Preprint, 1998 Fall Simulation Interoperability Workshop, Sept. 13-18, 1998, Orlando, FL.

A Distributed Simulation Facility to Support Human Factors Research in Advanced Air Transportation Technology

Keith Amonlirdviman
Todd C. Farley
R. John Hansman, Jr.
MIT International Center for Air Transportation
Massachusetts Institute of Technology
Room 33-113
77 Massachusetts Avenue
Cambridge, MA 02139
617-253-2271
keitha@mit.edu, farley@mit.edu, rjhans@mit.edu
http://web.mit.edu/aeroastro/www/labs/ICAT/

John F. Ladik
Dana Z. Sherer
TASC, Inc.
55 Walkers Brook Drive
Reading, MA 01867-3297
781-942-2000
jfladik@tasc.com, dzsherer@tasc.com

Keywords:
Human Factors, Distributed Simulation, Flight simulation, Air Traffic Control simulation

ABSTRACT: *A distributed real-time simulation of the civil air traffic environment developed to support human factors research in advanced air transportation technology is presented. The distributed environment is based on a custom simulation architecture designed for simplicity and flexibility in human experiments. Standard Internet protocols are used to create the distributed environment, linking an advanced cockpit simulator, an Air Traffic Control simulator, and a pseudo-aircraft control and simulation management station. The pseudo-aircraft control station also functions as a scenario design tool for coordinating human factors experiments. This station incorporates a pseudo-pilot interface designed to reduce workload for human operators piloting multiple aircraft simultaneously in real time. The application of this distributed simulation facility to support a study of the effect of shared information (via air-ground datalink) on pilot/controller shared situation awareness and re-route negotiation is also presented.*

1. Introduction

Human-automation interaction is a critical consideration in the design and operation of advanced avionics and Air Traffic Control (ATC) systems. The MIT International Center for Air Transportation (ICAT) has developed an integrated human-centered systems approach to the design and evaluation of new air transportation technologies such as terrain avoidance systems, heads-up display (HUD) systems, and air-ground datalink systems [1,2]. This approach, which considers the human as an element of the closed-loop control system, relies heavily on the use of real-time, moderate-fidelity simulation to evaluate prototype systems with the human operator(s) in the loop. Because the systems being researched are typically evaluated early in the conceptual phase, the ability to rapidly prototype and exercise many alternate designs is of particular importance. Given this dynamic environment, flexibility and freedom in the design of the simulation architecture are important considerations.

One area of current research at ICAT is advanced information and communication systems, including air-ground datalink systems. Of particular interest is the effect of such systems on air traffic controller/pilot interaction and shared situation awareness. A distributed simulation of a portion of the national airspace environment was designed and developed to support this research, facilitating the evaluation of alternative datalink concepts. The distributed simulation facility includes an advanced cockpit simulator, an ATC simulator, and a pseudo-aircraft control and simulation management station.

2. Requirements

A distributed simulation was needed to place experimental human subjects operating separate flight and ATC simulators in a common simulation environment. Experiments designed to study pilot/controller interactions require a real-time simulation facility capable of modeling and coordinating representations of weather and traffic between the pilot and controller subjects. Voice communications among all participants in the simulation are necessary to facilitate the verbal interactions under investigation. A centralized means of recording data and voice communications from the simulation is necessary for analysis, and the ability to recall and playback previously-recorded simulation runs is needed to facilitate the subsequent debriefing of test subjects. Finally, a flexible architecture is desirable so that new simulation objects can be easily implemented and modified.

Additional tools are also required for generating test scenarios and managing air traffic in real time throughout the simulation. Human factors experiments often attempt to study specific interactions between humans and automation or other humans. Scenarios that place human subjects in situations requiring a response must be designed and coordinated to stimulate these interactions. In order to generate such scenarios, a scenario management application must be able to set the initial states for all aircraft in the scenario. While these initial states are the same for each execution of the simulation, the actions of the human subjects will vary. Therefore, all aircraft not under the control of human test subjects must be controlled in real time during an experiment to emulate each aircraft's response to its environment in a realistic manner.

3. Distributed Simulation Architecture

The requirements for a distributed simulation appropriate for human factors research motivated the development of a custom simulation architecture that could be implemented and tailored more easily than existing distributed simulation architectures, such as DIS (Distributed Interactive Simulation) or HLA (High Level Architecture). This architecture incorporates existing applications into a simulation protocol on top of a simple network communications layer.

3.1 Network architecture

Network communications are handled by standard Transmission Control Protocol/Internet Protocol (TCP/IP) layer sockets using full-duplex byte streams. This system was primarily designed to run on workstations connected to an Ethernet 10Mbps Local Area Network (LAN), although the use of TCP/IP communications allows applications to be run from remote locations that are connected to the Internet. This network implementation is simplified by relying on high bandwidth, reliable connectivity. At the hardware level, however, network integrity and bandwidth are sensitive to other hosts that are not part of the distributed simulation, but are still connected to the same LAN segment. Network traffic or errors from these hosts degrade the performance of the distributed simulation unpredictably during a simulation execution. Large simulations, which use all of the network bandwidth, may require computers participating in the simulation to be isolated to an independent LAN segment.

The network architecture follows a client-server model, as illustrated in Figure 1, which centralizes at the simulation host the collection and distribution of simulation data. Client applications, which may be flight simulators, ATC simulators, weather services (*e.g.*, the Total Atmosphere Ocean Space (TAOS) system [3]) or

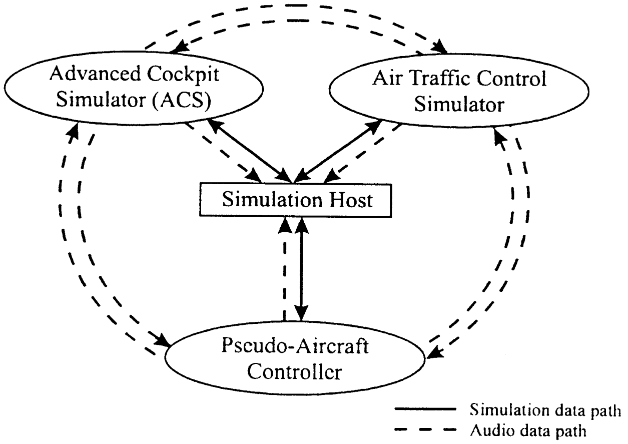

Advanced Cockpit Simulator (ACS)
Air Traffic Control Simulator
Simulation Host
Pseudo-Aircraft Controller
Simulation data path
Audio data path

Figure 1. Network architecture for the distributed simulation.

other applications, can enter or leave the simulation at any time by connecting or disconnecting from the host. The number of simultaneous connections supported by the host workstation's system kernel often limits the number of clients that may connect to a host, but no other limitations are imposed by the host software.

3.2 Simulation architecture

The host application controls all of the airspace information, such as the locations of airports and navigational aids, which are sent to client applications when requested, usually when the client first connects. Once connected, clients may declare objects (at present limited to aircraft and ATC types) that will be controlled by the client in the simulation. Clients may declare new objects (*e.g.*, aircraft taking off) or remove existing objects under their control (*e.g.*, aircraft landing) at any time during the simulation. There are no software limitations to the number of objects that may be declared by a client application.

The simulation host is responsible for keeping the simulation time. Updates of the simulation time are transmitted to client applications only when the client first connects, when the simulation time is disrupted—such as when the simulation is paused—or when a client explicitly requests an update.

The host application is also responsible for maintaining a log of the simulation execution. For analysis, the host may be restarted in a playback mode to replay the previously recorded simulation. Clients can then connect to the host to observe the simulation. For example, the flight simulator can connect using the same aircraft identifier string as any of the original aircraft in the simulation, and the cockpit simulator's attitude, trajectory, and alerting displays will reflect those of the original aircraft, even if that aircraft was a pseudo-aircraft.

3.3 Voice communications

Voice communications are also sent over the network, but the audio data is sent separately from the simulation data directly between the client computers in order to prevent transmission delays and to reduce the network load on the host computer. The dashed lines in Figure 1 represent the path of voice communications. While these are also full-duplex byte streams, voice data is sent in only one direction and is acknowledged in the return direction. The arrows on these paths indicate the direction of audio data only. The host may continue to receive and log the audio data, but it is not responsible for the distribution of the data. Participation in voice communications is therefore limited to clients that are explicitly declared at the outset of the simulation, because live communication streams must be established between all of the client computers. One advantage of decentralizing the voice communication is that multiple communication groups, analogous to different frequencies in radio communications, can be defined. A client may be programmed to participate in multiple communication groups at once, allowing the client operator to "tune" to a different communications channel ("frequency") when appropriate, although this capability has not been implemented in existing client applications.

4. Client Applications

In the following discussion, the screen captures from the different client applications that appear in Figures 2, 3, 4, and 5 were taken simultaneously during a simulation execution. During the discussion, note how the same weather cell and air traffic are perceived from the different client applications.

4.1 Advanced Cockpit Simulator (ACS)

The advanced cockpit simulator (Figure 2) is a part-task flight simulator that was developed to study human performance issues associated with advanced cockpit systems. The simulator emulates the Electronic Flight Instrument System (EFIS), Flight Management

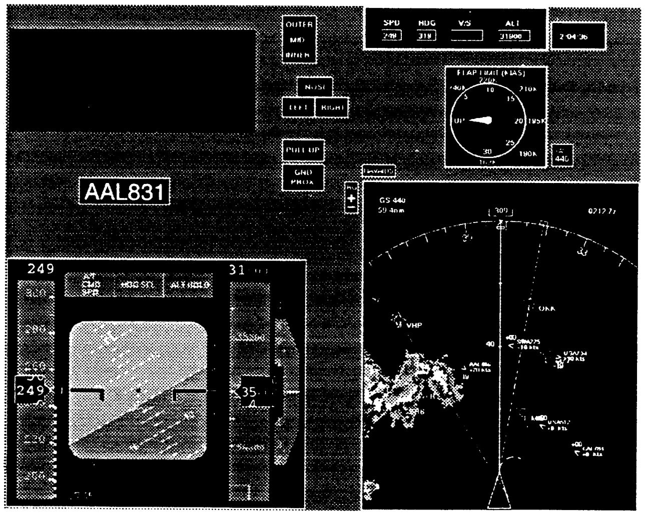


Figure 2. Advanced Cockpit Simulator (ACS) display, including prototype air traffic and weather displays.

Computer (FMC), and autoflight system found in modern "glass-cockpit" transport aircraft such as the Boeing 757/767 or 747-400. Entry of flight path information into the FMC is accomplished through a replica of the Boeing 757/767 Control and Display Unit (CDU). The autoflight system is controlled through a Boeing 737-200 autopilot Mode Control Panel (MCP). Direct flight controls are available using a side-stick controller and throttle quadrant, although these are not typically used when evaluating outer-loop, cognitive-level issues where it is assumed that aircraft control would be performed using the autoflight systems.

The cockpit simulator features advanced alerting and display systems for traffic, terrain and weather. A Traffic alert and Collision Avoidance System (TCAS) provides advanced warning of potential conflicts with other aircraft in the simulation. An Enhanced Ground Proximity Warning System (EGPWS) includes plan-, profile-, and perspective-displays of surrounding terrain. A wind shear alerting system indicates the presence and location of detected microburst activity. In addition, new traffic and weather display prototypes have been integrated into the cockpit simulator to support ongoing research into air-ground datalink systems.

4.2 Air Traffic Control Simulator

The Air Traffic Control (ATC) part-task simulator emulates the Plan View Display (PVD), Computer Readout Display (CRD), and Data Entry Control (DEC) used at most en route ATC centers in the United States. The PVD displays radar tracks and full data blocks for all tracked aircraft in the simulation within its assigned airspace sector, along with sector adaptation data such as airports, navigation aids, and airways. Although aircraft position updates are received continuously, target positions are updated once every 12 seconds on the PVD to emulate the update rate of the actual ATC equipment. Trackball inputs and/or alphanumeric keyboard commands may be used to display supplementary information such as a target's current trajectory, filed flight plan, or position history. The same input devices may be used to zoom or offset the plan view display. All data entry keyboard/mouse input sequences emulate those of the real DEC. In addition, a new NEXRAD-based weather display prototype has been integrated into the ATC simulator to support ongoing research into air-ground datalink systems. In Figure 3, which shows the ATC simulator display, flight plan information and a 6-mile segmented circle are displayed for the subject aircraft being simulated by the ACS.

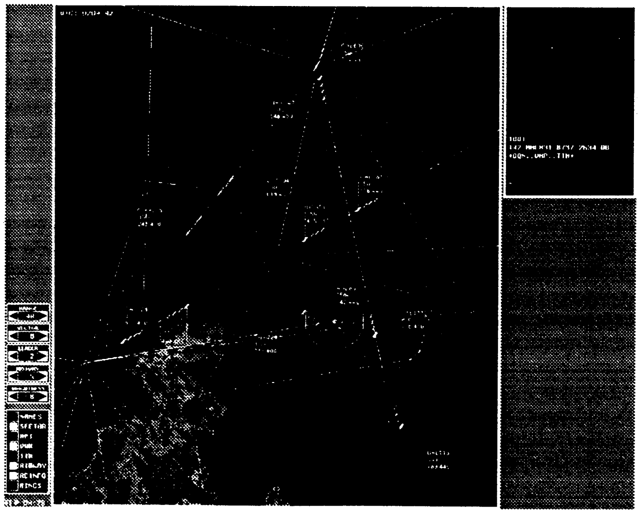


Figure 3. Air Traffic Control Simulator display, including a new NEXRAD-based weather display prototype.

4.3 Pseudo-Aircraft Controller

The pseudo-aircraft control station (Figure 4) manages simulation scenarios for human factors experiments in a distributed environment. This application allows for the creation and coordination of scenarios designed to place human subjects in predetermined situations, so that the response of the human subject to the situation can be studied. The client software enables a human operator to quickly control and manage the simulated air traffic in real time during an experiment. This application also simulates the flight dynamics of all pseudo-aircraft under its control. (For large simulations, this task may be distributed among multiple workstations running this client application, each controlling a subset of the pseudo-aircraft traffic.)

Many existing pseudo-aircraft control applications require the pseudo-pilot to use mouse clicks and alphanumeric commands to effect changes in flight paths or flight plans of the simulation pseudo-aircraft [4,5]. This control scheme requires the pseudo-pilot to quickly alternate between the mouse and keyboard. While this may be acceptable for small numbers of pseudo-aircraft or infrequent clearance changes from ATC, it quickly becomes unmanageable in the high-density, high-workload environments that are of primary interest in current Air Traffic Management (ATM) research.

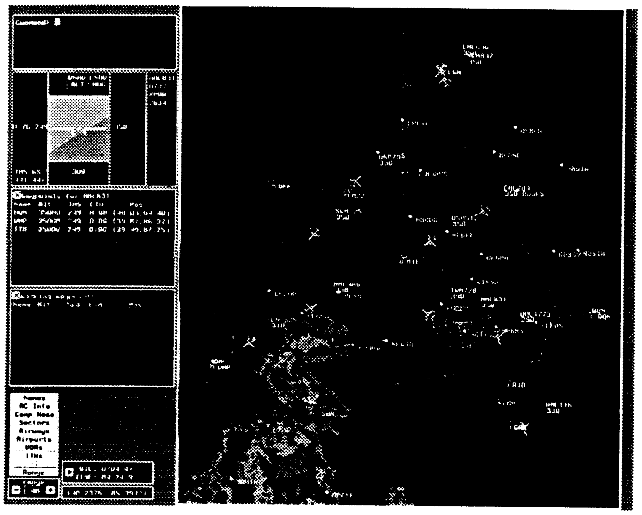
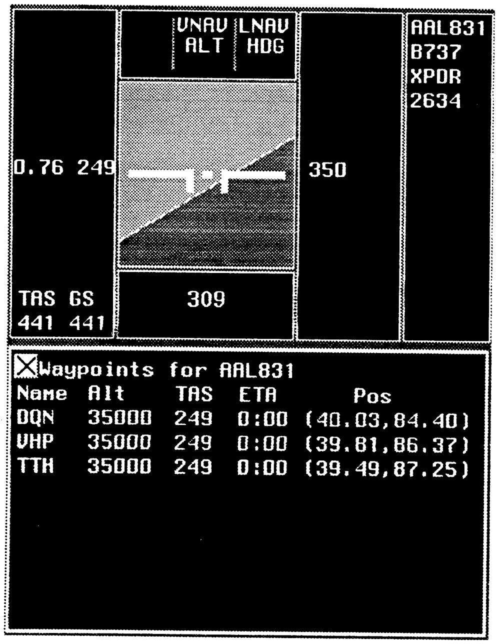


Figure 4. Pseudo-Aircraft Controller display.



```
VNAV LNAV        AAL831
ALT  HDG         B737
                 XPDR
                 2634
0.76 249      350

TAS GS
441 441   309

Waypoints for AAL831
Name Alt    TAS ETA   Pos
DQN  35000  249 0:00  (40.03,84.40)
VHP  35000  249 0:00  (39.81,86.37)
TTH  35000  249 0:00  (39.49,87.25)
```

Figure 5. "Pseudo-Cockpit" display.

An interface to the pseudo-aircraft control application was developed to enable real-time control of pseudo-aircraft by providing the pseudo-pilot with an intuitive "point-and-click" interface to exercise outer-loop control of an aircraft's autoflight systems. The pseudo-pilot is provided with a plan view display of the air traffic, which is continuously updated during the simulation. The pseudo-pilot can click on any aircraft to display that aircraft's "pseudo-cockpit", showing the aircraft's current attitude, airspeed, altitude, heading, and flight control mode, as well as its commanded states (Figure 5). The pseudo-pilot may also display the current waypoints for the selected aircraft, both textually in a list and graphically on the PVD.

If an aircraft object is under the pseudo-pilot's control (as distinguished by its blue color; other aircraft appear red on the display), the pseudo-pilot may change the aircraft's commanded states by using the *second* mouse button to click in the appropriate area of the screen (using the second mouse button rather than the primary mouse button prevents the pseudo-pilot from inadvertently changing the commanded state of a pseudo-aircraft). When the mouse cursor is in the PVD, a heading cue is displayed at 5-degree increments on the compass rose surrounding the selected aircraft and is also shown numerically (Figure 6). This cue aids the pseudo-pilot in determining the heading to a navigational aid or a heading clear of weather. This heading can be commanded by clicking the second mouse button. Similarly, a target altitude or airspeed is selected by clicking on the appropriate tape indicator. Flight control modes are set by clicking on the flight mode annunciators shown in Figure 5. Using just these controls, a pseudo-pilot is able to perform most of the routine tasks necessary to manage the air traffic during a simulation. (Note that in Figure 5, the subject aircraft being simulated by the ACS is selected as the active aircraft, so it cannot be controlled from the pseudo-aircraft control station.)

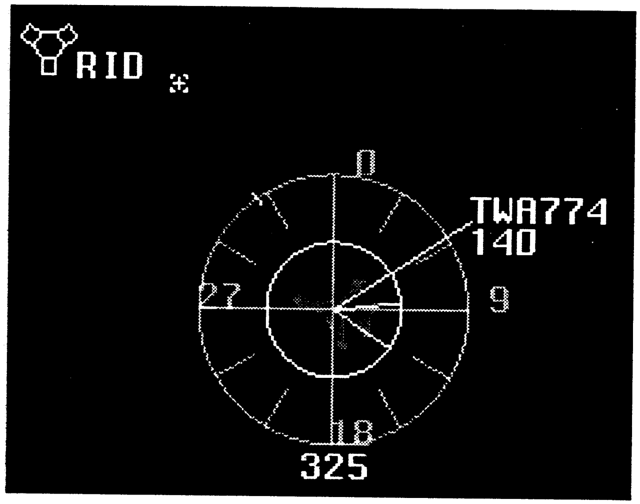


Figure 6. Compass rose surrounding a pseudo-aircraft with heading control cue displayed. The small set of crosshairs appearing to the right of the navigational aid is the mouse pointer.

Scenario generation and more sophisticated manipulation of pseudo-aircraft—such as programming and modifying waypoints or changing the actual states rather than the commanded states of an aircraft—are accomplished using a command line interface. This interface includes commands for creating, naming, and removing aircraft; manipulating and copying aircraft waypoints; and saving and restoring scenarios which provide the initial conditions for a distributed simulation.

The pseudo-aircraft controller also includes some elements of a robust situation generation approach developed by Johnson [6]. Robust situation generation is a method of automating pseudo-aircraft trajectories using state feedback to generate specific air traffic situations. For example, an experiment may require a collision hazard situation if no action is taken by the experimental subjects. The ability to reliably generate this situation is sensitive to the unexpected actions of the human subjects (*e.g.*, an unrelated course deviation requested by ATC long before the desired conflict). To make the situation more robust, the pseudo-aircraft can be set to adjust its speed to arrive at the desired conflict location at the appropriate time. Only some elements of the robust situation generation implementation could be included for use in the pseudo-aircraft control software, because many of the actions that pseudo-aircraft must take to reliably generate a situation require ATC clearance.

Finally, due to its real-time display and control interface, an instance of the pseudo-aircraft controller client running idly (*i.e.*, controlling no pseudo-aircraft) is ideal for observation of the simulation by those not actively participating. It may also be used to view playbacks of the simulation. This is especially useful during the debriefing portion of a human factors experiment, when it may be beneficial for the test subjects to review the simulation with all weather and traffic information revealed.

4.4 Weather Application (TAOS)

For the demonstration and experiment described herein, NEXRAD-based weather was integrated into the cockpit, controller and pseudo-aircraft displays statically (see Figures 2, 3, and 4). The data was collected and archived by a tool like TAOS (Total Atmosphere Ocean Space [3]), and then a series of static images were distributed off-line to the simulation suite. There was no link to real-time dynamic weather during the simulation.

5. Execution Example

This distributed simulation facility is currently in use to support a study of the effect of shared information (via air-ground datalink) on pilot/controller shared situation awareness and re-route negotiation. The experiment pairs a commercial airline pilot subject with an en route air traffic controller subject in a real-time simulated air traffic environment. The availability of shared traffic and weather information is manipulated as an independent variable in the experiment.

Test scenarios intentionally bring the goals of the pilot and controller into conflict in re-routing situations. Subjects interact within the simulation environment to resolve traffic and weather conflicts. Of particular interest are indications of each subject's recognition of the other's constraints, anticipation of needs and/or desires, willingness to comply/cooperate, and persistence in pursuing an alternate solution. The experiment will provide input in terms of the potential for shared information to effect more collaborative or competitive interaction between pilots and controllers.

In this experiment, each pilot/controller pair participates in six scenarios. The discussion that follows focuses only on one run of the distributed simulation executed during this experiment as an example of the performance typically achieved by the distributed simulation facility. This particular scenario contained one subject aircraft simulated using the ACS, 16 pseudo-aircraft controlled by a single execution of the pseudo-aircraft control application, one air-traffic controller, and a weather front, which provided the impetus for re-route negotiation. In this case, both the ACS and the simulation host application were run on an SGI Indigo^2 workstation. The ATC simulator was run on another SGI Indigo^2 workstation and the pseudo-aircraft control station was run on an SGI Octane workstation. The audio logging function was separated from the simulation host and run on an SGI Indigo workstation.

The 16 pseudo-aircraft which comprised the surrounding air traffic were managed by a single pseudo-pilot who was also responsible for accepting and responding to radio calls from the air traffic controller. The number of aircraft that a single pseudo-pilot can manage using the pseudo-aircraft controller is dependent on the pseudo-pilot's experience, so an upper limit to this number could not be determined.

Figure 7 shows the air traffic and weather front as seen from the pseudo-aircraft control station during this simulation execution. During this execution, both the subject pilot and the controller had access to air traffic and the weather radar information. To maintain aircraft separation and avoid the hazardous weather, the controller issued 17 route amendments over the execution's twelve-minute duration. Eight of the pseudo-aircraft were forced to deviate off course to avoid the weather front and/or other air traffic. The pseudo-pilot was able to negotiate and successfully accomplish all 14 ATC clearance changes directed toward the pseudo-aircraft in real time.

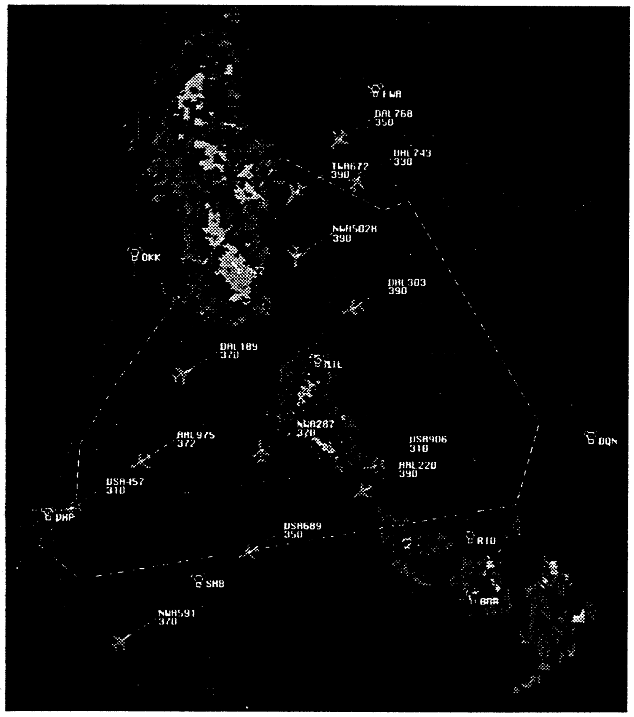


Figure 7. Air traffic and weather front as viewed from the pseudo-aircraft control station several minutes into a simulation execution.

Figure 8 shows the data rates experienced during this execution of the simulation, not including the bandwidth required by the voice communications. These values were obtained from the simulation log files by averaging the amount of data being transmitted during each second of the simulation. Therefore, these values do not reflect the actual instantaneous transmission rates experienced during the simulation. In this case, the average data rate required for the simulation data was 156 Kbytes/s. 75 audio transmissions were made during the simulation, each lasting an average of 3.9 seconds. The data rate for the audio data was 16 Kbytes/s, increasing the network load by an additional 48 Kbytes/s during each transmission. Because data must be repeated to each client application subscribing to the data, the bandwidth requirements for the simulation execution scale linearly with the number of clients connected. The bandwidth requirements do not necessarily scale linearly with the number of objects in the simulation, because the update rate for each object in the simulation depends on the speed of the computer controlling that object.

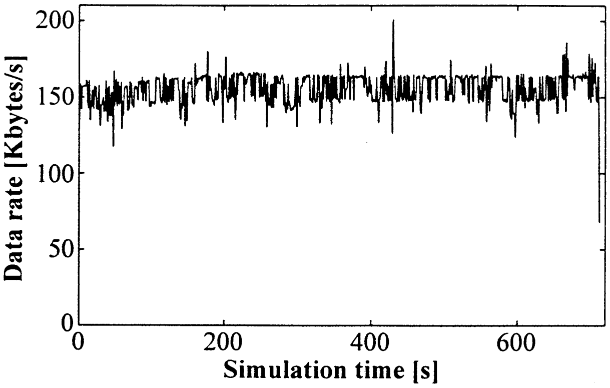


Figure 8. The data rate of the simulation data transmission plotted as a function of time during a single execution of the simulation.

Although the voice communications functioned normally during this execution of the simulation, some runs that were of comparable complexity as the one described above experienced interruptions and delays in the audio transmissions. Voice communications, which are more sensitive to network delays than the simulation data transmissions, may have been interrupted by an increase in the load on the network that was observed during these executions (while no attempts were made to completely quantify these delays, the network latency measured during these executions was on the order of a second, compared to the millisecond latency experienced during normal network operations). It is likely that transmission of the simulation data was similarly delayed during these executions, although this was not noticeable to the human subjects. As discussed in Section 3.1, future simulation exercises may require that participating computers be isolated to an independent LAN segment.

This distributed simulation architecture has also been validated in a remote simulation execution incorporating simulator facilities at MIT, located in Cambridge, Massachusetts and TASC, located in Reading, Massachusetts. TASC installed the host software and acted as the simulation server. TASC also installed and executed the ATC client application and the pseudo-aircraft controller application, while MIT executed the advanced cockpit simulator. The simulation appeared to function normally, although voice communications were not attempted in the remote simulation.

6. Conclusion

A distributed real-time simulation of the civil air traffic environment developed to support human factors research in advanced air transportation technology has been presented. The distributed environment is based on a custom simulation architecture designed for simplicity and flexibility in human experiments.

Several client applications—including an advanced cockpit simulator, an en route ATC simulator, and a pseudo-aircraft control station—have been developed to support real-time experiments with humans in the loop. The pseudo-aircraft control station in particular enables the creation of scenarios that govern a human experiment in a distributed environment. Once the simulation has begun, the pseudo-aircraft control station enables a single user to manage multiple aircraft emulating the air traffic observed by the human subjects.

This distributed simulation facility has been demonstrated in a study of pilot/controller re-route negotiation that is evaluating alternative datalink concepts. The experiment successfully joined pilot/controller pairs in a distributed airspace environment, although some difficulties were encountered with the voice communications. Preliminary results from this study indicate that shared information improves the situation awareness of pilots and controllers. While there is evidence from this study that improved situation awareness enables pilots and controllers to work more collaboratively in re-routing situations, there is also evidence from this study that improved situation awareness causes mistrust or frustration when the goals of the pilot and the controller are in conflict. The distributed simulation facility will be used to explore these human factors issues more fully in future experiments.

7. Future Work

In order to take advantage of real-time weather data, the air traffic management simulation could transition to DIS or HLA, which would allow it to make use of the weather and effects server capabilities of TAOS. TAOS provides consistent, tactically significant, high-fidelity environmental data on demand to distributed simulation federations. TAOS environmental data service provides a detailed dynamic description of the combined atmosphere-ocean-littoral natural environment using 4-D grids (three spatial dimensions plus time) to provide a common representation of the environmental base fields and embedded features. Base fields describe the ambient conditions, such as a temperature or wind field, while embedded features are fine-scale localized processes, such as clouds or dust storms. TAOS provides links to a wide variety of external data sources, ranging from live observations and data fields from operational sources (*e.g.*, commercial radar feeds and AWN, Automated Weather Network), to authoritative gridded forecast products provided by DMSO's MEL (Master Environmental Library) or public Internet sites.

Future development of this simulation facility calls for the integration of real-time weather models, to include four-dimensional wind, temperature, turbulence, icing, and convective weather phenomena. These weather elements are critical to a realistic simulation of air traffic management. Although this set of weather parameters is slightly different than the data set provided during the STOW'97 ACTD (Synthetic Theater of War Advanced Concept Technology Demonstration) and USACOM's (U.S. Atlantic Command) exercise, Unified Endeavor UE 98-1, TAOS can provide the additional parameters describing turbulence and icing. However, there are issues to be addressed with the temporal and spatial resolution of the data required for air traffic management scenarios that typically run in a smaller playbox (on the order of several hundred nautical miles, with greatest interest in the area surrounding an airport) and over a much shorter time (on the order of minutes or hours).

8. Acknowledgments

This work is supported by TASC as part of the FAA Center of Excellence in Operations Research and by the National Aeronautics and Space Administration/Ames Research Center under grant NAG 2-716. TAOS work is supported by the Defense Advanced Research Projects Agency and the Defense Modeling and Simulation Office through its Modeling and Simulation Executive Agents for the Natural Environment. The U.S. Army Topographic Engineering Center serves as the DARPA agent for Synthetic Environments and as the contracting organization for this work.

9. References

[1] E.C. Hahn and R.J. Hansman, Jr.: "An Experimental Study of the Effects of Automation on Pilot Situational Awareness in the Datalink ATC Environment" ASL-92-1, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 1992.

[2] A.H. Midkiff and R.J. Hansman, Jr.: "Identification of Important 'Party Line' Informational Elements and the Implications for Situational Awareness in the Datalink Environment" ASL-92-2, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, May 1992.

[3] D.A. Whitney, R.A. Reynolds, D.Z. Sherer, P.S. Dailey, M.L. Driscoll, MAJ M. Zettlemoyer, CDR R. Schultz, and I. Watkins: "Impacts of the Environment on Warfighter Training: STOW 97 Experiences with TAOS" 98S-SIW-224 *Spring Simulation Interoperability Workshop*, March 1998.

[4] R.A. Weske and G.L. Danek: "Pseudo Aircraft Systems: A Multi-Aircraft Simulation System for Air Traffic Control Research" *Proceedings of the AIAA Flight Simulation Technologies Conference*, Monterey, California, August 1993.

[5] J. Klinge, K. Smith, and P.A. Hancock: "DATIDS: The University of Minnesota Distributed Air-Traffic Information Display Simulator" *Proceedings of the 9th International Symposium on Aviation Psychology*, Columbus, Ohio 1997.

[6] E.N. Johnson: "Multi-Agent Flight Simulation with Robust Situation Generation" Master's Thesis, Department of Aeronautics and Astronautics, MIT, 1995.

Author Biographies

KEITH AMONLIRDVIMAN is an undergraduate student at MIT in the Department of Aeronautics and Astronautics, where he is a research assistant at the MIT International Center for Air Transportation.

TODD FARLEY is a graduate student at MIT in the Department of Aeronautics and Astronautics, where he is a research assistant at the MIT International Center for Air Transportation.

R. JOHN HANSMAN is a Professor at MIT in the Department of Aeronautics and Astronautics, where he is Head of the Humans and Automation Division and Director of the International Center for Air Transportation. He conducts research in several areas related to flight vehicle operations and safety. His current research activities focus on advanced cockpit information systems, including Flight Management Systems, Air-Ground Datalink, Advanced Alerting Systems, and Flight Crew Situational Awareness.

JOHN LADIK is a principal member of the technical staff at TASC involved in systems engineering, mathematical modeling, and statistical analysis.

DANA SHERER is a senior member of the technical staff at TASC. Her expertise is in systems engineering with a focus in simulation and modeling, environmental simulation systems, distributed computing, data analysis and scientific data visualization.