Rapid Development of a Multi-Aircraft Aviation System Simulation

Paul Lawrence Hamilton
ORION International Technologies, Inc.,
2000 North 13th Street
Arlington, VA 22202

Robert Brittain
ORION International Technologies, Inc.,
2210 Buena Vista Drive SE, Suite 211
Albuquerque, NM 87106

John B. McKinley
University Research Foundation
6411 Ivy Lane, Suite 110
Greenbelt, MD 20770

Abstract: *In one concept proposed under the NASA Small Aircraft Transportation System program, pilots would self-separate by observing other aircraft near an airport using ADS-B and a moving-map display. An Airport Management Module (AMM) would sequence this traffic. The question was, however, whether the concept was practical and whether an automated Cockpit Associate (CA) could relieve pilot workload and improve performance. Sandia National Laboratories' modeling and simulation framework, Umbra, licensed by ORION, was used to federate a number of instances of a commercial flight simulator, AMM, and CA into a multi-aircraft simulation. Umbra facilitated the capture and logging of both key events and the flight technical error of all aircraft. Pilots reported that self-separation was easy to perform and that the CA provided the expected benefits. This simulation played a key role in the development of a public flight demonstration, which showed the value of Umbra in prototyping complex systems.*

Keywords: Simulation platform, human-in-the-loop, aviation, flight simulation, systems simulation

1 Problem and Significance

Instrument Flight Rules (IFR) capacity is severely limited at non-towered airports outside of radar coverage. Once an aircraft drops below the radar horizon of the nearest facility, the airspace near the airport is locked by air traffic control (ATC) procedures until ATC receives word that the aircraft has landed. No additional departures or arrivals can take place during this interval. Consequently, these airports can experience significant flight delays. The expected introduction of very light jets, such as the Eclipse, combined with the development of improved GPS-based instrument approaches, has the potential to significantly increase operations at non-towered airports. If these operations are located on the fringes of major terminal areas (as is likely, since travel demand is greatest in major metropolitan regions), these IFR delays can affect and be affected by operations at the active primary airport. One proposal for overcoming these delays is the NASA Small Aircraft Transportation System (SATS) [1] Higher Volume Operations concept in which reception and graphic display of GPS-derived Automatic Dependant Surveillance – Broadcast (ADS-B) position data allows pilots to self-separate much as they would in visual conditions. However, in order to provide a more uniform flow of aircraft and to simplify the maintenance of separation, it is desirable to sequence them. NASA proposed and developed an Airport Management Module (AMM) to provide sequencing information to these aircraft within a special type of airspace surrounding the airport called a Self Controlled Area (SCA).

Working as part of the Maryland Mid-Atlantic SATS Laboratory (MMSL), a partnership of 29 private, public, and university institutions, the University Research Foundation (MMSL lead organization) and Applied Systems Intelligence, Inc. (ASI), proposed and developed a Cockpit Associate (CA) module to reduce pilot workload under these circumstances and in general. The CA is designed to function as a copilot and infers the pilot's needs by continuously monitoring the airspace for traffic, obstructions, airspace boundaries, airspace rules, and weather. The CA displays information, makes recommendations, monitors the aircraft's conformance to flight plan, and even mediates data link communications with ATC and the AMM. The CA is connected to the other data sources on board the aircraft via URF's Advanced Data Fusion Processor (ADFB).

The following questions had to be answered in order to test the validity of the Self Controlled Area concept:

- Is self-separation by pilots of AMM-sequenced aircraft in an SCA feasible?
- Does the Cockpit Associate provide a net workload improvement to the pilot without introducing

distractions that impair performance, particularly with regard to flight technical error, which is the accuracy to which required flight operations can be performed?

Given the cost of multi-aircraft flight testing, a human-in-the-loop simulation (HITS) test bed for evaluating the feasibility of these and other distributed air/ground IFR procedures and technologies, such as the AMM and CA, was required. Furthermore, rather than simply emulating the operation of these modules, the actual systems needed to be embedded in a simulation environment that would also contain multiple pilot stations, an ATC emulation, and some way of logging all flight data. Such simulation laboratories have been built and used by institutions such as NASA Langley, NASA Ames, and MITRE, but they are quite expensive. The Umbra simulation framework, [2, 3] developed by Sandia National Laboratories and licensed by ORION, offered a less costly alternative. Umbra provides a means to incorporate existing simulations and actual prototype software and hardware into a single complex simulation. In this situation, it would provide a means of logging both flight data and simulation events so that a given simulation experiment could be run repeatedly and the results analyzed. Furthermore, Umbra is well suited to rapid prototyping and possesses an extensive library of flight, UAV, and cockpit simulations that can easily be incorporated into a model. Finally, URF welcomed the ability to place its aircraft's automated flight data recorder in the Umbra simulator and play back a flight experiment for visualization and analysis [4].

2 Methods

The Umbra environment was used for both transmission of data and data logging. The first component of the simulation is a Pilot Station running Microsoft Flight Simulator, a commercial interface program for Flight Simulator, and an Umbra server. This machine is equipped with a yoke, pedals, and console as well as an Umbra server to connect with the Umbra system. This machine also sends out a Neutral Message Language (NML) data stream to the Cockpit Associate (and thence to and from the AMM), mimicking an actual installation. An Instructor Station with an Umbra client receives messages from Flight Simulator and sends NML messages to the Airport Management Module, Cockpit Associate, and other instances of Flight Simulator running in the simulation environment. AMM and CA are hosted on this machine. Three Target Stations also function as pilot stations, each of which runs a Flight Simulator instance and an Umbra server. This system is illustrated in Figure 1.

Post-processing software was developed to read the dynamic variables logged by Umbra (latitude, longitude, ground speed, altitude, and simulation time) and processes them for analysis and display of flight technical performance. This environment distributes the computation-intensive part of the simulation to individual workstations, which are coupled to one another only by a relatively low bandwidth NML stream. This allows the pilot station to perform a high-frame rate cockpit simulation, greatly enhancing realism and fidelity. It also provides modularity – traffic can be generated either from pilot stations, recordings from an actual flight, or a multiple-aircraft autonomous drone program (traffic generator).

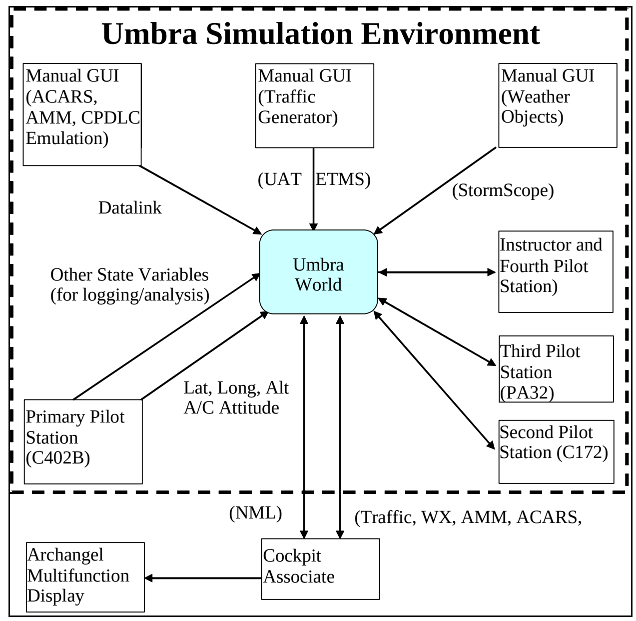

Figure 1 -- Flight Simulation Laboratory

2.1 Pilot Station

Several alternatives for Pilot Station software were considered for the simulation environment.

- Microsoft Flight Simulator
- Flight Gear open source simulation product
- FAA-approved PCATDs such as Elite's On Top
- X-Plane by Laminar Systems

Approved Personal Computer-based Aircraft Training Devices (PCATD) were eliminated because of proprietary code and interfaces, along with higher prices. Flight Gear, developed from NASA's LaRCsim, offered a good deal of flexibility in setting up a research environment; however, Microsoft Flight Simulator was chosen for the pilot station for several reasons. Although the code itself is proprietary, there exists a third-party interface program, FSUIPC, which maps the variables in Flight Simulator to a fixed set of variables, making it possible to easily write programs that interact with Flight Simulator and are version independent [5]. Flight Simulator is a Microsoft consumer product, well documented and supported by numerous third-party application developers. This makes it easy to obtain commercial off-the-shelf enhancements to tailor the program to a particular application.

Although Microsoft has not pursued certification, Flight Simulator software provides a flight simulation environment comparable to that of the software used in FAA-certified PCATD. Most Microsoft Flight Simulator users "fly" with a simple potentiometer-based flight yoke running off the computer's game port. More serious users prefer a more sophisticated yoke that connects to the computer through the Universal Serial Bus (USB). In order to bring the pilot station close to PCATD standards, a USB-based yoke, console, and pedals with operating toe brakes were installed. This provided fairly realistic physical (not virtual) controls for pitch, roll, yaw, power, prop, mixture, landing gear, flaps, and pitch trim. In a follow-on effort, a custom interface for the X-Plane [6] simulation program was developed and implemented, built around a proprietary interface [7]. This offered a chance to test head-up displays (HUD) and highway-in-the-sky symbology [8].

2.2 Simulation Aircraft

In order to provide a representative range of flight characteristics, two Flight Simulator aircraft and two third-party aircraft were selected. The primary aircraft is the Cessna 402, with secondary aircraft (Cessna 172, Cessna 182RG, and Piper Saratoga) included primarily to provide dissimilar traffic performance. These aircraft were picked based upon several criteria:

- Provide a range of speeds representing the piston general aviation fleet
- Represent aircraft used by URF and MMSL team members
- Have realistic flight characteristics

These aircraft were flown on the simulator to ensure that they had reasonable flying characteristics and were fully compatible with all the add-ons installed in ORION's version of Flight Simulator. Instrument substitutions and other changes were made when appropriate in order to correct the fidelity of the simulation models. This set of aircraft is easily expandable. Microsoft and various third-party developers have created compatible aircraft models for many types, including light jets such as the Eclipse. If required, the cockpit instrumentation can be further modified. For example, third-party developers have created glass cockpit modules. Microsoft provides a detailed set of tutorials for modifying aircraft, instrument panels, and flight models. This documentation is sufficient to develop prototype instrumentation, which could consist of glass cockpit displays, highway-in-the sky presentations, and other concepts of interest to the SATS initiative.

2.3 Technologically Advanced Aircraft

Technologically Advanced Aircraft are defined as aircraft that pose significant workload and proficiency issues due to a high level of cockpit automation. This typically involves an approach-qualified GPS with coupled autopilot and moving map display. Additionally, it was determined that a moving map display was necessary for flying the required intercepts and holds in a nonradar environment. In particular, the GPS-T approach developed for the MMSL Easton Airport (June 15-16, 2004) and the NASA SATS 2005 Danville public demonstrations, had a 90-degree angle between the initial approach fixes and the final approach course. Unlike most "T" approaches, there was no holding pattern depicted at this junction. Also, given the responsibility of the pilot to maintain separation in the SCA environment, it seemed reasonable to assume that the autopilot would play a significant role in workload management.

Although Flight Simulator does include a GPS, this unit does not function like any of the real approach-qualified IFR GPS units in actual cockpits. Consequently, it can not be expected to provide a pilot with realistic workload. As a result, a Garmin GNS-530 simulation was incorporated at the primary pilot station. The GNS-530 is a state-of-the-art multi-sensor navigator that provides navigational information from either the GPS satellites or from a ground-based source (VHF Ominrange Receiver or a localizer/glideslope). In addition to integrated satellite and ground station navigation with moving map display, it also has provision for Traffic Alert/Collision Avoidance System (TCAS), weather (Stormscope or uplinked weather radar), and even incorporates a VHF radio communications transceiver.

Our simulation incorporates a ground training simulator provided by Garmin for pilot training. It incorporates the same engine as the actual GPS and faithfully simulates the full function set of the actual GNS-530. The Flight Line 530XP interface by Reality XP served to couple the Garmin simulator to Microsoft Flight Simulator. By installing the Garmin simulator, all of the operating characteristics and modes of a modern, autopilot-coupled, approach-certified GPS are modeled. The Reality XP interface replicates the interface of the Garmin, using a right or left mouse click to turn the knob right or left. Scrolling is managed by pressing and moving the small wheel on the top of the mouse. The unit was fully coupled into the autopilot to allow automatic flight over a predefined flight plan and coupled approaches, thus providing both the capabilities and the challenges of a Technologically Advanced Aircraft.

2.4 Instructor Station

The Instructor Station hosts an Umbra client, situation display map, and post-processing software. It can optionally host a Flight Simulator instance with an Umbra server so that one target can be flown from it. From the instructor station, the operating conditions for each simulation experiment are set. Data recording is performed automatically to facilitate post processing. As described earlier, use of a multiplayer session makes it possible for the instructor station display to show the Pilot Station aircraft and other targets.

2.5 Target Stations

As delivered to URF, the simulation system was configured for one Pilot Station and up to three Target Stations. The system can easily be reconfigured for additional Pilot and/or Target Stations. As described above, each target station has an Umbra server, Flight Simulator instance, and FSNavigator. These stations can be set up on dedicated machines or on the Instructor Station. Pedals and console are not required for these stations, although a yoke/joystick, if available, makes takeoffs and landings easier. Figure 2 shows a Pilot Station as demonstrated at NASA SATS 2005.

Figure 2 -- Flight Simulation Station

2.6 Output

In order to provide a flexible means for obtaining appropriate output from a test run, there are two scripts for writing output files. The first script serves to capture simulation events and to provide a replay capability. The second is formatted to provide speed, position, direction, and altitude for input to a companion analysis spreadsheet. The use of scripting allows a user to reconfigure data collection to suit a particular experiment. Because these data collection scripts do not run on the Pilot Station machine, they impose no CPU demands and do not affect simulation speed, smoothness, or fidelity.

3 Results

The result of investigations performed with the simulation bore out both of the hypotheses that motivated the creation of the simulation.

- Self-separation and AMM sequencing proved an easy task for all of the pilots
- The Cockpit Associate provided useful assistance without any sacrifice in flight technical error.

The simulation was flown extensively by ORION and URF pilots, who reported that the multi-airplane simulation was a realistic portrayal of instrument flight operations. During initial tests, it proved to be a good systems test facility for both the Cockpit Associate and the AMM. It was also flown by members of the general public and proved robust and reliable. Pilots reported no difficulty in self-separating their aircraft, even with three or more aircraft sequenced into the SCA.

The synoptic collection of both flight technical data and event records from a simulation run proved easy and effective. Figure 3 shows an Instrument Landing System (ILS) approach from a map view and Figure 4 shows the corresponding profile view. Note the factor of four magnification of the Y-axis (cross-track position) in order to make deviations visible. It was also easy to perform statistical analysis upon this flight technical error data to determine any change due to the CA.

Pilots reported a decrease in workload when using the CA and indicated that adding word recognition functionality into the system would have further benefits. One particular area of interest was whether pilot interaction with the Cockpit Associate would degrade flight performance during the critical phase of final approach. Messages and charts presented during this phase may include path conformance alerts, altitude conformance alerts, and spacing alerts with lead aircraft, runway traffic display, and checklist items. Experiments with four pilots over 16 approaches produced cross-track and altitude deviation results for flights with and without the Cockpit Associate that were statistically similar, concluding that the Cockpit Associate had no degradation on pilot performance, as shown in Tables 1 and 2.

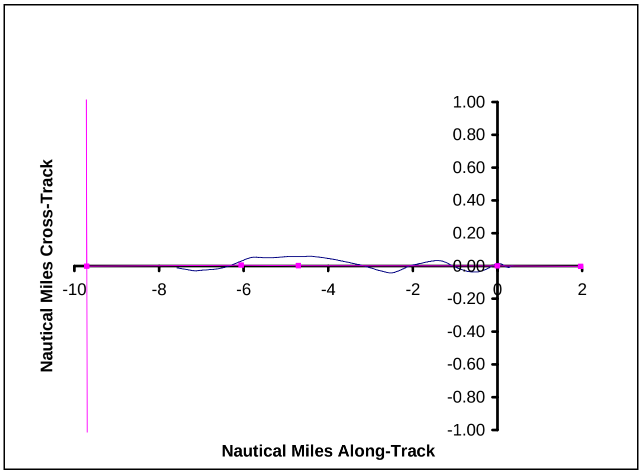

Figure 3 -- ILS Approach - Map View

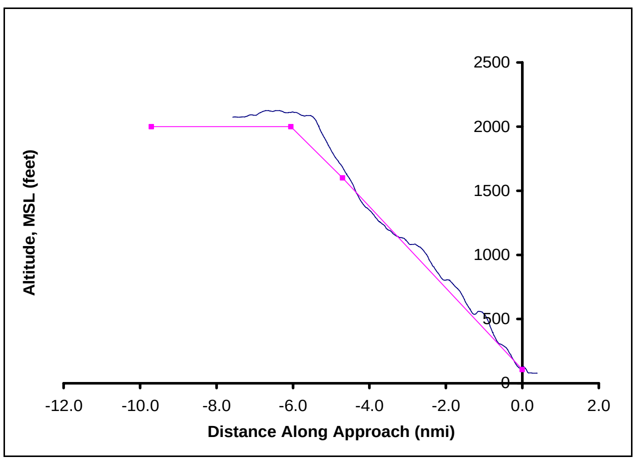

Figure 4 -- ILS Approach - Profile View

Table 1 -- With Cockpit Associate

Cross-track	
Standard Deviation	0.04 nm
Mean Deviation	0.01 nm
p = 95% Confidence	0.09 nm

Altitude	
Standard Deviation	48 feet
Mean Deviation	106 feet
p = 95% Confidence	285 feet

One unexpected benefit of the system was that it also proved valuable for exploring the functioning of the Advanced Data Fusion Processor (ADFP), Cockpit Associate and Airport Management Module in a realistic environment. Several issues were noticed with these systems that had not been evident during earlier tests without the Umbra simulation system.

Table 2 – Without Cockpit Associate

Cross-track	
Standard Deviation	0.07 nm
Mean Deviation	-0.06 nm
p = 95% Confidence	0.23 nm

Altitude	
Standard Deviation	105 feet
Mean Deviation	52 feet
p = 95% Confidence	287 feet

The ADFP posed major validation challenges. In addition to expected code changes resulting from simulation and flight testing, the ADFP underwent several major version updates to accommodate the AMM NASA updates, the new avionics systems, and the formatting requirements from the CA. The Umbra test environment directly facilitated discovery of software errors in the ADFP code resulting from undocumented message types for some interface avionics. This modeling and testing was of critical importance in getting the airborne systems to work properly together.

A year after this work was completed, as part of the NASA SATS 2005 event, a full-scale public flight demonstration was performed at Danville (VA) Regional Airport using an AMM and multiple aircraft, including a URF “Flying Laboratory” test airplane equipped with ADS-B, data link, moving map, Advanced Data Fusion Processor, and a Cockpit Associate. The simulation project described here was an important step in making this demonstration possible.

4 Discussion and Conclusions

The use of a simulation framework for assembling a complex simulation model from a number of pre-existing sub-models was shown to be a feasible method for rapid system prototyping. Providing a standardized “wrapper,” these legacy programs can be quickly and reliably joined together in a highly modularized, robust model. Such models make it possible to test much of the functionality of a proposed system as well as to develop and test procedures for using the system. Although it is possible to link multiple models without the use of a simulation framework, the ability to easily collect synoptic data from the system is a significant benefit.

The Umbra simulation and integration platform has been shown to be suitable for the rapid development and deployment of a multi-aircraft human-in-the-loop aviation systems simulation. Although not originally intended for real-time simulation, Umbra's versatility and ability to transmit data with very little overhead makes it a good candidate for linking flight simulator instances, experimental hardware and software, and logging and analysis software. It is to be expected that it can be similarly applied in other simulations.

5 Acknowledgments

The authors wish to acknowledge the contributions of Thomas Pierce, URF flight simulation laboratory engineer, for overseeing the simulation setup and operation; Dr. M. C. Ertem of URF for technical design assistance; and URF simulation flight test pilots Dr. Norris J. Krone, Jr., Eric Heidhausen, and James Cramer among others. This work was funded by NASA/NCAM under contract NCA-1-02001.

In addition, the authors wish to acknowledge the developers from the Sandia National Laboratories' Intelligent Systems and Robotics Center. ORION staff supported the development of Umbra and now has licensed the software from Sandia for commercial purposes. This publication incorporates technology licensed from Sandia National Laboratories under Sandia License Number 02-C01059.

6 References

[1] Holmes, B., "Small Aircraft Transportation System A Vision for 21st Century Transportation Alternatives," White Paper, NASA Langley Research Center, May 2002.

[2] Gottlieb, E. J, R. Harrigan, M. McDonald, F. Oppel., and P. Xavier, "The Umbra Simulation Framework," *SAND2001-1533, Sandia National Laboratories*, Albuquerque, NM, 2001.

[3] Gottlieb, E. J., M. J. McDonald, F. J. Oppel, J. B. Rigdon, and P. G. Xavier, "The Umbra Simulation Framework as Applied to Building HLA Federates," *Informs*, 2002.

[4] McKinley, J. B., E. Heidhau, and J. A. Cramer, "Flight Testing of an Airborne Synthetic Vision System with Highway-In-The-Sky on a Head-Up Display," 24th Digital Avionics Systems Conference, Washington D.C., October 30, 2005.

[5] Dowson, P., "FSUIPC SDK. 26th Release, http://www.schiratti.com/dowson.html.

[6] Myers, A., "X-Plane," Laminar Research, http://www.x-plane.com.

[7] Supnik, B. S. and S. Barbour, *"X-Plane Plugin Software Development Kit,"* http://www.xsquawkbox.net.

[8] Ertem, M. C., "An Airborne Synthetic Vision System with HITS Symbology Using X-Plane for a Heads-Up Display," 24th Digital Avionics Systems Conference, Washington D.C., October 30, 2005.