Stone Axes and Warhammers:
A Decade of Distributed Simulation in Aviation Research

Frank M. Sogandares
Center for Advanced Aviation System Development (CAASD)
7515 Colshire Drive, McLean VA 22102-7508
Mailstop N420
(703)883-6919, fsoganda@mitre.org

Abstract

Ten years ago, MITRE/CAASD built a realtime, Human-In-The-Loop (HITL) research laboratory. The focus of this lab is integration and human factors research for the air traffic control and aviation communities. The last ten years have been illuminating in terms of the evolution of laboratory capabilities, infrastructure, and corporate culture. This paper will describe the laboratory environment, its history and vision, and will also provide some examples of how distributed simulation technology has been applied, and continues to evolve, in a real-world HITL simulation environment serving a broad range of research needs.

Background

The MITRE Corporation's Center for Advanced Aviation Systems Development (CAASD) is the Federal Aviation Administration's (FAA) federally funded research and development center (FFRDC). CAASD is chartered to work in the public interest, and in addition to direct FAA support, also supports the international civil aviation community with research and modernization activities. CAASD has several laboratory facilities which address a variety of business needs. This paper describes the CAASD Air Traffic Management (ATM) laboratory, which is a realtime, distributed HITL simulation system for exploration of air traffic control concepts.

The United States National Airspace System (NAS) is comprised of people, procedures, data and automation which work together to perform Air Traffic Control (ATC) activities. The people making up NAS include air traffic controllers, automation specialists, and a variety of other support personnel, totaling approximately 25,000. Research in the ATM laboratory focuses on issues related to air traffic controller or pilot roles and responsibilities, or experimentation and evaluation with concepts or prototypes.

Air traffic controllers provide hazard monitoring and critical separation assurance services to aircraft. Controllers work in distinct domains, each of which has its own unique issues and requirements. The ATC domains are: tower, terminal, enroute, oceanic and flow control.

The tower domain generally involves the operation of the airspace directly surrounding an airport (typically a 5 mile radius, up to 3,000 ft.), and movement of aircraft on the airport surface.

Surface movement on the airport is very controlled, with aircraft movement thoroughly coordinated between the tower and the pilot. Physically, this domain includes the airport structures, the Air Traffic Control Tower (ATCT) and other facilities, aircraft on the runway, taxiway and apron, and pilots and controllers involved in the activities of moving these aircraft. The NAS currently includes approximately 475 tower facilities.

Following departure, aircraft transition to the airspace of the terminal ATC domain. Terminal airspace typically surrounds one or more airports and reaches altitudes of approximately 10,000 feet. Aircraft in the terminal area may maneuver rapidly as they prepare to transition inbound to the tower's airspace, or outbound to cruise speed and altitude. In the terminal domain, air traffic controllers perform most of the arrival traffic sequencing and spacing as aircraft approach the airport for landing. Terminal area controllers work at a Terminal Radar Area Control Facility (TRACON), which may incorporate more than one terminal area. The NAS currently includes approximately 200 TRACON facilities.

Aircraft which exit terminal airspace to fly at higher altitudes will transition to airspace within the en route domain. Aircraft in enroute airspace typically level off at altitudes between 21,000 and 33,000 feet, fly at high speed, and maneuver less frequently or rapidly in comparison to terminal area flight. The airspace in the continental United States (CONUS) is divided into 20

Proceedings of the 16th Workshop on Parallel and Distributed Simulation (PADS'02)

interlocking volumes of 3-dimensional airspace, each controlled by an Air Route Traffic Control Center (ARTCC); each ARTCC is responsible for aircraft within its airspace. The airspace inside each center is further subdivided into interlocking volumes of airspace known as sectors. Each sector is staffed by air traffic controllers, responsible for separation assurance and safe handling of the aircraft. In en route airspace controllers use radar to facilitate separation assurance.

Over the ocean, aircraft fly in the oceanic ATC domain. Aircraft in this environment also fly at high altitudes and high speed. Oceanic airspace differs from en route in that radar surveillance is not normally available. This requires procedural methods for providing separation assurance to aircraft. Pilots check in at predefined locations; controllers monitor aircraft progress and use rules governing separation to determine if separation assurance criteria are met. The New York (ZNY) and Oakland (ZOA) ARTCCs are responsible for the largest portions oceanic airspace.

The Flow Management domain has the CONUS-wide view of NAS and manages the overall flow of traffic in the NAS. Problems in one airspace can result in a disruptive constraint across the CONUS at an apparently unrelated location. National flow management is performed at the Air Traffic System Command Center (ATSCC). The ATSCC works in a strategic manner, trying to predict and manage flow problems before they become serious. To address problems that occur in a given ARTCC, the ATSCC works with Traffic Management personnel on site within the ARTCC. ARTCC traffic managers also elevate serious local flow problems to the ATSCC for coordination at the national level.

The flight deck domain includes the cockpit environment and avionics. Pilots and co-pilots work on the flight deck, communicating with air traffic controllers, and performing the activities required to safely pilot their aircraft.

Laboratory Requirements

The laboratory environment has been designed to meet numerous business requirements. Among the more noteworthy are the following.

Interactivity

The laboratory must perform real-time simulation, allowing subjects to interact with a simulation, and allowing simulation data to be stored and retrieved for later study.

ATC and flight deck activities involve humans, working in realtime with the assistance of automation and other humans. Supporting rational inferences or observations about these activities requires that data be captured from simulations.

Flexibility

The simulation environment must support integration research, concept and prototype development and software reuse.

Among other things, this requires that the simulation environment be highly configurable. Experimenters will need the flexibility to add or enhance functionality for simulations. In some cases, users will need to focus on one domain. In other cases experimenters will need to create large scale simulations which span multiple domains.

User Interface Emulation

The laboratory user interfaces and functionalities must present domain experts with a reasonable representation of their environment.

The guideline for reasonableness is subjective. An acceptable implementation is considered 'not-distracting' by a domain expert. The types of research questions being asked will help define reasonableness; in some cases, the laboratory will not meet the fidelity requirements of an experiment.

Message Level Interfaces

The laboratory must implement the message-level interfaces required for the integration of prototypes.

Prototypes which work in operational environment use well defined interfaces. Integrating these prototypes into the lab will require implementation of these interfaces. Additionally, prototypes which use the lab and these interfaces as their development platform will be easier to transition to an operational environment.

Ease of use

Simulations and simulation clients must be 'easy' to execute, configure, pause, and resume. Simulations must startup and shutdown cleanly.

This is an extremely important requirement. A simple to use environment allows developers, analysts and experimenters to execute simulations without the assistance of programmers.

Clearly the breadth of needs suggests that the simulation environment use a distributed, as opposed to monolithic, architecture. A distributed architecture would better meet the flexibility requirement, allowing different 'standalone' simulations to be more easily brought together as one. Furthermore, simulations involving many aircraft are computationally expensive to run; a distributed architecture allows the lab to better utilize computing resources, simulating more aircraft. However, the ease-of-use requirement is more difficult to meet with a distributed architecture.

Given this choice, the following important architecture related requirements emerge.

Coordination. Simulations must execute in coordinated, discrete realtime. The simulation environment must perform simulation wide time-management.

Multiple Simultaneous Execution. The laboratory must support the execution of multiple, independent, simultaneous simulations. Simulations must not interfere with one another. As a practical matter, during the development and test phases, and often for demonstrations and evaluations, numerous simulations will be required to run simultaneously.

Data Distribution. Client applications must obtain state information from objects of interest in the simulation at runtime. The objects of interest in our aviation simulations are aircraft. In addition to a distributed aircraft database, client applications will also need to perform normal interprocess communications and handle networking exceptions consistently.

Hardware Infrastructure

Laboratory hardware consists of computers, displays and various aviation-specific hardware which support backend computing and control, and display processing for simulation. Currently, the lab hosts numerous Solaris-based machines ranging from Ultra 1's to Ultra 60s and Blade 1000s, several linux machines, starting at 400MHz Pentium II machines, and several SGI machines. Additionally, the lab heavily uses the CAASD network which provides out-of-lab computing resources, storage, backups, and other support.

Aviation specific hardware is composed primarily of ATC consoles and an audio system used to simulate the radio environment. ATC consoles are medium to high fidelity mock-ups of the 'real' capabilities used by enroute and terminal area air traffic controllers. These consoles include Sony 2048x2048 pixel displays, measuring ~28" diagonally and include keyboards and trackballs similar to those in field use. The laboratory has eight displays, each driven by a Sun Ultra 5 with a TechSource display card.

Voice communication is an important factor in ATC. In the lab, radio frequency simulation allows human participants to interact with each other in a realistic manner, and allows experimenters to observe and perform data collection. Simplified enroute simulations may require one frequency, while the flight deck can require more than 10 frequencies, including navaids; different domains have different complexity requirements. From a radio frequency perspective, integrating these domains into a single simulation can be challenging.

Other hardware infrastructure is closely tied to specific capabilities, and will be discussed with those capabilities.

Software Infrastructure

MITRE/CAASD has developed simulation software infrastructure that facilitates the construction of distributed simulations. All APIs were implemented in the C programming language under various research and development programs. These infrastructure components simplify interprocess communication, the use of a distributed database, and the tasks of managing complex, highly distributed simulations.

Interprocess Communications

Comm Service is an API providing interprocess communication capability to the programmer. Comm Service performs non-threaded, blocking I/O, using a simple application-level protocol. Comm Service ensures that a message is complete before returning it to the user. It also handles signals, and allows the programmer to specify protocol.

Distributed Aircraft Database

Inter-Target Generator Protocol (ITP) is a distributed aircraft database. In ITP, aircraft are represented internally as a state vector of approximately 70 disparate attributes ranging from kinematic state to radio frequency and landing-gear position. ITP makes a clear distinction between 'producers', which update aircraft state data, and 'consumers' which reflect those updates. ITP has the following features:

Publish/subscribe mechanism. This feature allows client applications to subscribe to aircraft attributes according to subscription options and client-specified location criteria or unique-id.

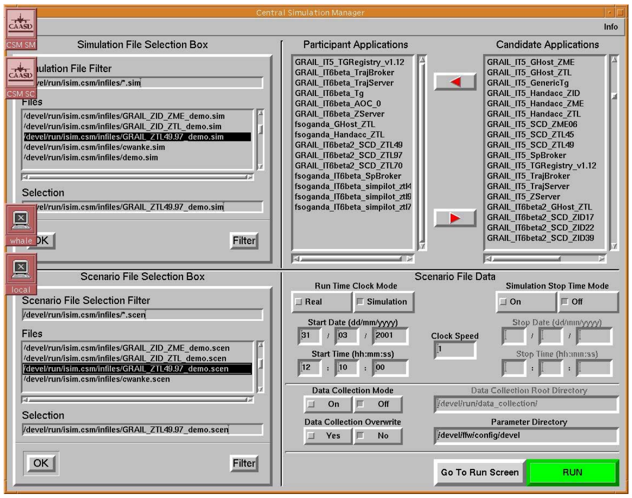
Attribute Selectivity. Client applications can tailor their request for aircraft attributes in order to reduce network bandwidth requirements.

Extensible Attribute Set. ITP allows applications to create simulation specific aircraft attributes, which can be updated with the canonical ITP attribute set.

Handoffs. ITP allows applications to assume or divest 'ownership' of an aircraft through a 'handoff'. Under such a handoff, any client may become the producer for an aircraft, provided the producer client divest ownership. ITP allows third-party applications to facilitate handoffs.

Simulation Control

The Central Simulation Manager (CSM) is a standalone X Windows application that runs on the Solaris platform. CSM allows users to easily execute distributed simulations. CSM also allows users to create new simulations, or modify existing ones by adding, removing or modifying participant applications, and facilitates the integration of different applications together into a single simulation. The CSM front panel is shown in Figure 1.



Simulation File Filter: vel/run/isim.csm/infiles/*.sim

Files:
/devel/run/isim.csm/infiles/GRAIL_ZID_ZME_demo.sim
/devel/run/isim.csm/infiles/GRAIL_ZID_ZTL_demo.sim
/devel/run/isim.csm/infiles/GRAIL_ZTL49.97_demo.sim
/devel/run/isim.csm/infiles/cwanke.sim
/devel/run/isim.csm/infiles/demo.sim

Selection: vel/run/isim.csm/infiles/GRAIL_ZTL49.97_demo.sim

Participant Applications:
GRAIL_IT5_TGRegistry_v1.12
GRAIL_IT6beta_TrajBroker
GRAIL_IT6beta_TrajServer
GRAIL_IT6beta_Tg
GRAIL_IT6beta_AOC_0
GRAIL_IT6beta_ZServer
fsoganda_GHost_ZTL
fsoganda_Handacc_ZTL
GRAIL_IT6beta2_SCD_ZTL49
GRAIL_IT6beta2_SCD_ZTL97
GRAIL_IT6beta2_SCD_ZTL70
fsoganda_IT6beta_SpBroker
fsoganda_IT6beta_simpilot_ztl4
fsoganda_IT6beta_simpilot_ztl9
fsoganda_IT6beta_simpilot_ztl7

Candidate Applications:
GRAIL_IT5_GHost_ZME
GRAIL_IT5_GHost_ZTL
GRAIL_IT5_GenericTg
GRAIL_IT5_Handacc_ZID
GRAIL_IT5_Handacc_ZME
GRAIL_IT5_Handacc_ZTL
GRAIL_IT5_SCD_ZME06
GRAIL_IT5_SCD_ZTL45
GRAIL_IT5_SCD_ZTL49
GRAIL_IT5_SpBroker
GRAIL_IT5_TGRegistry_v1.12
GRAIL_IT5_TrajBroker
GRAIL_IT5_TrajServer
GRAIL_IT5_ZServer
GRAIL_IT6beta2_GHost_ZTL
GRAIL_IT6beta2_SCD_ZID17
GRAIL_IT6beta2_SCD_ZID22
GRAIL_IT6beta2_SCD_ZID39

Scenario File Selection Filter: /devel/run/isim.csm/infiles/*.scen

Files:
/devel/run/isim.csm/infiles/GRAIL_ZID_ZME_demo.scen
/devel/run/isim.csm/infiles/GRAIL_ZID_ZTL_demo.scen
/devel/run/isim.csm/infiles/GRAIL_ZTL49.97_demo.scen
/devel/run/isim.csm/infiles/cwanke.scen

Selection: /devel/run/isim.csm/infiles/GRAIL_ZTL49.97_demo.scen

Start Date (dd/mm/yyyy): 31 / 03 / 2001
Start Time (hh:mm:ss): 12 : 10 : 00
Clock Speed: 1
Data Collection Root Directory: /devel/run/data_collection/
Parameter Directory: /devel/ffw/config/devel

Figure 1 - The CSM user interface

CSM performs activities for application health checking and ensures that participating applications start, pause, and shutdown cleanly. Once a user has started a simulation, CSM provides the user with a visual representation of application status. Applications that execute as CSM applications are required to follow the CSM protocol, implemented by CSM/IF; these applications can also run standalone.

Importantly, CSM also facilitates time synchronization between participating applications. As a general requirement, all laboratory computers use the Network Time Protocol (NTP) for internal clock synchronization; CSM. assumes all computers are similarly synchronized.

At simulation startup, CSM distributes timing information to all applications. This information includes the simulation start time as a wall time, a corresponding simulation time, and a clock rate. These times include a predetermined wait period, typically 15 seconds, designed to accommodate network latency.

Following the wait period, all applications use the system clock to update their simulation clock at the specified rate; applications cannot 'get behind' since simulation time is always based on to the system clock. This accommodates applications which become temporarily CPU bound, and allows them to remain synchronized.

Simulations can also run in 'fast time'. From the CSM user interface, the user can specify an integer scale factor which determines the clock rate. CSM also allows the clock rate to be changed at runtime.

Laboratory Capabilities

This section provides an overview of the lab capabilities at a domain-level, including support capabilities. Humans which interact in a simulation can play the following roles: aircraft pilot, air traffic controller, and analyst.

Low-fidelity interactive pilot

This position is typically referred to as the 'simpilot' or 'pseudopilot' and consists of a simple user interface which displays information about multiple aircraft on an area-wide basis. Simpilots maneuver aircraft in response to voice clearances from a controller over a simulated radio channel. The simpilot user interface allows simpilots to send speed, heading and altitude change commands to simulated aircraft objects, which model the desired behavior according to aircraft performance characteristics and other external factors such as weather. One simpilot can pilot many aircraft simultaneously. An example of the simpilot user interface is shown in Figure 2.

ACID	MODE	ALT	SPD	HDG	COMMENT
DAL1816	AUTO	010	250	031	
DAL585	AUTO	042	250	221	
DAL915	AUTO	191	290	222	CALL IN
N131XL	MANUAL	197	160	155	
N9120N	MANUAL	120	143 130	150	CALL IN
PDT3204	AUTO	157	180	270	
TRS67	AUTO	014	250	221	
VAR736	AUTO	160	203	204	

Command AC: N9120N

N9120N: <Hdg: 150.0>

Change Frequency: 108.049

Simulation Clock: 13:19:06

Figure 2 - Low fidelity interactive pilot user interface

Medium-fidelity interactive pilot

This position is known as the 'flight deck' or 'cockpit simulator' simulates one aircraft, and is far more complex than the simpilot positions. The cockpit simulator is a shell, fabricated at MITREs' in-house machine shop, which roughly models the Douglas DC9/MD80 flight deck. Two pilot positions and room behind for observers are provided. Each pilot position is fully operational, allowing either pilot to control the aircraft throughout all phases of flight. The cockpit includes software configurable touchscreen avionics, and can accommodate

COTS avionics. General aviation aircraft can be simulated with a Frasca 242, a COTS simulator.

Figure 3 - Cockpit and OTW Capabilities

Interactive Ground Movement Simulator

This capability is the 'groundsim', and is under development. Groundsim allows a pseudopilot to approach, land and taxi an aircraft in a simulated 3D environment. Pseudopilots can either interactively pilot an aircraft or follow a predefined path, and can also switch between aircraft and the tower view. Groundsim includes an interactive pseudopilot station, a 3D rendering environment, and a plan-view display of the airport with runways. Internally, groundsim constructs least-cost paths between runway and gates, and models type specific aircraft behaviors such as turning abilities; departures are not modeled.

Out-the-window

The out-the-window capability (OTW) allows pilots, controllers and analysts to visualize a simulated 3-D environment. This capability is integrated with domains and cockpit capabilities, allowing pilots and tower controllers to visualize their respective environments. OTW utilizes SGI Performer and Open GL, and the display processing is performed by two-processor SGI Onyx computer with an RE2 graphics subsystem. The output is projected onto a curved 21 foot wide by 5 foot high projection screen with 3 Electrohome projectors. The cockpit and OTW are shown in Figure 3.

Enroute simulation

The enroute simulation capability allows enroute sector controllers to perform standard ATC activities for simulated aircraft. Enroute controllers communicate with simpilots using a simulated radio channel, and the simpilots can maneuver according to controller directives. This enroute capability is the combination of two distinct capabilities: a display capability (DSR) and a processing capability (NAS/Host). The enroute controller uses a medium fidelity clone of the actual display system in the field, DSR, which presents air traffic in plan view, and allows the controller to pan, zoom, adjust aircraft datablocks, and input standard control messages. The processing capability emulates the real NAS software which processes radar and flight data, and otherwise supports controllers, allowing them to perform activities such as amending routes and assigning altitudes. This host emulation can be configured to run for any ARTCC, and support an arbitrary number of sectors.

Conflict Probe

In addition to enroute, the lab has integrated a CAASD prototype decision support tool, URET, into the laboratory. URET provides a conflict-probe capability by comparing aircraft trajectories and predicting violations of aircraft separation standards. URET alerts controllers to predicted violations and them to perform 'what-if' types of analyses to resolve the problem. and. Figure 4 shows the DSR clone and conflict probe capabilities together.

Recall that the flexibility requirement for the laboratory, specifies that these different capabilities should be integrable. From a simulation perspective, this means that a pilot in the cockpit can fly the aircraft through enroute airspace, interacting with ATC in an appropriate fashion, transition to terminal airspace under the control of a terminal controller, and land and taxi in a realistic rendering of the actual airport environment, while talking to the tower and ground controllers as appropriate. For some evaluations, a simple terminal simulation is sufficient. Others may require that terminal and groundsim run in concert. Others still may require that

Figure 4 - Enroute Sector Suite: Radar Controller Display and URET equipped Data Controller

enroute be incorporated. The distributed architecture of the laboratory makes these different types of simulations possible; experimental designers have the flexibility to build simulations of the appropriate scope.

Terminal Area Simulation

The Terminal Area Simulation Facility (TASF) consists of a terminal area specific target generator, controller displays, and simpilot displays. The controller display is a medium fidelity clone of the TRACON ARTS3E controller display, and displays simulated aircraft, and allows many controller functions such as pan, zoom, and transfer of control. The display can be configured to model any airport.

Examples of Utilization

This section provides concrete examples of bringing distributed simulation to bear on problems important to the aviation community. What follows is a representative cross-section of laboratory facilitated demonstrations, evaluations, and experiments.

Procedure Enhancement

Time-based metering (TBM) is an ATC concept with the goal of optimizing aircraft throughput. TBM requires aircraft to arrive at a specific location, or meter fix, at a specified time, or 'slot'. Aircraft in arrival streams are assigned slots based on their estimated time of arrival (ETA) at the meter fix, subject to constraints of the airspace and other factors. TBM has workload and complexity implications for the air traffic controller, is cognitively difficult, and is not fully supported with software decision tools.

The first Laboratory experiment was completed in 1992, and examined procedural issues that accompany TBM. How early slots are assigned by the arrival scheduling tools, has implications for the controller and the traffic. This experiment examined some of these implications, and required the integration of enroute and terminal area capabilities, a prototype flow management tool, TMA, built by NASA for the FAA, and a simpilot capability [7]. TMA performed slot assignments, those slot assignments were displayed to the controllers, and the enroute and terminal controllers issued clearances to the simulated aircraft pilots, over a simulated radio channel, in an effort to help aircraft meet their slot times.

System Integration

In 1995, the FAA tasked CAASD with performing a preliminary study of the effects of wind errors, on the Final Approach Spacing Tool (FAST) prototype. FAST provides advisories to controllers regarding sequencing and spacing for aircraft nearing final approach. The simulations integrated FAST into the laboratory infrastructure and included the terminal area and simpilot capabilities. Terminal air traffic controllers watched the terminal displays, and issued directives to pilots, who flew the appropriate course [8].

System Enhancement

All commercial aircraft seating 30 or more passengers are required to have the Traffic Alert and Collision Avoidance System (TCAS). TCAS is an additional defense against midair collisions and uses aircraft transponder reports to determine the closest point of approach (CPA), and time to CPA between proximal aircraft. TCAS uses altitude specific criteria for allowable CPA and time to CPA to determine if aircraft are predicted to violate proximity constraints. When these constraints are predicted to be out of conformance, TCAS coordinates an alert between both flight decks aurally and visually on their TCAS displays. The most critical alert is the resolution advisory (RA), which requires immediate corrective action, and provides the flight deck with climb or descend instructions for collision avoidance; pilots are required to follow RAs or any overriding ATC instructions. TCAS is unpopular with the controller community[10]. Controllers are not notified by TCAS when RAs are posted to the flight deck. From a controller perspective, spontaneous altitude changes are disruptive, and often the controller has better overall situational awareness and could provide better resolution. Typically, aircraft complying with RAs communicate with ATC only after initiating corrective action.

In the TCAS RA Downlink Experiment [9], the actual TCAS logic was integrated into the cockpit simulator, and RAs were effectively 'downlinked' and displayed to the controller. Notionally, controllers have better overall situational awareness, and could provide alternative, less disruptive solutions, given this notification; at the very least, providing this notification helps the controller have better situational awareness. This experiment examined several issues such as the operational impact of downlinking this information, the display characteristics at the ATC display, and the impact of non-standard pilot responses. The terminal area simulation was used with a simpilot, allowing active controllers to control the traffic and use the RA downlink capability. Objective and subjective data were collected.

System Enhancement

In response to resolution advisories, pilots often adjust their altitude more than required, and may fail to return to their assigned altitude once the RA has cleared; from an

ATC perspective this is undesirable. The TCAS Operations Working Group proposed modifications to the TCAS visual display and aural alerts, which would actively direct the pilot to reduce the rate of climb or descent if appropriate. The cockpit TCAS system was appropriately modified and used to evaluate proposed modifications, saving the FAA and industry considerable resources [5].

Development of New Procedure

For the airlines, significant fuel and time savings are available at high altitudes. During the initial segment of transoceanic flight, when an aircraft is heavily loaded with fuel, these altitudes are not attainable; as the aircraft burns fuel, however, more optimal altitudes become attainable. Transoceanic flights are assigned cruise altitudes which are generally initially acceptable, but may become constraining at later stages of the flight when the fuel weight is lower. Non-radar oceanic procedures do not allow an aircraft to spontaneously cross the altitude of another inside specified longitudinal separation bounds, and there is no way for ATC and the flight deck to safely coordinate such a climb.

The TCAS In-Trail Climb concept [3], used the TCAS display of proximal aircraft to facilitate these climbs in oceanic airspace. This concept was prototyped and evaluated in the lab with representatives from the airlines and ATC. The evaluation capability required the cockpit, OTW, and included simulated TCAS avionics, and a terminal controller station. The concept was approved as a test procedure for limited airlines, and has since been enhanced to use Automatic Dependent Surveillance-Broadcast (ADS-B) technology.

Airport Redesign

CAASD was tasked by the Mexican civil aviation authority to study the options for expanding the Mexico City Airport [6]. The geography around Mexico City is mountainous and requires extra considerations for safety and throughput. CAASD built a virtual prototype for each proposal. Using OTW, and the cockpit, operations were demonstrated at each proposed site. These simulations were valuable contributions to the final site selection.

System Enhancement

CAASD has developed the Transition Airspace Controller Tools (TACT) for the purpose of assisting air traffic controllers performing TBM [4]. TACT utilizes existing ATC displays, requiring only modest new symbology to assist controllers with the complexity and workload of TBM.

TACT was prototyped and evaluated in the laboratory using the enroute capability and low-fidelity simulated pilot interface. Active air traffic controllers from different ARTCCs were brought into the lab to control simulated traffic with the DSR clone. Controllers issued clearances to simpilots over a simulated radio channel as they performed TBM with and without TACT. Figure 5 illustrates some features of the evaluations.

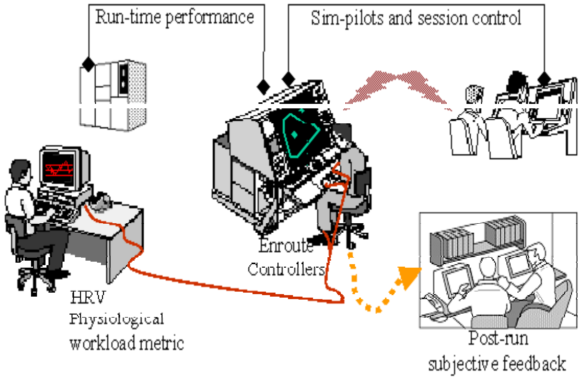


Figure 5 - Setup for TACT evaluations

Figure 6 illustrates the TACT Mileage-Distance-Marker (MDM) symbology. Data was collected during the evaluations and reduced offline, along with subjective controller feedback. Variability of participant heart-rate was also digitized during the evaluations and processed offline to measure indicators of cognitive workload.

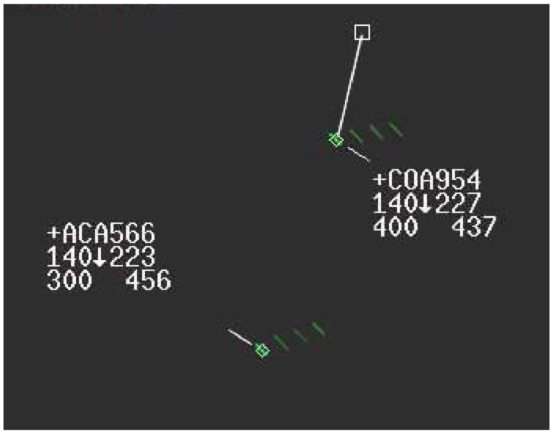


Figure 6 - Full data blocks. ACA566 and COA954 display the usual full datablock. COA954 also displays a TACT MDM.

System Enhancement

Adaptive Path Ghosting (APG) is a technique which allows controllers to optimally coordinate dependent streams of traffic arriving along separate paths [1] and was the approach taken to solve a runway utilization problem at Newark Airport. APG was integrated into the terminal area simulation, and the OTW simulation was configured to run as the Newark ATCT. Numerous evaluations were performed with Newark controllers, allowing them to control simulated aircraft spaced with APG. APG was implemented in the ARTS-IIIe automation at NY TRACON.

Development of Operational Concepts

Automatic Dependent Surveillance-Broadcast (ADS-B) is a new surveillance technology that the FAA and aviation stakeholders are testing for potential implementation in the NAS. In support of the Safe Flight 21 Ohio River Valley project, CAASD assists in the definition of the operation concept and performs about 6 lab evaluations per year. ADS-B/Cockpit Display of Traffic Information (CDTI) operational concepts are defined with representatives from the Cargo Airlines Association (CAA), commercial airlines, air traffic controller and pilot unions and the RTCA[2].

These operational concepts involve a variety of issues, and may include modifications to ATC procedures, automation, and cockpit avionics prior to implementation in the NAS. Using the cockpit, OTW, and terminal area simulations, concepts and procedures are first prototyped and tested in the lab, in order to identify and mitigate risk, as well as to determine concept feasibility. For feasible concepts, the lab facilitates further refinement of procedures and avionics display modifications, as well as training of the pilots and controllers using those procedures in a field demonstration. The field demonstration is performed within a testbed airport in the NAS using commercial and general aviation aircraft.

Conclusion

The choice to build the laboratory using a distributed architecture has allowed us to demonstrate a broad array of capability and flexibility to meet a variety of business requirements.

In the future, the lab will perform further domain integration, and improve its software infrastructure, possibly with the US Department of Defense High Level Architecture (HLA). Among the anticipated benefits are increased use of laboratory capabilities to examine integration issues, and more effective reuse practices.

Acknowledgements

The author gratefully acknowledges the assistance of his coworkers. The contents of this material reflect the views of the author. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty, guarantee, or promise, expressed or implied, concerning the accuracy of the views expressed herein.

References

[1] Barker, David R. "Adaptive Path Ghosting for Newark", WN96W0000021, The MITRE Corporation, McLean, VA, February 1996 (not approved for public release.)

[2] Bone, Randall, et al. "Safe Flight 21 Ohio River Valley: 2001 MITRE CAASD Simulations." In preparation.

[3] Cieplak, James J., ed. "Oceanic in-trail climb using TCAS II: program plan." MP 95W0000241, The MITRE Corporation, McLean, VA, September 01, 1995

[4] DeSenti, Christopher T. "Transition Airspace Controller Tools (TACT) visualization aids for radar controllers:concepts for cognitive assistance with flow control tasks in transition airspace." MTR 00W0000044, The MITRE Corporation, McLean, VA, July 2000.

[5] Domino, David A., Kaye, Ronald D., Bisaga, Gary J. "Preliminary human factors assessment of proposed changes to TCAS II resolution advisory cockpit displays and aural alerts." MTR 96W0000030, The MITRE Corporation, McLean, VA, May 1996 (not approved for public release.)

[6] Lisker-Melman, Dr. B., W. J. Swedish, F. A. Amodeo, M. S. Friedman, M. E. Heffron, M. K. Higgins, R. W. Kleinhans, Dr. S. V. Massimini, R. M. Tarakan, "Future Airport Development for Mexico City -- Studies of Technical Feasibility (El Futuro Aeroportuario de la Ciudad de México -- Estudios de Factibilidad Técnica)". MTR-00W090, The MITRE Corporation, McLean, VA, . October 2000.

[7] Small, Duane W., Benel, Russell A. "Timing the Issuance of Meter-Fix-Time Flow Restrictions: Report of the Results of the First I-Lab Experiment." MTR93W0000163, The MITRE Corporation, McLean, VA, December 1992 (not approved for public release)

[8] Gordon, Laurence M. "Results of a Preliminary Study to Show FAST Sensitivity to Errors in Wind Data." WN94W0000216, The MITRE Corporation, McLean, VA, January 1995 (not approved for public release.)

[9] Hoffman, Ronald B., Kaye, Ronald D., Sacher, Barbara H., Carlson, Laurel S. "TCAS II resolution advisory downlink evaluation report." MTR 95W0000080, The MITRE Corporation, McLean, VA, August 1995, (not approved for public release.)

[10] "Why Controllers Hate TCAS". System Notes. *IFR*, October 1994, p. 20-23.