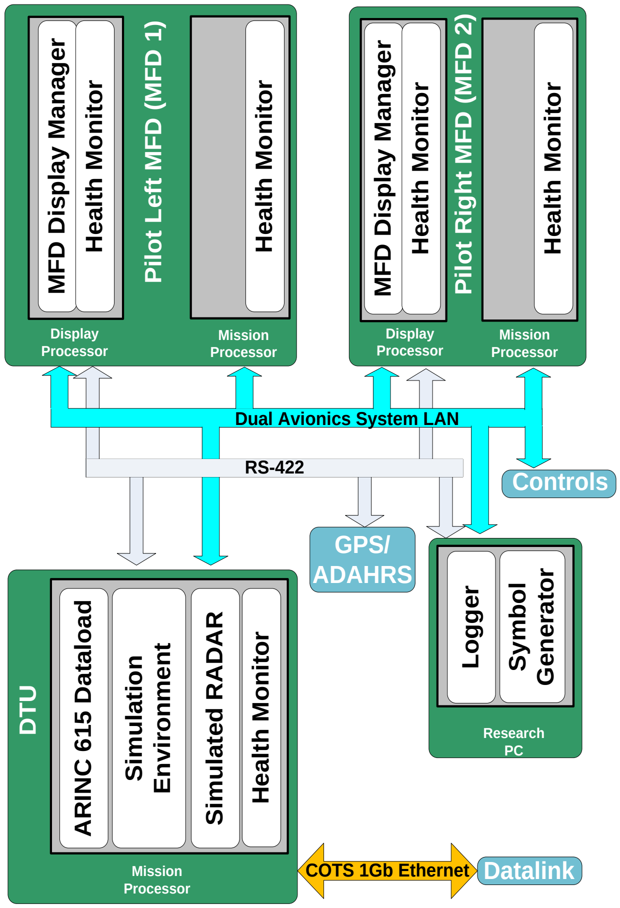

Figure 1. System Architecture

The system's source for flight critical data is a lightweight Air Data Attitude Heading Reference System (ADAHRS) originally intended for Unmanned Aerial Vehicle (UAV) applications. It produces an RS-422 serial data stream that is bused to both MFDs, the training mission processor, and components of the research infrastructure performing data logger and a Heads-Up Display (HUD) symbology generator functions. The cost and form factor of conventional optics HUDs prohibit their use in the research aircraft. In its place, a daylight readable LCD display is installed at the center of the glareshield, and functions as a HUD repeater device. This repeater display is necessarily opaque and non-conformal.

All interactions between the MFDs and the training application hosted on the Mission Processors are arbitrated by an ARINC Graphics Server (AGS) application which runs on the MFD Display Manager Partition on each MFD. The AGS application is responsible to ensure partitioned access to the display surface itself. Layer and window management within the AGS application and its configuration files provides the rules to ensure that the training function is properly partitioned from critical flight symbology; that the training application is allowed to present displays to the pilot when conditions are proper, and more importantly to ensure that training symbology can never corrupt or obscure symbology of higher level of criticality. This single, open, ARINC 661 standard-based hook into the OFP allows a flexible, lower cost path to integrating a training function as compared to a traditional tightly-coupled OFP integration, while the partitioning inherent in the system allows the training application to be developed to standards consistent with the lower criticality of the training function.

Within the cockpit, communications among the MFDs and the training mission processors is carried on a Dual Avionics System LAN (ASL), an ARINC 664 based Ethernet network. As this Ethernet network carries only the remainder of the training data that is not critical for flight, it is permitted to contain a mix of hardened Avionics Ethernet and COTS devices.

Training system control inputs are provided in two ways. Each MFD's display is surrounded on all four edges by a total of 30 line keys. Of these keys, 21 are available for use with the training format when it is active. Additionally, a simulated F/A-18 control grip and throttle are installed and an interface board converts the digital and analog signals from the Hands On Throttle and Stick (HOTAS) controls on these devices and provides switch state via the aircraft's Ethernet network.

Finally, a dedicated training datalink is connected via a second Ethernet interface on the MP/DTU. As datalink requirements often vary depending on the training system and site, we have architected for datalink agility. During the course of our research, we have integrated multiple datalink systems, operating on diverse radios, waveforms, bandwidths, and frequency bands, with link capabilities varying from full TCP/IP connectivity to narrowband, time-slice allocated, fixed-sized packets. These various datalinks have been integrated in the aircraft through one of two installation systems. First, Ethernet and power connections have been brought to an access cover location on the aircraft belly, allowing for installation of various datalinks and their matching antennas, each system pre-mounted on one of a set of interchangeable cover plates. Second, a set of engineered launcher rail adaptors allow fitting of either single or paired training pods conforming to the AIM-9 form factor on the aircraft's original under-wing pylons.

Deployment Considerations

Given the research intent of our training system, the processing resources are permanently installed in embedded training form. Migrating the mission processing functions, datalink, and in some cases, ADAHRS function, to a removable embedded training pod may enable a more flexible application of this system architecture to other aircraft. Additionally, in a system envisioned to provide training for multiple aircraft or roles, specialized hardware may be needed to accurately simulate systems used only in certain training curriculums. In this case, allocation of that specialized hardware into only a limited set of training pods would allow more flexible use of training equipment.

The use of a partitioned, standards-based ARINC 661 remote application architecture to integrate the training applications into the OFP enables this functionality to be allocated to an external pod. This contrasts with the allocation of the training function in a traditional tightly coupled OFP architecture, which could not be offloaded to a removable pod without risk to critical flight systems.

Software Architecture

Training Function Software Architecture

The aircraft systems simulation is in the form of a simulation kernel that loads and manages execution of a configurable collection of plugins, termed Simulation Elements (SEs). On the aircraft, each of these SEs simulates a device or system that would be present on the aircraft for which training is being performed. For example, a Fire Control Computer SE maintains information about aircraft kinematics, master mode and submodes, targeting, steerpoints, etc. A separate Weapons Inventory SE records the simulated stores loadout of the training aircraft. Another pair of SEs perform computations for prelaunch munition cueing and postlaunch simulated munition flyout. The modular nature of these Simulation Elements allows for selection of desired training function from a pool of available, interoperable training components.

All of these individual SEs communicate by publishing and subscribing variables on a Virtual Data Network (VDN). The VDN is the datastore for all training data that would be distributed on physical buses in the actual aircraft for which training is being performed. The VDN also carries all data about live, virtual, or constructive entities that have either been brought in from or will be published out onto the training datalink. All VDN state variables and distributed simulation data is available to SEs within the MP/DTU, as well as on the training Ethernet network through the use of a network VDN library.

One noteworthy feature of this architecture is the mechanism used to integrate the live aircraft state data needed for the simulation. All SEs used in the live aircraft are developed originally in a second instance of the system operated as a virtual simulator. In this case, a number of SEs exist to provide the basic flying model of the aircraft itself. For example a set of SEs which include Force and Moments, Equations of Motion, Aerodynamics, and Ground Model SEs simulate the virtual aircraft's interactions with its environment, while another set including Hydraulic, Electrical, Engine, and Fuel System SEs simulate the state of the aircraft itself. A data dictionary of standard bus values is populated from these SEs.

When the training function is run in the live aircraft, all these SEs are configured out of the system, and in their place, an ADAHRS interface SE is run which parses the RS-422 data stream. This data stream contains sensed state data and publishes the same set of variables that the above-mentioned SEs would publish. Thus, this SE is not so much a simulation element as a data bridge. In this case, the difference is minimal. In this way, both the data dictionary and the training SEs themselves are in fact identical between the virtual simulator and the live aircraft, with the only actual difference between the systems being the SEs used to either simulate or interface the aircraft state data. In this way, the difficulty of maintaining identical behavior between the live and virtual training assets is dramatically reduced.

Currently Supported Functionality

A number of SEs simulate aircraft functions in the training flight deck. As the primary areas of training research we have performed are close air support and air to air intercept operations, a matching set of simulation SEs have been implemented. The training SEs are as follows:

- FCC
- Weapons Inventory
- FaacCueingIntf
- FaacFlyoutIntf
- MFD
- VmtsRadarIntf
- HudIgInterface
- MfdDisplay661

<u>FCC:</u> The Fire Control Computer is a Model SE whose outputs are information about steerpoints, targets, master mode and submodes, and sensor of interest of the Flight Deck. Its control inputs include such things as mode selection and HOTAS control switch inputs.

<u>Weapons Inventory:</u> The Weapons Inventory is a Model SE whose outputs are information about remaining stores available for launch, selected

munition, and configurations of the specified munitions. For example, air to ground munitions can be launched with high drag / low drag setting, varying number of munitions dropped in a ripple/salvo configuration, etc. By contrast, air to air munitions record caged/uncaged, slave or boresight launch, etc. The processing function of the Weapons Inventory provides very generic weapon flyout and cueing models for situations when more specific SEs, such as the FaacCueingIntf and FaacFlyoutIntf described below, are not available.

FaacCueingIntf: The FAAC Cueing Interface is a Model SE that is a wrapper around library code provided by FAAC Incorporated. This is a reapplication of code traditionally incorporated into actual military OFPs, designed to provide prelaunch cueing in real operations (FAAC Inc.). In this case, the true weapon models have been replaced with representative unclassified models.

FaacFlyoutIntf: The FAAC Flyout Interface is a Model SE that provides a real time simulation of post-launch munition flyouts. This SE takes distributed simulation entity data from the VDN, which as discussed above, is consistent with the larger distributed simulation exercise, and provides that data to the flyout model in order to allow simulation of munition active guidance based on post-launch target maneuvering. Post-launch datalink support messages are also simulated, if applicable. This is again a wrapper around a third party library that is taken from a fielded non-drop range scoring system, with the high fidelity models replaced in our flight deck by representative unclassified models. The output of this model is the position and behavior of the simulated munition during its flyout. This data is pushed into the VDN entity datastore, and is distributed out onto the simulation datalink. In this way, Plan View Display maps at the instructor station can visualize the flight of the munition, and targets can, if applicable to the mission scenario, respond with appropriate countering behaviors during the munitions' flight.

MFD: The MFD SE is a Model SE that maintains information about the internal state of the MFD. Control inputs for this SE come from the Bezel SE, and are processed to determine the MFD state, such as page selection, configurable display settings such as overlay layers or optional data blocks. Two instances of this SE run, and model the internal state of the left and right MFDs. .

VmtsRadarIntf: The VMTS Radar Interface is a Model SE that simulates the function of an air-to-air fire control radar. The SE collects entity data from the VDN and passes this data to a set of radar simulation processes also running on the Mission Processor, which simulate the navigation, ground mapping, and air to air fire control capabilities of a representative fielded fighter radar. The output of those processes is a set processed of radar returns. These returns are then published by the SE, and are used by the HudIgInterface and MFDDisplay to present targeting and situation awareness data to the pilot. Control inputs to this SE include HOTAS control inputs and bezel key presses from the MFDs.

HudIgInterface: The HUD Image Generator Interface is a View SE that collects aircraft, target, and other state data and generates a CIGI data stream which wraps an IG-specific symbology packet format. These packets are then packaged and delivered to the IG, providing the information necessary to render the HUD display in a virtual simulator, or the HUD repeater display in the live aircraft simulator.

MFDDisplay661: The MFD Display 661 SE is a View SE that collects information from a wide variety of Model SEs and generates an ARINC 661 data stream which drives the AGS in the physical MFD in the live flight deck. In this case, due to the tight coupling of display and control data required by the ARINC 661 protocol this SE also incorporates the behaviors of a Controller SE, by processing bezel presses, and generating control commands which are sent to other SEs. A stores page allows display and selection of loaded munitions. A Horizontal Situation Indicator (HSI) format is available, which includes a presentation of waypoints and navaids recorded in an airborne simulation navigation database. Additionally, track files of datalinked blue force positions and sensed enemy positions collected from the VmtsRadarIntf are also provided on the HSI map. A radar format is available, which presents a simulated fire control radar scope. Finally, a training system status format is available, which displays real time information about the aircraft's ADAHRS health, datalink connectivity, and other health monitoring information.

Training Software Implementation Pattern

The training functions are implemented in software as a model/view/controller pattern. As an example, stores management will be detailed.

The Model is the Weapons Inventory SE. It loads a configuration file at startup that identifies the number of stores locations (pylons) on the simulated aircraft, and populates those stores with a configuration-selected default loadout. Through the duration of the flight, it manages changes to the inventory such as changes to the set of selected stores for launch, modes and settings of munitions, the removal of launched munitions from wing stores, and tracking launched stores in an airborne pool until the end of their flights. All of these settings are recorded in VDN variables published by the Weapons Inventory SE.

The View is a component of the MFD Display SE. The view presented by this SE collects and displays information from many models. In this case the View of interest is the Stores Management System (SMS) page. This SE subscribes to the variables published by the Weapons Inventory SE, and uses these variables to produce the display presented on the MFD. In this case, the necessary data is in the form of a set of runtime parameters of ARINC 661 widgets. The MP/DTU then sends these parameters via ARINC 661 protocol to an ARINC Graphics Server (AGS) process which runs on the MFD. With these parameters, along with a symbology definition file, the AGS has all the information necessary to render the visual presentation of the SMS page.

The Controller is a component of the Bezel SE. Bezel key presses are sent from the MFD in the return data of the ARINC 661 data stream. However, the MFD itself knows only the number of a bezel key, and not its semantic meaning. The Bezel SE subscribes to a number of variables, including the active page on the MFD, and information about bezel key labels, to turn this simple key number into a semantically meaningful request. The Bezel SE then sends this request to the Weapons Inventory SE, completing the control loop.

While these interfaces are internal to the software and are tailored to the specific functionality in question, the interfaces between model, view, and controller are well defined and provide a very clean partitioning of function. Thus, it is possible to replace components of the pattern with alternate implementations depending on research needs. For example, a replacement View implemented as an OpenGL window and a replacement Controller accepting input from a touch panel interface have been created, allowing a very lightweight but low fidelity virtual simulator to be added to the training system when higher participant count is needed.

Finally, due to the segregation of processing responsibility in the Model/View/Controller pattern, it is possible to run multiple instances of those functions within a single simulator system for debugging, demonstration, or monitoring purposes. For example, a virtual simulator might be configured to run the Weapons Inventory, MFD Display, and Bezel SEs as would normally be run in the live aircraft. This provides display presentation and interaction to the pilot on the physical display installed in the virtual simulator's main instrument panel. In addition, a repeater display for an instructor station may be added simply by adding the aforementioned OpenGL display and touchscreen controller SEs to that configuration. Due to the segregation of functions, there is no software configuration overhead required to extend the system to provide these duplicate displays.

Distributed Simulation Over the Datalink

A final noteworthy component of the simulation architecture is the connection to the datalink. In the live aircraft, a separate Ethernet interface is dedicated to the datalink. This allows segregation of data flows within the flight deck from data flows intended for distribution. The primary data flow over the datalink is information about the state of entities, munitions, and interactions in the distributed exercise, and the live aircraft's mission processor communicates externally using either standard Distributed Interactive Simulation (DIS) datagrams or via a High Level Architecture (HLA) Run-Time Infrastructure (RTI). The research architecture currently utilizes Real-time Platform Reference Federation Object Model (RPR-FOM) 1.0 and HLA 1.3, but has the capability to support other FOMs and HLA standards. For the purposes of these research and development efforts, the primary messages of interest are entity updates, fire and detonation interactions, emitter and jammer updates, and signals.

As might be expected, the network infrastructure is complicated and relies on a combination of several protocols, bridges, and translators. But an important feature of the overall architecture is that, once all these bridges and translators are navigated, there is a single, completely connected, HLA/DIS network connecting the various ground sites, which is bridged through the datalink, all the way to the flight deck itself. In this way, the live aircraft is connected to the distributed exercise in a manner functionally indistinguishable from a traditional virtual simulator participant.

The distributed simulation connection for the mission processor on the live aircraft is performed by an SE provided by the simulation environment infrastructure. This is not considered a "training" SE because it does not simulate an aircraft or OFP function. This SE is in essence a protocol bridge that cross-fills between the VDN entity list and the distributed simulation protocol. Depending on the environment needed, this SE interoperates with DIS or HLA. A TENA interface is also available but has not yet been used in a training study. Three separate SEs provide connection to these three protocols, but the SEs share a large pool of common source code, differing only in the protocol interface portion itself.

Incoming entity data from the exercise is processed by this SE, and an equivalent persistent entity is created in the VDN datastore. Subsequent data is then matched up against the existing entity in the VDN, and incremental updates are applied. Conversely, local entities created by the exercise, such as ownship position or locally simulated munitions in flight are identified, correlated, and published into the distributed simulation exercise by this SE. Equivalent paths are in place for Emitters, signals and other distributed entity persistent objects.

External events such as fire, detonation, collision, and simulation control are received from the distributed federation and are in turn converted to a common format and broadcast to various SEs within the exercise, via a configurable distribution list. SEs can also publish a locally generated event, which is likewise distributed to interested local SEs, and then converted and forwarded to the datalink.

When the simulator is used as a virtual participant, the simulator is directly connected to the larger distributed simulation exercise and participates as a standard HLA or DIS player. Likewise, in the live aircraft, the network between the Mission Processor and airborne interface side of the datalink also carries HLA or DIS traffic. Due to the ready access to this standard protocol network within the live flight deck, it is possible to easily embed additional training functions, such as embedded Semi-Automated Forces (SAF) or Plan View Display for instructor/safety pilot use, into the live trainer.

Where the datalink connects to the live aircraft's airborne DIS/HLA network, however, the data is bridged through a protocol translator application that converts the standard distributed simulation data into a set of over-the-air packets that is tailored for the specific datalink and waveform selected.

Depending on the features and/or constraints of the installed datalink, these protocol translator applications perform different functions. For example, one application was implemented to pair with a time-sliced datalink with a fixed packet size limit, and provided message prioritization and rate limiting, and packet fragmentation and reassembly. By contrast, a different application was built to pair with a datalink without any intrinsic provision in its protocol for reliable or near-reliable delivery. This application provided a buffer for storage and retransmission of a designated subset of messages which demanded reliable delivery.

At the ground side of the datalink, a matching instance of the same protocol translator application bridges the tailored datalink protocol back to a standard protocol used in distributed simulation. This is then distributed over the exercise LAN or WAN, and is sometimes converted to other distributed simulation protocols using a broker application. This distributed simulation networking is performed using methods well known within the field.

TESTING

Bench Testing

The laboratory environment consisted of an identical set of display hardware to the flight test aircraft, with accommodations for simulation of the sensors normally present on the aircraft. A simulated GPS/INS was created to feed the MFD application position and attitude information. Also, the training simulation environment was setup to connect to the MFD over the ARINC 661 protocol using the same Internet Protocol (IP) ports and addresses as equipment on the aircraft. In this way the laboratory environment was identical to the aircraft, and no changes to software were required when moving between locations.

Bench testing consisted of engineering verification of ARINC 661 pages displayed on the MFD. Functional unit tests, including connectivity, configuration, and filtering tests, were conducted utilizing the simulation environment in the lab. The MFD pages were exercised throughout aircraft operating and performance limits, as well loading of the training functionality with LVC entities and data.

The tests verified that the user can enter and exit the training mode and that the display reversions occurred properly during failure situations. It determined that the ADAHRS interface SE was correctly parsing the simulated RS-422 data stream to drive the standby instruments. The tests also verified that the training functionality of the various pages reacted to control inputs. The stores page displayed the correct munitions and allowed munitions to be selected and fired. The HSI page presented the waypoints and navaids for the preplanned route and the track files for other networked participants. The radar modes function correctly, allowing the radar page to present targeting information to the pilot that can be used to lock constructive participants in the distributed training exercise.

Flight Test Aircraft Integration

As the existing aircraft instruments were removed to provide room for the research MFDs, an additional dissimilar standby instrument system is installed in the experimental cockpit, and has no data connection in common with the experimental displays. Power for this standby system is common only at the aircraft bus level, and an additional internal battery provides continued operation in the case of loss of bus power.

A simulated F-18 control grip and throttle grip was fitted in place of the aircraft's original stick grip and throttle grip, through engineered adapters that mounted the grips to the aircraft's control column and throttle arm in place of the aircraft's original equipment. As the flight test aircraft is a single-engine aircraft, the left and right throttle grip of the F-18 controls were mounted together to a single bracket and do not move independently. Existing aircraft switches on the stick and throttle were relocated or assigned to equivalent HOTAS function switches on the training grips. The training flight deck is show in Figure 2.

Figure 2. Trainer cockpit with MFDs Installed

A debug connector was installed on the main instrument panel that allows diagnostic connection to the aircraft systems while on the ground. Connections available include serial debug ports on the MFDs, connection to the training system Ethernet network, and strapping connections required to enable dataload. An interface cable can be attached when the aircraft is on the ground, and when the aircraft is airborne, the connector is covered by a dust cap. In this configuration, the strapping pins are open, and dataload is inhibited.

After completing installation of the hardware into the research aircraft an initial flight test was conducted, attempting to provide engineering verification of basic avionics functionality and basic training functionality. The test verified that the pilot was able enter and exit the training mode and that the display reversions occurred properly during failure situations. It verified that the training functionality in several of the pages reacted to control inputs. The stores page displayed the correct munitions and allowed munitions to be selected and fired. The HSI page presented the waypoints and navaids for the preplanned route. Since there were no additional participants, no track files were displayed and the test could not verify that targeting capabilities were functioning. The pilot was able to enter the radar page and determine that it was running.

Due to several circumstances, the ADAHRS did not initialize correctly and could not attain GPS lock. While the test indicated that the RS-422 data was being parsed correctly, the lack of GPS lock meant the full functionality of the positional displays could not be verified.

NEXT STEPS

Flight Tests

Once further verification of the basic avionics and training functionality is complete, flight tests will be extended to incorporate LVC entities. Phase 1 will consist of a single live aircraft with virtual wingmen against a combination of virtual and constructive aggressor forces in air-to-air intercept scenarios, as well as constructive targets and additional participants in air-to-ground close air support scenarios. Phase 2 will add additional virtual and constructive entities as well as an additional live wingman, again in both air-to-air intercept and air-to-ground close air support scenarios. The potential third phase will examine incorporating the second live aircraft as a red force in air-to-air engagement scenarios, pending a full risk assessment. The flight test will be conducted under internal funding sources. Once the phased assessment of the LVC capabilities is complete, the architecture will be utilized to examine the human factors implications of pilots participating in these distributed training exercises.

Future Functionality

The first area of functionality due for future integration is a more complete Electronic Warfare (EW) and Radar Warning Receiver (RWR) system. The current system lacks any meaningful RWR capabilities. Integration of an existing RWR simulator is planned to address this gap. Existing VDN emitter data will be used as an input for this model.

With a longer outlook, the modular, "plug-in" nature of the simulation environment itself supports the concept of selecting training functions for a specific student based on a "menu" of available simulation capabilities. As an example, different radar simulators could be integrated, such that a pilot training for transition to a naval aviation interceptor role might use the Virtual Mission Training System (VMTS) baseline radar, which is very similar to an F-18 radar, whereas a pilot training for transition to the F-35 might train with a radar simulator representative of the capabilities of that particular aircraft. This possibility of tailoring the flight deck on a per-curriculum basis while using a baseline pool of avionics and training aircraft represents a dramatic change from traditional flight training.

Additionally, sensors and missions widely removed from the currently implemented training needs may also be readily inserted into the flight deck. Examples include forward looking infrared (FLIR) or Synthetic Aperture Radar (SAR) sensor simulation, which can be provided as a merged video input to the MFDs, or substitution of different model SEs, in order to provide training for dramatically different roles such as Anti-Submarine Warfare or Electronic Warfare.

CONCLUSIONS

This paper presented a new architecture for injecting virtual and constructive entities into live aircraft. For platforms where the Operational Flight Program (OFP) development is not complete, the proposed architecture could be easily accommodated. Retrofitting currently fielded platforms to accommodate the architecture presents more of a problem, as it would require the cooperation of the aircraft manufacturer that originally developed the OFP.

The architecture provides the capability to train with systems and sensors not physically present on the platform. It is capable of utilizing pilot inputs as well as participant state data and interactions sent over a datalink, enabling embedded distributed training on live platforms in LVC exercises. The system has been installed in an experimental aircraft, with flight tests planned for the next several months. The architecture was developed under internal funds, and the system verification flight tests will also be conducted using internal funding resources.

REFERENCES

Ausink, J. A., Taylor, W.W., Bigelow, J. H., & Brancato, K. (2011). Investment Strategies for Improving Fifth-Generation Fighter Training. RAND Corporation, Santa Monica, CA. Retrieved from http://www.rand.org/pubs/technical_reports/TR871.

FAAC, Inc. ZAP Missile Launch Envelope. Retrieved from http://www.faac.com/zap.html

Sidor, G. (2012, March 28). CAF LVC Pilot Project Overview. (Briefing)

U.S. Air Force, (1997, October 8). Operational Requirements Document for Distributed Mission Training. Washington, D.C., CAF 009-93-1-A.

U.S. Air Force, (2003, October 20). USAF Distributed Mission Operations CONOPS White Paper.