A MODULARIZED APPROACH FOR COMPREHENSIVE AIR TRAFFIC SYSTEM SIMULATION

Joachim K. Plaettner-Hochwarth[1] & Yiyuan J. Zhao[2]
University of Minnesota, Minneapolis, MN 55455

John E. Robinson III[3]
NASA Ames Research Center
Moffett Field, CA 94035

Abstract

This paper explains the design and computer implementation of a comprehensive dynamic air traffic system simulation program known as ComDATSS, currently under development at the University of Minnesota. ComDATSS simulates all essential components of an air traffic management system and their dynamic interactions. It provides a numerical testing environment for evaluating advanced ATM system designs and automation tools. Compared with existing ATM system simulation programs, ComDATSS employs an "as-it-is" physical modeling approach, generates four-dimensional aircraft trajectories, and allows for sufficient randomness in the simulations. In addition, the computer implementation of ComDATSS is implemented as a modular structure that can easily be modified and maintained.

Introduction

Air traffic in both the United States and the world has increased significantly over the last several decades, and is projected to grow further. These increases have severely affected the existing air traffic control infrastructure in the United States, and have triggered the wide-spread development of air traffic management facilities in many other countries. These increases have also spurred active research efforts by governments, industries, and universities on air traffic management to improve the efficiency and safety of air travel.

Several automation tools have been developed to assist the pilots in reducing fuel consumption and providing separation assurance, and to assist controllers in guaranteeing flighty safety and increasing efficiency. The Flight Management Systems (FMS) on board aircraft can generate optimal flight paths that achieve specified arrival times while saving fuel. The Tactical Alert and Collision Avoidance System (TCAS)[1-2] can warn pilots of immediate intrusions, and even provide pilots with advisories for conflict avoidance. In the last decade, researchers at NASA Ames Research Center have been developing a set of computer tools for air traffic control automation, called the Center/TRACON Automation System, or CTAS[3-8]. CTAS can assist air traffic controllers by generating flight advisories that guarantee flight safety and allow for an expedient flow of traffic.

Recently, the concept of free flight has been proposed for future air traffic management (ATM) [9]. In a free flight environment, individual aircraft can change their flight paths in real time, in order to achieve optimum flight paths. Controllers are only supposed to intervene to resolve potential conflicts. While the feasibility, potential problems, and implementation details are still being investigated, the free flight concept has provoked studies on a series of advanced air traffic management issues.

A major challenge in the development of ATM automation tools and novel concepts is the verification and testing of these concepts. It is important to verify the adaptability and benefits that a sub-system can offer to the entire ATM system. An advanced system concept must guarantee safety, and be compatible to the operations of various system components.

In the history of single aircraft development, a combination of basic theories, computational methods, numerical simulations, wind tunnel tests, and flight

[1] Ph.D. Candidate, Department of Aerospace Engineering & Mechanics, Student Member AIAA.
[2] Associate Professor, Department of Aerospace Engineering & Mechanics, Senior Member AIAA.
[3] Research Engineer.

tests have been used. However, the air traffic management system is an enormously complex system. First-order theories in many cases do not yet exist. Actual flight testing of new system concepts, such as conflict resolution algorithms and free flight concepts, is simply too risky and too expensive to conduct frequently. As a result, a system-wide computer simulation becomes an essential means for the verification and testing of novel tools and concepts.

This paper documents the architecture design of a comprehensive dynamic air traffic system simulation program (ComDATSS) currently under development at the University of Minnesota. This program utilizes advances in computer science and models all essential components in an ATM system: aircraft dynamics, 4-D flight trajectories, pilot controls, aircraft navigation, radar/data-link, weather conditions, and controllers. It is basically a fast-time simulation program with options for real-time simulations. It can be used for simulating both existing infrastructures and future concepts such as free flight.

A site in the Internet has been created that provides updates on the current status of the development of ComDATSS: http://www.aem.umn.edu/research/atc.

Before we proceed with the description of ComDATSS, several terms need to be defined and relative merits of existing ATM simulation programs need to be reviewed.

System-Wide Computer Simulations

A simulation program is designed to emulate the operation of an actual system. In computer simulations, mathematical models are used to represent the operations of physical objects. Occasionally, some physical objects may be connected with a computer to conduct hardware-in-the-loop simulations. Computer simulations can be conducted either in fast-time, or real-time. For computer simulations, mathematical models of physical objects can either be analytical expressions or numerical solutions of a set of equations. These models can mainly reveal the input/output relations of a physical object, or describe the inner working details of the object.

Computer simulations may be classified according to implementation structures inside a computer. Traditionally, computer simulations are used to simulate the progression of a single system, such as the flight of an aircraft. A system-wide simulation, on the other hand, needs to emulate the operations and interactions of all relevant components in a system. In these simulations, different computer processes need to be created to emulate the behavior of different projects, and proper synchronization among these processes must be achieved. If calculations are carried out in several computers, connected through a network or other communication devices, the simulations can be called distributed simulations. Synchronization of various calculations is important whether a single object or an entire system is simulated. Some computer simulations are mainly used to describe the input/output performances of a system or an object, such as in benefit studies. These simulations can be considered simulation modeling. On the other hand, some simulation programs model the inner working details of a system or an object, and can be used to test the operation of the system or the object. This latter type can be called simulation testing.

For the system-wide computer simulation of the air traffic system, the following objects and elements need to be modeled: Physical objects include aircraft, radar, communication, navigation & surveillance (CNS) devices, data-links, and display equipment. Human participants include controllers, pilots, and dispatchers. Natural environment elements include weather conditions, terrain, and airspace geometry. In addition, the simulation system needs to model procedures, such as standard arrival routes (STAR), standard instrument departure (SID), FMS, flight plans, etc. Descriptions of operational constraints, such as runway capacities and separation standards, are also needed in the simulation process. Finally, certain inputs are needed to start the simulation process. These inputs include the initial times and locations when arrival and/or departure traffic enter the simulation program, and aircraft characteristics of this traffic including the respective flight plans.

Review of Related Work

Because of the importance of modeling and computer simulation for the evaluation of new ATM system concepts, a wide range of simulation models have been developed and used in various parts of the world. Fast-time simulation was recognized as an important analysis tool in the development of CTAS[5]. A simple fast-time simulation program was developed at NASA Ames Research Center. This program is essentially a queuing model. It generates stochastic arrival times of aircraft traffic entering into the terminal radar approach control area (TRACON) via meter fixes. Simplifying assumptions on air traffic control are made that include the use of fixed time-based separation constraints at the threshold and meter fix, constant times for an aircraft to fly between the feeder-fix and the runway as a function of gate. However, actual aircraft trajectories are not calculated. As a result, conflicts in flight are ignored and information needed for controller decision-making is not generated.

NASA Ames Research Center has also developed a real-time hardware-in-the-loop simulation environment (simulator) for air traffic control systems[8]. The operation of this system involves controllers and pseudo-pilots. The NARSIM[10] system developed at NLR is also a real-time simulation package consisting of both computer models and hardware equipment.

Sponsored by NASA Ames Research Center under the AATT[11] program, a group of scientists at MIT surveyed and compared most existing ATM modeling capabilities. Existing models were divided into different categories according to their intended applications, such as capacity and delay models, conflict detection and resolution models, human/automation models, cost/benefit models, and noise models. Existing models were also classified into microscopic, mesoscopic, and macroscopic models, depending on the levels of modeling details. It was concluded that some serious deficiencies still remain. In particular, some of the best existing models still suffer in several fundamental aspects including: lack of sufficient stochastic options, limited representation of weather and winds, in-adaptability to new ATM concepts, such as free flight, and the use of a large amount of resources (acquisition costs, training, input/scenario setting-up, and data collection). Significant support for the development and experimentation of ATM system models was recommended.

The comprehensive system-wide simulation model described in the this paper is designed to provide a numerical testing environment for the research and development of ATM automation tools, advanced algorithms, and novel concepts. In this regard, it is somewhat related to three of the existing simulation packages: TAAM, SIMMOD, and RAMS. These three models represent some of the best modeling capabilities at the present time.

TAAM, or Total Airspace and Airport Modeler, was developed by the Preston Group in Australia. It is currently one of the most advanced existing simulation models that includes a system-wide, microscopic implementation to simulates the entire air traffic system. One of the major drawbacks of this software is the price tag. A single site license cost about $350,000 in 1997. Besides the high price it also lacks stochastic options and does not cover all ATM components. The rule set, it uses, is fixed and thus is inflexible to be used for testing new ATM algorithms/concepts. Through its complexity it requires a lot of resources and training to set up.

The SIMMOD model was developed by the FAA. It can be categorized by its main feature as an airfield and terminal area airspace model. The major drawback of this model is that it uses a node-link system on which all the aircraft move, as opposed to a 3D simulation like in TAAM. Because of this implementation type it is inflexible and cannot be used to simulate new concepts like free flight.

Eurocontrol in Europe developed the RAMS (Reorganized ATC Mathematical Simulator). It is a General purpose ATC modeling environment for en route and terminal airspace as well as controller workloads. Since this software is only available through Eurocontrol and since it uses a closed structure, its assumptions are basically unknown.

Recently, researchers at NASA Ames Research Center have developed a computer simulation program known as Future ATM Concepts Evaluation Tools, or FACET[12]. In addition, researchers at NASA Langley Research Center are also developing a desktop computer program for simulating free flights[13]. It is encouraging to see that the need to develop new computer simulation capabilities for ATM research is being recognized.

Developing System-Wide Simulation for Air Traffic Systems

A system-wide computer simulation program for air traffic systems should desirably have some or all of the following properties: It should be sufficiently <u>flexible</u> to be used for simulating both the current and future systems. For example, it should be possible to use this simulation system to study the free flight environment, and/or the use of data-links. A good system-wide simulation program should offer sufficient <u>realism and accuracy</u>. To this end, theoretical modeling of objects and procedures should be made carefully and consistently, so that the system-wide simulation can generate meaningful results. Simultaneous operations and interactions of various objects should be simulated well, within the capabilities of computer resources. In addition, a simulation program should be developed for <u>easy maintenance and expansion</u>, and should be made <u>user-friendly</u>. Finally, sufficient data points should be recorded during a simulation run and <u>convenient post-simulation analysis</u> tools should be available as well.

There are a host of research issues in the proper development of system-wide simulations. These issues include mathematical modeling of objects in an air traffic system, such as aircraft dynamics and radar response characteristics. The time progression scheme for a system-wide simulation should be selected carefully to balance simulation speed, accuracy, and synchronization. In addition, the development of a system-wide simulation requires a careful architectural design of the computer implementation.

A Comprehensive Dynamic Air Traffic System Simulation program (ComDATSS) is being developed at the University of Minnesota. Compared with existing simulation programs, ComDATSS is intended to be a flexible and convenient numerical testing tool for researchers and developers. It is designed to provide features that are not available in existing simulation models, and to take advantages of advanced computer structures.

Specifically, ComDATSS has or will have the following features: (1) It provides a microscopic modeling of all essential components in actual air traffic system operation. These components include aircraft dynamics, pilots, navigational systems, controllers, dispatchers, weather elements, surveillance via radar and/or data-links, and necessary communications. It simulates the working process of an entire ATM system and thus provides a numerical experiment environment. (2) Four-dimensional aircraft trajectories are generated in the simulation program by an "as-it-is" approach. In this approach, intended aircraft trajectories are expressed as flight objectives, which are consistent with flight plans, specified flight procedures, and controller commands. These flight objectives are fed into pilot models. Pilots control aircraft motions according to measured aircraft conditions from navigation systems, and aircraft respond to pilot controls under ambient conditions and limitations of aircraft dynamics. In particular, airspace geometry affect aircraft flight paths through flight objectives and flight constraints. As a result, ComDATSS offers sufficient flexibility in simulating the current ATM infrastructure, any advanced concepts such as free flight, as well as any part of the airspace. (3) Sufficient randomness can be included in the simulation at every ATM component. (4) Initial traffic in the ComDATSS simulation can either be supplied in the form of recorded actual traffic, or internally generated using assumed stochastic distributions. The latter choice is especially convenient for individual researchers to test concepts quickly. (5) Both fast-time and real-time simulation modes will be accommodated. (6) ComDATSS has been coded using C++ with multi-process structures. Modular designs have been used for easy modification, expansion, and maintenance.

When completed, ComDATSS can be used for a wide variety of ATM researches and development needs. These applications can include the initial testing of ATC automation tools and algorithms, and evaluations of benefits of improving individual ATM components. In the real-time mode, one can play the role of a controller managing a large number of aircraft or a pilot flying through a congested airspace. In other words, it can be used as an education tool. Compared to real-time hardware-in-the-loop simulations, ComDATSS can be operated by one person and can be used in a controlled simulation environment.

ComDATSS also provides opportunities and platforms for studies of theoretical modeling of aircraft trajectory prediction accuracy, pilot responses, controller decision-making, etc. These modeling processes are useful for the understanding of ATM system operations.

The development of ComDATSS involves three coupled steps: (1) design of computer implementation architecture, (2) mathematical modeling of all essential components in an ATM system, and (3) computer programming. At the present time, a prototype simulation program has been developed. This prototype program is comprised of a computer architecture that contains modules for all ATM components, interactions of various ATM components, stochastic generation of arrival aircraft traffic, graphical display, and controller command entries. A fixed-increment time advance scheme and numerical solutions of aircraft dynamical equations are used to simulate the ATM system progression.

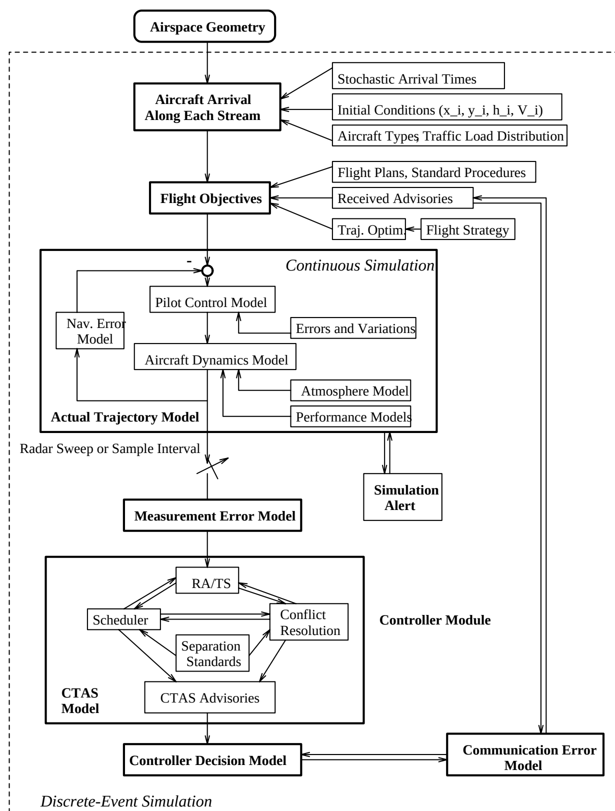

Fig. 1 ATM components and information flow in a comprehensive simulation.

Computer Implementation Architecture

Figure 1 shows the information flow among various components in an air traffic system.

For the easy maintenance and expansion of the simulation program, a modular approach has been adopted for the computer implementation. A module is used to represent each ATM component, and each module interacts with other modules through specified inputs and outputs. Internal logic of these modules can be modified at any time. Figure 2 shows how the modules of ATM components are integrated in ComDATSS. It also shows the definitions of classes and functions for an object-oriented programming (OOP) environment. Boxes with shadows represent classes whereas plain boxes represent functions. Also shown are information connections among these modules. Each module can be represented by an object. Thus, a module in later mention always refers to an object and vice versa.

The actual core of the software is written entirely in the object-oriented C++ programming language. The object-oriented programming approach makes it possible to simulate the entire air traffic system as it is. This requires to model all parts of the ATM system as objects and to let them communicate with each other as in reality. C++ was chosen since it is certainly one of the most widespread and powerful object-oriented languages. Java – with the currently available computational power – lacks sufficient execution performance, which is essential for a comprehensive detailed simulation of an ATM system.

Several processes are defined in the computer implementation. Computer processes are independent programs that perform various tasks in the simulation program. Most of these processes directly represent an ATM component whereas others are more of supportive nature. All processes are implemented in such ways that they can be run on different machines within a network, if necessary, so that the computational load can be distributed. The data exchange and communication among various modules are handled by a special protocol. If the simulation processes are run on a single computer, it depends on the operating system design of

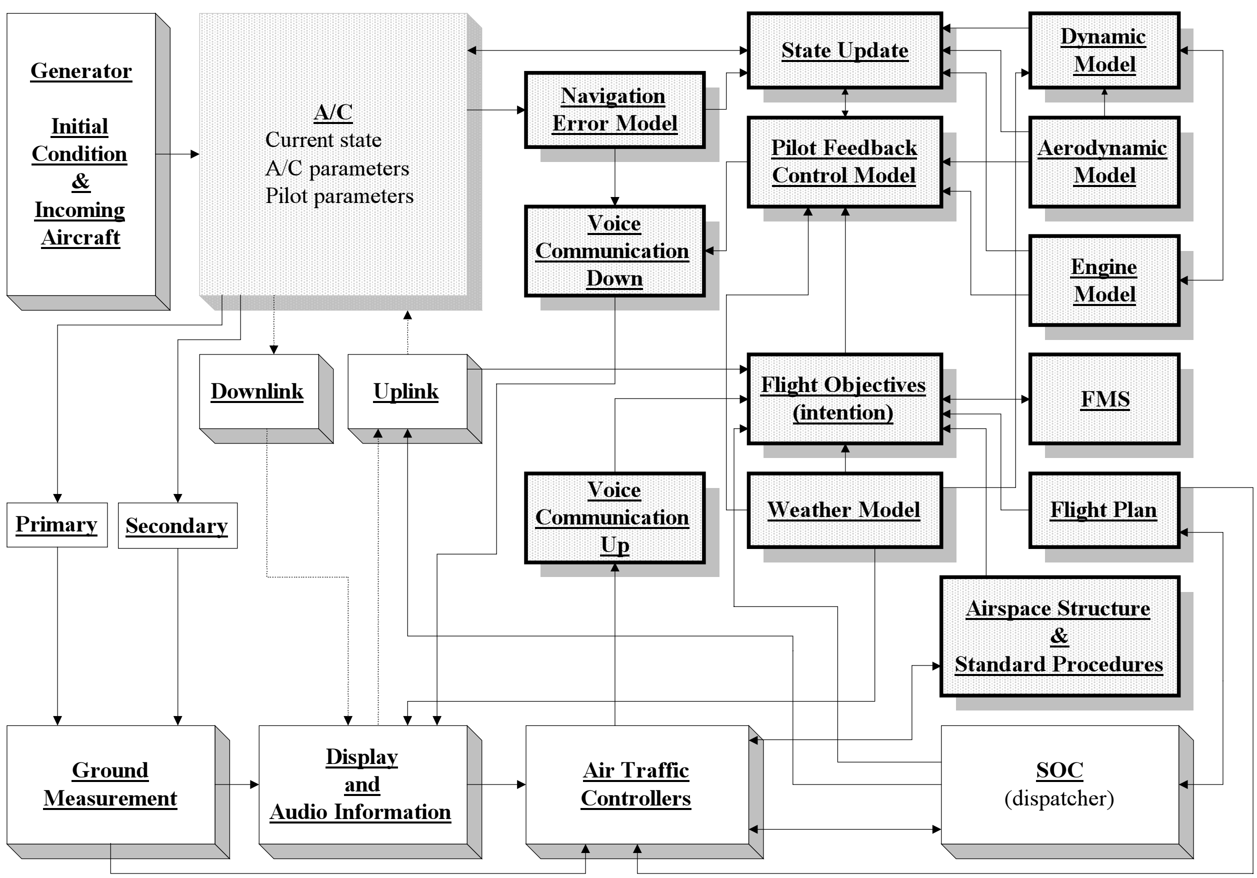

Fig. 2 Modules of the computer implementation.

this particular computer whether the processes are run in parallel or not. On a single processor machine, a parallel computation is not possible. For a distributed configuration over several computers, the processes are run in a parallel manner.

Descriptions of Computer Processes

- Administrator

This process is the core of the simulation. Its main task is to handle all the aircraft and update their states. When the administrator is started, it initializes the simulation environment and waits for other processes to connect. During this time, a file is loaded which describes the initial airspace with all present aircraft. Once all connections are established the simulation can be started. As the simulation evolves, the administrator accepts commands from controllers and forwards them to the respective aircraft and its Flight Objective Module. The internal objects - that represent the aircraft themselves - handle the commands and carry out appropriate maneuvers. A message from an aircraft to the controller is forwarded in the opposite direction with the same internal mechanisms. In addition to the controller, an airline dispatcher can be added. In computer implementations, dispatchers would communicate with the aircraft in the same way as controllers do.

In the simulation environment, an independent object is used to represent each aircraft. In addition, an integrator is employed to support the updating of aircraft states. In periodic cycles, the administrator calls these objects to update their states. The first task an aircraft performs when it receives an update call is to look for pending commands from the controller. If there is no command pending, numerical integration of the aircraft dynamics equations are carried out by the integrator.

If a new aircraft is created, the administrator adds the new aircraft to its internal list and includes it in the update cycle. Besides aircraft that were created by the generator, the administrator can also handle aircraft that were handed over by another administrator. Actually, one administrator is able to represent an airspace with several sectors and respectively assigned controllers. It can also carry out the hand off procedure between controllers from different sectors. The mechanism to accept aircraft from other administrators enables the option of distributing the workload of updating aircraft to several computers. In order to reduce the workload of the administrator, so that it can update the aircraft in fast cycles, the task of distributing aircraft information is allocated to another independent process called distributor. The administrator periodically sends the relevant data of all available aircraft to the distributor.

- Distributor

The distributor process does not correspond to an actual component in an ATM system. Instead, its task is to distribute the available aircraft data in various ways. The first option is that another process can connect to the distributor to request the aircraft data. This data is the result of the update by the administrator process and is available after each update cycle. With this option a direct down-link from the aircraft to the controller can be included. As a second option, a ground measurement unit such as radar can be simulated. Radar processes can request aircraft data from the distributor process and models of radar measurement errors can then be used before the aircraft data reaches a graphical display. Another feature, that is implemented for realistic radar simulation, is that new aircraft data is only available when the aircraft was swept by the radar signals. The internal structure of the distributor enables primary as well as secondary radar to be simulated.

- Generator

The generator process produces initial states of aircraft arriving at the simulation boundary. It can either read from a data file containing recorded or specified arrival aircraft data, or as in the current prototype, generate arrival aircraft data according to specified stochastic distributions. During the start of the generator process, a data file is read that contains all the initial data of the entry points to be simulated. These can be the meter fixes of the TRACON airspace or any other locations. Other initial conditions of the entry point are statistical data such as the distribution of aircraft types or airlines, which are retrieved from a special data file. This data file can be generated by using the recorded data of an existing airport.

In addition to the random aircraft generation, an aircraft can also be entered manually during the simulation process.

- Controller Process

The controller process is an interface used to direct the aircraft in the airspace. For this purpose, the interface has common controller tasks and procedures implemented.

After it starts, the controller connects to the administrator process and establishes a data and message channel. This is used to send commands by the controller to the administrator where they are forwarded to the respective aircraft as flight advisories. The controller uses an input line to write messages to the aircraft. A parser that contains standard phrases of the language used by controllers, processes this input. The aircraft on the other side of the communication channel uses a similar mechanism to handle commands and send confirmations back to the controller. As means to watch the aircraft, the controller uses the GUI display interface. Eventually the controller process will be equipped with a separate GUI interface.

- <u>Observer</u>

This process is connected to one of the distributor output channels and prints the aircraft data to a terminal shell. It can be used to get a quick information on the data without having to start the GUI and analysis tools. Due to the restricted number of characters in a terminal shell this only works for a small number of aircraft in the observed airspace.

- <u>Graphical User Interface (GUI) Display</u>

The GUI display shows the aircraft in a selected airspace. Eventually, two types of GUI displays are planned: information display for controllers and displays of "actual" traffic situations. GUI displays for controllers illustrate aircraft status based on measurements and standard controller console display format, whereas actual traffic displays directly provide aircraft states generated in the simulation program. In the current prototype, this actual traffic display has been implemented.

For the GUI the script language Tcl/Tk is used. This language has as its biggest advantage of being not system specific. A Tcl/Tk script runs on every machine where the Tcl/Tk has been installed. The code is thus independent of the machine's window system.

Figure 3 shows a screen-shot of the GUI display that has been implemented in the current prototype. The

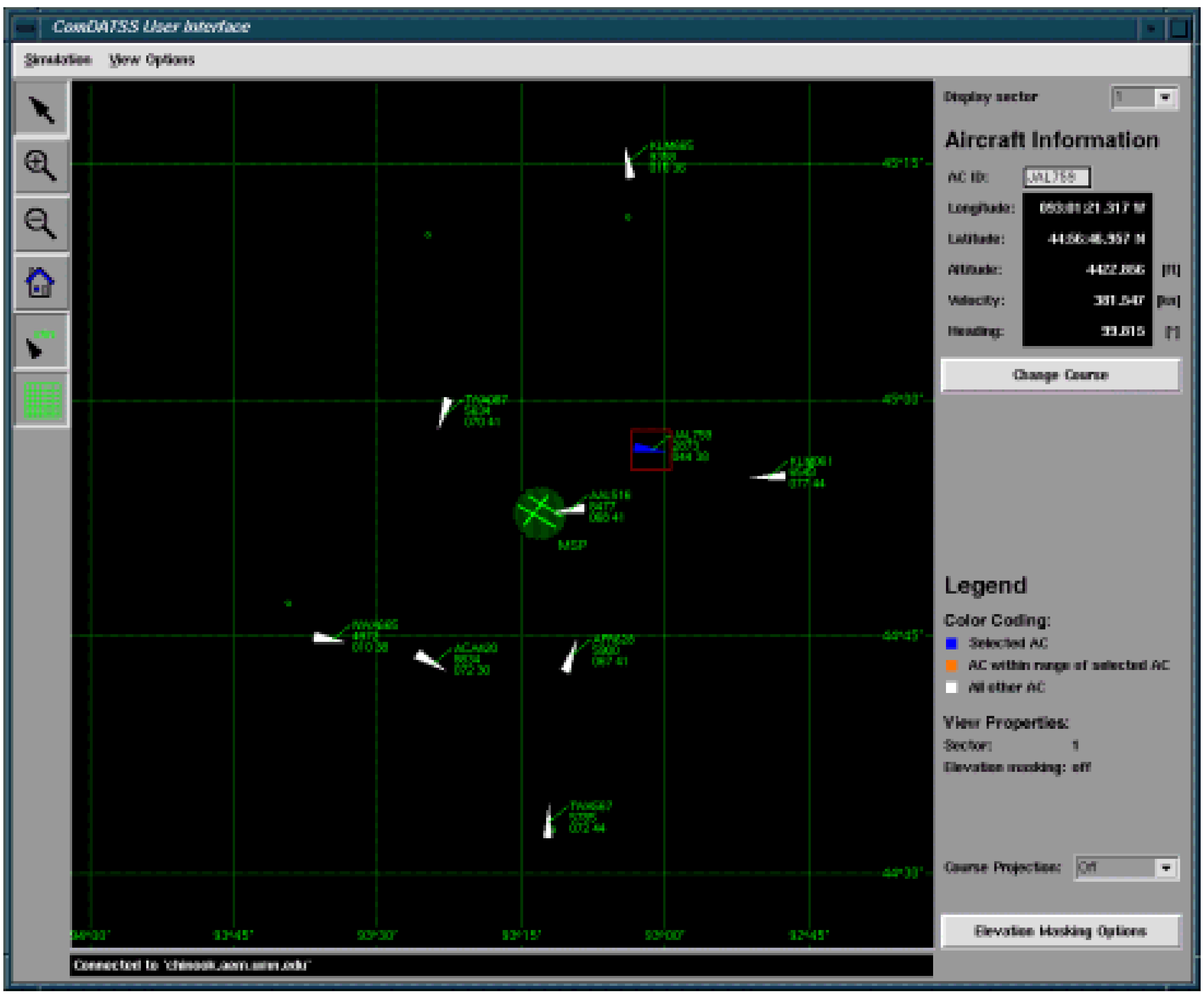

Fig. 3 Screen shot of the traffic display.

aircraft are symbolized by an arrowhead icon, which points into their heading direction. Attached to this icon are displays of the call sign of the aircraft, and information on the aircraft's altitude and velocity. A latitudinal-longitudinal grid is an additional help to keep better track of the aircraft positions. An icon marks the location of the Minneapolis - St. Paul International Airport (MSP). In addition to this, small circles indicate the locations of the five meter fixes of the TRACON area. There are several options to manipulate the display area. Part of the display can be zoomed out to observe a certain air traffic situation in more detail. Furthermore there can be several sectors defined, each of which can be selected by a drop-down box. By the selection of the respective button, the aircraft information tag and the latitudinal-longitudinal grid can be switched on and off. By selecting an aircraft with the mouse pointer its current data is displayed in the aircraft information group box.

- Analysis of a Simulation Run

For the analysis of a simulation run, ComDATSS records all aircraft data in data files. In these files, the states of the aircraft and other relevant information are stored as functions of time. The sampling rate can be varied so that short-term simulations as well as long-term simulations can be evaluated without having to create huge data files. The output data is stored in a simple format with tabulators as space delimiters, so that it can be imported by applications like Matlab® or Microsoft® Excel for analysis purposes. In addition to the aircraft information, a message log can be saved to a data file as well. It contains messages from the system as well as communication logs between the controller and the aircraft.

Mathematical Modeling of Objects in the Prototype of ComDATSS

Modeling needs for the ATM components in Figure 1 are now discussed. Because of the modular structure, any changes in the models of individual objects can be easily made without affecting the overall simulation program structure.

- Arrival Traffic Modeling

Certain initial arrival traffic information is needed to start a computer simulation. Essential information for the arrival traffic should include arrival times at a fix, initial aircraft positions, altitudes, speeds, and headings, aircraft types, aircraft weights, and other useful information. The ComDATSS program is designed to be able to use two types of initial traffic data: user-specified traffic data (e.g., recordings of actual traffic data) or stochastically generated traffic data. The ability to generate traffic scenarios stochastically is highly desirable for a simulation program, as it enables one to use the program quickly and is useful in theoretical modeling studies of arrival traffic patterns.

In the prototype of the ComDATSS program, traffic distributions among different arrival streams are modeled by specified probability ratios. The total arrival rate is a parameter that can be changed. Mixes of aircraft types along each stream are also modeled by probability ratios, such as (0.8/0.12/0.04) for large/small/heavy. The Boeing 757 aircraft is considered a special class due to its turbulent wake characteristic. Gross weights of arrival aircraft are generated according to truncated Gaussian distributions, where the means and standard deviations depend on aircraft types. Initial aircraft conditions at specified boundary points are aircraft locations (x, y), altitudes (h), and airspeeds. Truncated Gaussian distributions are used for generating these initial conditions, where the means and standard variations depend on aircraft types. Finally, a combination of exponential distribution and uniform distribution is used to model inter-arrival times. Because of the modular structure of the simulation program, any of these distributions can be changed easily.

- Modeling of Actual Aircraft Trajectories

In practice, actual aircraft trajectories are obtained when pilots control aircraft flights by following specified and/or perceived flight commands. Actual aircraft trajectories are affected by aircraft navigation and status displays, pilot feedback control responses, aircraft response dynamics and performance limits, and weather conditions. Therefore, the generation of aircraft trajectories in ComDATSS should include models of flight objectives, navigation errors, pilot responses, aircraft dynamics and performance limits, and weather conditions. All these models should allow for inclusions of appropriate randomness.

- Aircraft Flight Objectives

In actual flights, pilots control aircraft according to flight plans, desired arrival times, published STAR or SID procedures, company regulations, and controller advisories. The Aircraft Flight Objective component synthesizes nominal flight paths for an aircraft based on the current aircraft state, flight plans, final runway orientation, standard airspace procedures, and controller commands. Controller commands are for pilots to avoid other aircraft or to achieve a certain specified time of arrival (STA) at the specified point, and may include changes in speed and altitude, as well as insertions of additional waypoints for vectoring. Outputs of the flight objective module are a set of minimum flight commands that specify a complete aircraft trajectory.

The use of a Flight Objective Module offers great simulation flexibility. It can easily be modified to account for free flight trajectories. Adherence to airspace constraints such as airways and terminal flight routes is also reflected in the flight objective.

Adherence to airspace geometry is achieved through specifications of flight objectives. As a result, the simulation program can simulate any specified airspace geometry such as center or TRACON. At the present time, flight objectives connect initial aircraft states to a specified runway threshold. Therefore, the current prototype is designed for simulating traffic over Center/TRACON airspace to runway touchdowns. Figure 4 shows a typical terminal airspace geometry.

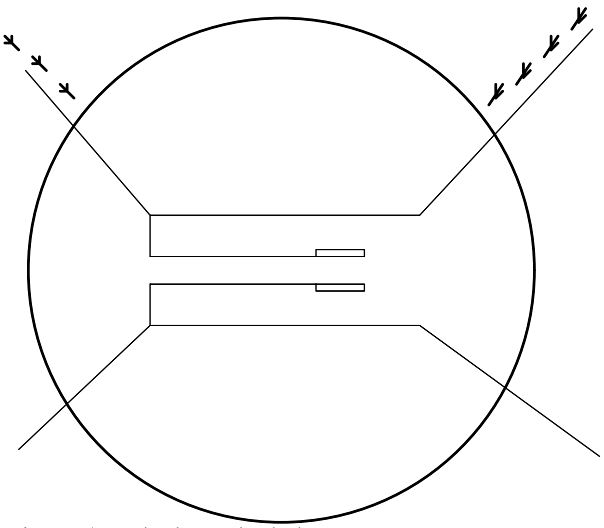

Fig. 4 A typical terminal airspace geometry.

- Pilot Feedback Control Actions

Pilots apply controls in a feedback manner to fly aircraft in order to achieve flight objectives. Different pilots may have different characteristics. Models of actual pilot flying techniques are too complicated to express mathematically. In the current prototype, pilot control actions are modeled by a combination of time delays and nonlinear inversion. Time delays reflect pilot reaction times. Nonlinear inversion is a technique for designing nonlinear feedback control laws. Its central idea is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that experiences of linear control designs can be applied. Feedback control gains are selected to match frequency and damping of pilot controls.

It is important to develop stochastic models that describe average and variations of piloting techniques. Variations of piloting techniques are expressed by using probabilistic coefficients of pilot control laws and reaction times. In the current prototype, truncated Gaussian stochastic distributions are assumed on feedback control gains. Feedback control gains are determined randomly for each aircraft and remain constant during the entire flight of that aircraft.

Uses of specific control inputs (elevator, aileron, rudder, thrust) depend on assumptions on flight procedures and flight objectives. Details are omitted. These control laws specify pilot actions as functions of measured aircraft states. Navigation errors can be incorporated to pollute the on-board measurements of aircraft states. In addition, weather conditions and aircraft performance limits affect aircraft motions when these control laws are applied to aircraft equations of motion. In the current version, point-mass equations of motion are used.

- Communication, Navigation, and Surveillance Models

Navigation and surveillance processes introduce errors to the original sources of signals, and can be considered as measurements of actual signals. In the simulation, these measurements are expressed as the "actual" signals from the internal computer simulations plus some random errors. There are three types of aircraft measurements: on-board navigation systems, ground-based (primary and secondary) radar systems, and air-ground data-links. These systems determine aircraft states with varying levels of accuracy. Characteristics of the random errors can be selected to reflect characteristics of different measurement processes.

Pilot/controller communication is conducted via voice in the current air traffic control system and may possibly be done in the future via direct data-link. Mathematically, the communication process introduces delays and random errors to the original advisories that are transmitted. The amount of delays is also random in nature.

- Controller Decision Module

The controller decision module can either be connected to an automation tool or manual input from a computer terminal. As a result, it can be used to test automated air traffic control tools (such as algorithms of aircraft scheduling or conflict resolution, like Active FAST), or to accommodate a human controller in the simulation loop. In the latter case, graphical displays of traffic information are used for controller manual inputs of flight advisories.

It is challenging to develop models for inner controller decision processes. Figure 1 shows a preliminary attempt to adopt the CTAS structure. Design of logical structures in the controller decision module will be considered at a later time. The current focus of the simulation program development is to develop an air traffic environment with all components except the working logic inside controllers' head.

- Simulation Alert

Simulation alert is a background process that checks if there is any conflict based on "actual" aircraft trajectories generated in the computer simulation program. The purpose is to provide statistics on "real" aircraft separations and delay capabilities. If every possible pair of aircraft is checked, the simulation alert process will surely use up a substantial amount of computer time. As computer memory and speed advance, it will become eventually feasible to compare all aircraft trajectories. At the present time, simulation alert is checked on any aircraft that is mouse-pointed on the display. In the next step, a random scheme will be deployed that determines aircraft to be checked. For example, some incoming aircraft can be selected according to a certain probability and these aircraft will be followed by the alert process until they land. As computer speed increases, the selection probability can be made higher as well.

- Dispatchers

Dispatchers help pilots to determine flight plans. A module is allocated for the modeling of dispatcher inputs.

- Weather Model

The weather model in the simulation is implemented as an object in the OOP environment. Each aircraft object can request atmospheric data at its current location. This data is then used in the equations of motion and the feedback control of the respective aircraft. In the current version, the weather model returns zero winds and standard atmosphere conditions. A more advanced weather model can replace this module later.

Demonstration

A laptop demonstration or a pre-recorded video will be used to illustrate the functionality of ComDATSS.

Summary

This paper discusses the need and requirements for system-wide computer simulations of air traffic systems as numerical testing environments in research and development of advanced ATM automation tools and concepts. After a review of related work, the paper reports the development of a prototype computer program for a comprehensive dynamic air traffic system simulation known as ComDATSS. The design of computer implementation architecture is explained, and various modeling issues are examined.

Acknowledgements

This project is supported by NASA Ames Research Center under NCC2-990, monitored by Tom Davis. We thank Doug Isaacson for many helpful discussions.

References

1. Williamson, T., and Spencer, N. A., ``Development and Operation of the Traffic Alert and Collision Avoidance System (TCAS)," Proceedings of the IEEE, Vol. 77, No. 11, Nov., 1989, pp. 1735-1744.

2. Walsh, J., and Wojciech, J., ``TCAS in the 1990s," Navigation: Journal of the Institute of Navigation, Vol. 38, No. 4, Winter 1991-92, pp. 383-397.

3. Erzberger, H., Davis, T. J., and Green, S.,"Design of Center-TRACON Automation System," presented at the AGARD Guidance and Control Symposium on *Machine Intelligence in Air Traffic Management*, 11-14 May 1993, Berlin, Germany.

4. Erzberger, H., "Design Principles and Algorithms for Automated Air Traffic Management," AGARD Lecture Series No. 200 on *Knowledge-Based Functions in Aerospace Systems*, Madrid, Paris, and San Francisco, November 1995.

5. Denery, D. G, and Erzberger, H., "The Center-TRACON Automation System: Simulation and Field Testing," NASA TM 110366, August 1995.

6. Davis, T. J., Erzberger, H., Green, S. M., and Nedell, W., "Design and Evaluation of Air Traffic Control Final Approach Spacing Tool," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 4, July-August 1991, pp. 848-854.

7. Davis, T. J., Krzeczowski, K. J., and Bergh, C., "The Final Approach Spacing Tool," presented at the 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, California, September 1994.

8. Denery, D. G., Erzberger, H., Davis, T. J., Green, S. N., and McNally, B. D., "Challenges of Air Traffic Management Research: Analysis, Simulation, and Field Test," AIAA Guidance, Navigation, and Control Conference, August 11-13, 1997, New Orleans.

9. Radio Technical Committee on Aeronautics (RTCA), Free Flight Implementation, Final Report of RTCA Task Force 3, published by RTCA, Inc., 1140 Connecticut Ave., NW, Suite 1020, Washington, DC 20036, October, 1995.

10. den Braven, W., and ten Have, J. M., ``NARSIM, A Real-Time Simulator For Air Traffic Control Research," NLR TP 90147 L, National Aerospace Laboratory NLR, The Netherlands.

11. Odoni, A. R. et al., "Existing and Required Modeling Capabilities for Evaluating ATM Systems and Concepts," Final Report, Modeling Research under NASA/AATT, March 1997.

12. Bilimoria, K., Sridhar, B., Chatterji, G., Sheth, K., and Grabbe, S., "FACET: Future ATM Concepts Evaluation Tool," draft paper, 1999.

13. Miles, E. S., Davis, P. C., and Wing, D. J., "Development of a Free-flight Simulation Infrastructure," AIAA 99-4193, 1999.